DOCUMENT RESUME

ED 329 239 IR 014 898

AUTHOR	Duffield, Judith A.
TITLE	Problem Solving Software: What Does It Teach?
PUB DATE	Apr 90
NOTE	27p.; Paper presented at the Annual Meeting of the American Educational Research Association (Boston, MA, April 16, 1990).
PUB TYPE	Reports - Research/Technical (143) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; Computer Software; Grade 4; Instructional Effectiveness; Intermediate Grades; Learning Strategies; Pretests Posttests; *Problem Solving; Teaching Methods; *Thinking Skills; *Transfer of Training

ABSTRACT

The purpose of this study was to examine the potential of computer-assisted instruction (CAI) for teaching problem solving skills. It was conducted in three phases. During the first phase, two pieces of problem solving software, "The King's Rule" and "Safari Search," were identified and analyzed. During the second phase, two groups of six fourth-grade students were each observed using one piece of software for seven 30-minute sessions. Think-aloud protocols were collected at the beginning and end of the observational period. Posttests were administered to assess problem solving ability and transfer. In the third phase, these data were first analyzed separately by software, then the results were compared. While the students used limited versions of the strategies the software claimed to teach, students were also found to have developed several strategies that allowed them to succeed in the program without using the desired strategies. No transfer of the problem solving strategies was observed. This type of research will provide valuable clues for the design of effective problem solving software. (21 references) (Author)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it
Minor changes have been made to improve reproduction quality
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

ED329239

Problem Solving Software:
What Does It Teach?

Judith A. Duffield

Department of Educational Technology
College of Education
San Diego State University
San Diego, California 92182-0311

Presented at the Annual Meeting of the
American Educational Research Association
Boston, MA

April 16, 1990

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Judith Duffield
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR014898
ERIC

Abstract

The purpose of this study was to examine the potential of computer assisted instruction for teaching problem solving skills. It was conducted in three phases. During the first phase, two pieces of problem solving software, The King's Rule and Safari Search, were identified and analyzed. During the second phase, two groups of six fourth-grade students were each observed using one piece of software for seven 30-minute sessions. Think-aloud protocols were collected at the beginning and end of the observational period. Posttests were administered to assess problem solving ability and transfer. In the third phase, these data were first analyzed separately by software, then the results were compared. While the students used limited versions of the strategies the software claimed to teach, students were also found to have developed several strategies that allowed them to succeed in the program without using the desired strategies. No transfer of the problem solving strategies was observed. This type of research will provide valuable clues for the design of effective problem solving software.

Problem Solving Software: What Does It Teach?

One of the fundamental goals of education is to teach our students to become successful problem solvers, yet how people learn to solve problems is still not completely understood (Andre, 1986). While some researchers and theorists believe that problem solving can be taught through general heuristics (Bransford & Stein, 1984; Rubenstein, 1980), others point to the importance of domain-specific strategies (Gick, 1986) and domain-specific knowledge (Gick, 1986; Greeno, 1980). Research comparing the problem solving skills of novices and experts have found that all three are important (Larkin, 1980). Novices tend to make better use of general problem solving strategies than experts, but experts make better use of strategies that are domain-specific and have access to more and better organized domain-specific knowledge than novices.

The extent to which general or specific problem solving strategies are useful, in that they are transferable to new problem situations, seems to be dependent upon the context in which those strategies are learned (Adams, 1989). Strategies learned in a narrow domain, and applied only to the solution of one type of problem, are less likely to be applied to other appropriate problem situations than those strategies that are taught in a range of contexts.

Gagne (1985) defines problem solving "as a process by which the learner discovers a combination of previously learned rules and plans their application so as to achieve a solution for a novel problem situation" (p. 178). This definition points toward the importance of learning the prerequisite intellectual skills and verbal information, and having the opportunity to apply them, first to familiar, then to new, problem situations. In addition to selecting the correct knowledge to solve a problem, the student must also be able to adopt an appropriate problem solving strategy. The degree to which the learner can transfer knowledge and appropriate strategies to new problems would be a measure of problem solving ability (Mayer, 1987).

The implications for education at this point seem to be to supply the learner with domain-specific knowledge, general and specific problem solving strategies, sufficient practice in using them in a variety of contexts, and then the opportunity to apply them to new situations. But how? Providing ample opportunities for 30 students to discover higher order rules requires time, planning, and resources that are not always available to the average teacher.

Computers are thought to hold great potential for creating environments for developing problem solving skills. Problem solving and simulation software has been found to increase general problem solving skills, domain-specific problem solving skills, and domain-specific knowledge (Norton & Resta, 1986; Woodward, Carnine & Gersten, 1988). As computers become more available to classrooms across the country, they hold the potential for providing students with the opportunity to combine skills and knowledge to discover higher order rules.

A review of software catalogs shows a growing awareness of this potential on the part of producers and distributors. Most catalogs have sections devoted to programs that claim to teach problem solving, but what do these programs really teach? Occasionally publishers will provide teacher's editions and manuals that describe the objectives and the strategies designed into the software, as well as suggestions for additional activities, but most educational software, which is packaged for the home market, contains little more than basic operating instructions. Information beyond the age or grade level and subject area is rarely provided. Thus, teachers and administrators have little to go on when they make decisions about the applicability of the software for their students.

While a variety of general software evaluation instruments exist (e.g., Wager, Wager, & Duffield, 1989), there is a lack of criteria specifically designed to evaluate software that claims to teach problem solving skills. How can teachers and administrators identify software that effectively teaches problem solving skills? What attributes should such software possess? This study was designed to begin to answer these questions by identifying the characteristics of specific pieces of problem solving software and determining how that software affects student learning behaviors, student acquisition of skills, knowledge, and problem solving strategies directly addressed by the software, and student ability to transfer those problem solving strategies to new problems.

The study consisted of three phases. The purpose of the first phase was to identify the characteristics of particular examples of problem solving software. During this phase, two pieces of problem solving software were selected and analyzed in detail. The selection was based on obtaining two pieces of software that represent the range of available programs. Most software that claims to teach problem solving falls into two general categories: (a) that which uses a puzzle type environment to teach problem solving, and (b) that which teaches problem solving in the context of a school subject. One piece from each category was selected and analyzed in light of problem solving theory and research to determine the expected outcomes and the pedagogical strategies used to obtain those outcomes.

During phase two of this study, students were observed using the two pieces of software in order to determine the effects of the software on student learning behaviors. These behaviors included student interactions with the software and the other students, and strategies used to solve problems. Also during this phase, a posttest was administered to assess student ability to solve problems similar to those included in the software and student ability to transfer this problem solving ability to new problems. In phase three, the information collected in the first two phases of the study was analyzed in order to identify how various characteristics of the software seemed to affect student learning behaviors, student acquisition of problem solving strategies directly addressed by the software, and student ability to transfer those problem solving strategies to new problems.

This study is intended to be the first step toward the development of an evaluation instrument for problem solving software. As such, it is of importance to researchers and practitioners alike. This study provides those interested in the development, evaluation, and selection of problem solving software, as well as those interested in how problem solving skills are acquired, with descriptions of how specific characteristics of problem solving instruction presented by a computer affect problem solving abilities in students.

METHOD

Analyzing the Software

Software Selection Process

Software that claimed, either in its catalog description, its packaging, and/or its supporting documentation, to teach problem solving was identified. From this pool of over 50, two pieces of software were selected. To reflect the range of available problem solving software, one piece of software was selected from each of two, fairly evenly divided, general categories. One piece, The King's Rule (O'Brien, 1983), teaches problem solving primarily in the context of a school subject. This type of software emphasizes the use of the skills and knowledge taught in math, social studies, or some other school subject as the primary vehicle for teaching problem solving skills. The students are required to apply their knowledge of the subject to be able to solve the problems

ERIC

presented. The other piece of software, Safari Search (O'Brien, 1985), teaches problem solving primarily in a puzzle type of an environment. The problems in this type of environment generally require very little in the way of prior skills or knowledge in school subjects. They emphasize the use of mazes, search strategies, multiple classification rules, the replication of patterns, or other general strategies to solve problems.

While the two pieces of software used in this study differ in terms of the context in which the problems are presented, the selection decision was based on obtaining software that was similar in terms of the target learner group, the complexity of the problems presented, the number and difficulty of the skills to be learned, and the estimated time required to complete a session. The availability of the software also affected the selection decision.

Software Analysis Procedure

In order to identify possible criteria that could influence the effectiveness of software designed to teach problem solving skills, a review of the literature was conducted in two areas of theory and research: problem solving and computer assisted instruction (Duffield, 1989). Those factors likely to affect the acquisition of problem solving abilities and those factors likely to influence the effectiveness of computer assisted instruction were identified. The two pieces of software selected were then analyzed in terms of the identified factors.

The factors examined during the software analysis process included the following: the events of instruction embedded in the software (Gagne, 1985), the stated or implied instructional objectives of the software, the type of learning outcome (Gagne, 1985), the stated or implied prerequisite skills, the general and specific problem solving strategies embedded in the software, the range of contexts within which the strategies, skills, and knowledge included in the software are taught, the degree to which these strategies, skills, and knowledge are directly or indirectly taught, the type of practice activities provided, the type of problems presented, the type of feedback the students are given when they practice solving the problems, grade level, and supplemental print materials. A detailed description of these factors and a complete analysis of the selected software may be found in Duffield (1989).

The King's Rule. The King's Rule is problem solving software that uses a castle theme to present problems in a math context. To solve a problem, the students study a given set of three numbers that follow a mathematical rule randomly selected from the pool of rules for that level and propose additional sets that may or may not follow the same rule that the given set follows. After each set, the students are told whether or not their set follows the rule. This is the hypothesis testing portion of the program. If the students receive several "no" responses, another set that follows the same rule will be given as a clue. When the students think they know what the rule is, they take a five-question quiz. All five questions must be answered correctly to pass the quiz and solve the problem. Tokens are awarded for each problem solved and three tokens are required to pass to the next level.

The rules used are of three basic types: (a) operation, (b) equation, and (c) characteristic. The rules introduced at the first two levels perform an operation (add, subtract, multiply, or divide) with a constant on a beginning value to create the second number in the set. The same operation is applied to the second number to create the third number in the set. If the rule was to multiply by five, a given set could be 2, 10, 50. Equation rules use the first two numbers in the set to create the third. If the rule was that the first number divided by the second number equals the third, a given set could be 24, 4, 6. Characteristic rules require one or more numbers in each set to have an attribute, such as being an odd number or being divisible by three. Other rules require the numbers to be in numerical order or that the sum of the numbers in the set have a characteristic, such as being even or divisible by a constant. A summary of the rules presented at each level is provided in Table 1.

Insert Table 1 about here

The required prerequisite knowledge is basic math facts. The strategies the software intends to teach are the domain-general strategy of hypothesis testing and the domain-specific strategy of identifying patterns and relationships. The problems presented are problems of induction that share surface similarities, but vary in their underlying structural principles. These differences and the underlying principles are not made explicit, but left for the students to discover.

Safari Search. Safari Search is problem solving software that uses a safari theme to present problems in a puzzle context. The students select the animals they wish to search for from 12 games or safaris. In each safari, the students are presented with a five by five grid in which one or two animals have been randomly hidden. The students select one of the 25 boxes in the grid and open it. The clue they receive varies according to the rules of the safari and the location of the animal(s). The students continue gathering clues until they think they know where the animal(s) is/are hiding. The clues are counted, but no score is kept from one safari to the next. In the first three safaris, the students are immediately told if they open a box where an animal is hiding. In the rest of the safaris, the students must press Escape when they are ready to guess where the animal(s) is/are hiding.

There are five basic types of puzzles that each use similar specific strategies: (a) random guessing, (b) proximity, (c) distance, (d) rows and columns, and (e) direction. The clues provided in the one random guessing puzzle simply tell the student whether the animal is there or not. This puzzle is intended to show how inefficient trial and error is as a strategy. The clues in proximity puzzles tell if the animal is in a box that is next to the selected box. The clues in distance puzzles tell how far away from the selected box the animal(s) is/are. The clues in rows and columns tell how many animals are in the same row and/or column as the selected box. The clues in direction puzzles tell whether or not the animal(s) is/are to the left, right, above, or below the selected box. The types of clues given at each level is summarized in Table 2.

Insert Table 2 about here

No prerequisite knowledge was identified. The strategies the software intends to teach are the domain-general strategies of making inferences, inventing tactics, and collecting, organizing, and using information. The domain-specific strategies the software intends to teach are the strategies the students invent to solve each type of puzzle. The problems presented are problems of induction that share surface similarities, but vary in their underlying structural principles. These differences and the underlying principles are not made explicit, but left for the students to discover.

Observing the Software in Use

Subjects

The subjects were students at the research school at The Florida State University. Students are selected to attend the school so that they represent the same proportions of ability, sex, race, and socio-economic status that exist in the school-age population of Florida.

After the software used in the study was selected and analyzed, 105 third and fourth grade students were tested on (a) their ability to perform the prerequisite skills identified for both pieces of software and (b) their ability to solve problems similar to those presented by the software. The students were also given a list of software which included those pieces being used in the study. The students were asked to indicate which pieces of software they had used.

From the pool of students tested, those students who were eligible to participate in the study were identified. Those who indicated that they had used the selected software were eliminated from the study. Those who could not perform the prerequisites were also eliminated, as were those who were able to solve problems similar to those taught by the software. Of the remaining students, two independent groups, each consisting of three girls and three boys were randomly selected. All of the students selected were in the fourth grade. Each group used one piece of software.

A group size of six was chosen in order to provide a group that was large enough for the students to be able to freely interact with each other, while still small enough for the individual student's learning behaviors to be observed. This size group provided a sufficiently wide variety of learners for observing similarities and differences in learning between and within groups. A larger group would have made individual observations more difficult and would not have been likely to add to the range of observed behaviors.

Observational Procedures

Students in both groups used their software for seven 30-minute sessions. The time allowed reflected an estimate of the time required for the students to become proficient witn the software. All sessions took place in a computer lab. Each group received instructions on how to operate the computer and the software during the first session. These instructions provided the minimum amount of information required for the students to be able to begin to use the software, and did not include information about the problem solving strategies involved. The students were also asked not to use the software outside of the lab for the duration of the study. The students were allowed to interact with each other during the sessions, but worked primarily on their own. The experimenter only offered technical assistance in the use of the computer and the software.

While the students were using the software, the experimenter collected observational data on the learning behaviors of the students by (a) taking notes and (b) tape recording the sessions. The notes were taken using a form divided into six boxes, one for each student, arranged to reflect the locations of the computers in the laboratory. Arrows between boxes were used to record student interactions. Rather than attempt to systematically observe the students, the experimenter decided which students to observe based on what interactions were taking place and on making sure that each student was observed during each session. A tape recorder was placed close to the middle of the room and left running during each session.

After each session, the notes and the tape recording were reviewed and a narrative description of the observations was constructed. These observations included how the students interacted with the software and each other, how long each student took to solve one problem as presented by the software, how long each student took to consistently be able to solve the problems presented by the software, how quickly each student tired of the program, and what the students seemed to be learning, as indicated by their statements and questions.

Think-Aloud Procedures

Twice during the observational period, the students were each asked to solve a problem using a think-aloud protocol (Ericsson & Simon, 1984). Collecting a protocol involved having a student sit at a computer with the experimenter. The students were instructed to say what they were thinking as they solved one or more problems. The experimenter would occasionally ask the students to explain (a) why a box or set was chosen, (b) what they thought the rule was or where they thought the animal might be, and (c) how they solved a problem. The number of problems solved during a protocol depended on the difficulty of the problems and the time taken to solve them. Notes were taken to indicate the boxes opened and the sets tested. Each protocol was taped and the tape transcribed.

During the first session, the students received instruction on how to think aloud while solving problems, and were encouraged to practice while using the software. The first think-aloud protocols were collected during the second session, after the students had had the opportunity to learn to use the software, but before they were likely to be consistently able to solve the problems. A second set of think-aloud protocols was collected during the two days following the conclusion of the observational period. Due to scheduling conflicts, these final two sessions occurred on two separate days and were organized by homeroom, rather than by software group. The information gathered during the think-aloud sessions was used to determine which strategies the students were using to solve the problems presented by the software and how each student's use of problem solving strategies changed over time as a result of using the software.

During the same sessions in which the second think-aloud protocols were collected, students were administered a posttest that assessed their ability to (a) solve problems similar to those included in the software and (b) transfer this problem solving ability to new problems.

ERIC

Measurement Instruments

Prerequisite skills test. A pretest instrument was developed to assess student ability to perform the prerequisite skills identified for The King's Rule. No items were developed for Safari Search because no prerequisites skills were identified. The prerequisite skills identified for The King's Rule were basic number facts in addition, subtraction, multiplication, and division, and the identification of odd numbers. A 49-item test was developed that included 10 fill-in-the-blank type items for each type of number fact and nine items for odd numbers. A split-half, Spearman-Brown reliability coefficient of .97 and an internal consistency reliability coefficient of .95 were obtained with the 105 participating third and fourth grade students.

Problem solving skills tests. Pretest and posttest instruments for each piece of software were developed to assess student ability to solve problems similar to those included in the software. These instruments required the students to apply the problem solving strategies addressed in the software in order to solve problems similar to those included in that software.

The pretest and posttest developed for The King's Rule each consisted of four short-answer items. Each item used one of eight rules selected from those used in the first three levels of the program. An attempt to match the rules by difficulty was not completely successful and resulted in the forms not being parallel with regard to the domain-specific strategy of discovering patterns and relationships. The tests should be parallel for the domain-general strategy of hypothesis testing. When the pretest and posttest scores obtained from 49 third and fourth grade students were compared, it was found that 76% of the students attained the same score, or within one point of the same score, on both tests.

The pretest and posttest developed for Safari Search each consisted of four short-answer items using two types of puzzles presented by the program. When the pretest and posttest scores obtained from 49 third and fourth grade students were compared, it was found that 84% of the students attained the same score, or within one point of the same score, on both tests.

The methods for assessing the reliability of both the pretests and posttests for The King's Rule and Safari Search indicate the reliability of the tests over time when group means are compared and do not assess the reliability of the tests for individual scores. Therefore, only group means will be discussed when examining the results of these tests for the study participants. The majority of information on the development of individual problem solving strategies will be obtained from the observations and the think-aloud protocols.

Near transfer tests. Posttest instruments for each piece of software were developed to assess student ability to use the skills, knowledge, and problem solving abilities taught by the software to solve new problems. The measures of near transfer required the students to apply the skills and knowledge addressed in the software to related types of problems in the same domain.

A 5-item, short-answer instrument was developed for each piece of software to assess near transfer. The near transfer instrument developed for The King's Rule consisted of math word problems. A split-half, Spearman-Brown reliability coefficient of .75 and an internal consistency reliability coefficient of .84 were obtained with the 12 fourth grade study participants. The near transfer instrument developed for Safari Search consisted of map skills. A split-half, Spearman-Brown reliability coefficient of .83 and an internal consistency reliability coefficient of .83 were obtained with the 12 fourth grade study participants.

Far transfer tests. The measures of far transfer required the students to apply the skills and knowledge addressed in the software to related types of problems in other domains. The problem solving pretests and posttests developed for one piece of software were used as transfer tests for the students who had not been exposed to that software because the software analysis revealed that both pieces of software were designed to teach similar skills.

RESULTS

Observational Data

The six students selected for each piece of software used their software for seven 30-minute sessions over a two week period. The students were asked to talk aloud while they used the software and were allowed to interact freely. They were allowed to help other students as long as they did not solve the problem for the student they were helping.

The observations made while the students were using the software resulted in narrative descriptions of the learning behaviors of the students. These descriptions were analyzed in terms of the behaviors that were predicted by the software analysis.

The students were arbitrarily assigned pseudonyms before the study began. The first letter (A through F) indicates the student and the second letter (K or S) refers to the software used. These will be used when individual students are referred to. The results of both analyses follow. The results for The King's Rule are summarized in Table 3 and for Safari Search in Table 4.

The King's Rule

General Observations

The students using The King's Rule seemed to enjoy the program and remained on task and involved throughout the study. At the end of the first week, the students asked if they would be returning the following week. They were happy to find out that they would be.

As expected, the program was reliable and easy to use. The directions were clear and within the reading ability of the students. The students understood that they were to enter sets of numbers and could take a quiz, but several had trouble understanding what was meant by a rule. It took three sessions and several explanations for all of the students to realize that the given set was not the rule, but a set produced by applying the rule. As a result, when the given set was 3, 10, 17, for example, a student with this misunderstanding would subtract the numbers and write the differences as the new set. Since there are only two differences, a third seven would be added to complete the set. The students expressed discomfort at doing this, but could think of no other solution. AK and BK persisted in calling the given set the rule throughout the sessions, even after they had learned to distinguish between the two.

Levels achieved. The students varied in their progress through the levels of the program. Two students, DK and FK, were the only ones to reach level 2 during the first session. DK went on to reach level 4 the next session. He continued working on level 2 through level 4 through the next three sessions. During the last two sessions he worked on level 3 and level 4. On the day of the posttest, DK completed level 6 while the think-aloud protocols were being collected. FK stayed on level 2 during the second session and worked her way up to level 4 during the third session. She spent the rest of the sessions working on levels 2 and 3.

It took CK and EK three sessions to reach level 2. CK reached level 3 during the fifth session and EK reached level 3 during the seventh session. The other two students, AK and BK, reached level 2 during the fourth session and never did reach level 3. One student, AK, missed the sixth session and EK had to leave early during the fifth session.

Prerequisite knowledge

The first two levels of the program were intended to serve as a review of the prerequisite skills and as an introduction to the program. While all the study participants had passed the prerequisite skills test, this test did not measure the level of automaticity of those skills. As a result, only two students were able to rapidly progress beyond the first two levels of the program. The rest were severely limited by the speed at which they could calculate. They counted on their fingers to solve the addition and subtraction rules in level 1. To solve the multiplication and division rules in level 2, they also used counting sticks (marks made on paper for counting) and multiple addition (15 + 15 + 15 instead of 3 x 15) when they ran out of fingers.

Often, these students were able to state the rule being used, but were unable to create a new set that tested their hypothesis. An inability to select an appropriate beginning number and calculation errors appeared to cause most of the problems. A common error in subtraction rules was to start the set too low. When the last number in the set went below zero, the students were stumped and usually made the last number a zero, which, of course, meant that the set did not follow the rule. A common difficulty with division rules resulted from picking a beginning value that could only be evenly divided by the rule once. The last number in the set was usually designated as zero when this happened.

Target Strategies

Domain-general strategies. The domain-general strategy of hypothesis testing was identified in the software analysis as one of the intended products of the problem solving activities in the software. If students had adopted this strategy, they would have exhibited it by (a) trying several possible rules for a set, (b) testing sets that would eliminate competing rules, and/or (c) trying sets that should not fit the rule. It would also be expected that, since the first three levels of the program use given sets that make the rules transparent, these behaviors would be less visible in these levels than the last three levels, where the given sets are designed to appear to follow several rules.

All students tested hypotheses. They examined the given set, found a possible rule, and entered a set that would test whether or not their rule was correct. The most common strategy used was to subtract the numbers to see if the differences were the same. If they were, the students constructed sets by adding or subtracting that difference. If the differences were not the same, DK, FK, and, in the later sessions, CK, usually went on to try other rules. The others usually tried subtracting again or constructing sets based on one of the differences.

As expected, DK, as the only student to regularly work on level 4, was most often observed demonstrating behaviors that indicated that he was testing hypotheses. He was the first to try an odd number hypothesis and to recognize an A x B = C rule. When confronted with an A x B = -C rule, DK tried several hypotheses, usually ignoring the minus sign. FK attempted to help him by suggesting alternatives and discussing what the minus might mean, since they had never encountered negative numbers before. DK eventually figured out the rule and passed the quiz.

DK was the only student observed entering sets that he thought should not follow the rule he hypothesized. He would enter a set and say that he did not think that it would work, but he just wanted to see. Occasionally the set did work, and he would say "What? Oh, no, that means it couldn't be..." and other expressions of surprise. No students were observed developing several alternative hypotheses at once and devising sets to eliminate one or more of the competing rules.

All of the students entered sets following different rules when they could not figure out a rule or when a set brought unexpected feedback. Unfortunately, these different sets were not always constructed to test a hypothesis. When the students had no idea what the rule was, the strategy they often adopted was to try whatever had worked in the past, in hopes that it would work again. This behavior was most often observed when the students moved up to a new level and continued to try to apply the domain-specific strategies they had learned in the previous level.

Both DK and FK, and, to a lesser extent, CK commonly tried to apply several hypotheses to a given set. They would decide on a rule and test it with one set. If that set did not follow the rule, they would try to find another rule that would work. In doing so, they would go through a long list of what the rule could and could not be, expressing their varying degrees of confidence (e.g., I know it couldn't be multiplication, and I don't think it could be subtraction, but it might be division).

Domain-specific strategies. The domain-specific strategy of recognizing numerical patterns and relationships was identified in the software analysis as one of the intended products of the problem solving activities in the software. If students had adopted this strategy, they would be expected to be observed examining the sets for a variety of patterns and relationships.

All students recognized some patterns and relationships. The types of rules the students considered seemed to depend on the types of rules they had encountered in the past and the length of time spent on levels 1 and 2. DK and FK were much more likely than the other students to look for a wide variety of relationships or patterns in the given sets. AK, BK, CK, and EK only looked for a function that would transform one number in the set to the next. For example, the question they would try to answer was: What happened to the first number to turn it into the second and to the second number to turn it into the third? While AK, BK, and EK were only observed testing addition and subtraction rules during the study, CK also tested multiplication and division rules.

DK and FK were more likely to consider the possibility that a set might have been constructed following a common attribute rule (e.g., all odd numbers), or that two of the numbers were used to create the third (e.g., A + B = C). DK was able to recognize an A x B = C rule by just studying the given set for a few minutes. Recognizing common attribute rules was more difficult. The first time he encountered a common attribute rule, DK observed that the numbers in his given set were all odd, but he was unable to construct a set to test and failed the quiz. FK commented that she had not thought about all the numbers being odd. She had considered that they might all be even, but not odd.

The students developed more specific strategies that enabled them to detect patterns and relationships in the given set. DK found division rules difficult to solve until he noticed that they were the same as multiplication rules, but with the set in reverse order. AK, BK, CK, and EK adopted the strategy of counting the interval from the first number in a given set to the second number in order to determine the rule. Unfortunately, they persisted in this strategy when they advanced to level 2, where it was no longer effective.

Unanticipated Strategies

Several unanticipated strategies were observed. Two of these strategies most likely developed out of (a) an inability to solve the rule, and (b) equating "yes" feedback with success in the program. Several students adopted the strategy of entering the given set. That was the one sure way that they could get a "yes." They were not real sure why it worked, but it did, and that was enough to make it useful to them. Another strategy was to try a set that had worked before, on a different rule. Both these strategies seemed to be used when the students had no clue as to the identity of the rule.

Two students, CK and FK, were observed entering sets like 44, 44, 44 or 1, 1, 1 several times in a row, even though the given set could not have followed the same rule. They had developed a strategy for obtaining a clue. By repeatedly entering a set that they knew did not follow the rule, they were able to reach the clue threshold for the program. They used this strategy whenever they had tried one or two sets and still had no idea what the rule was.

When the students received a given set that was too difficult to figure out, they generally gave up and went to the quiz. They knew that the worst that could happen was that they would fail the quiz and lose their tokens, which would happen anyway since they couldn't figure out the rule. This way, they would at least have the possibility of getting an easier rule in the next round.

Taking the quiz without knowing the rule turned out to be quite an effective strategy. The program randomly presents five given sets selected from the rules at that and, occasionally, other levels. Rarely do more than one or two of the given sets in the quiz follow the rule being tested. Since those sets that don't follow the rule are randomly selected, it is often quite obvious which they are. While this strategy was used by all of the students, DK was more likely to talk about giving up, while at the same time persisting with the rule. The others had few qualms about opting for the quiz without knowing the rule and were less persistent.

When taking the quiz, it was easy for the students to become distracted and forget which rule they had discovered. To combat this, BK wrote down the given set as a reminder, before he took a quiz. The others relied on their memory. If they could not remember and failed the quiz, they would return to hypothesis testing, remember the rule by looking at the given set, and immediately return to the quiz.

Insert Table 3 about here

Instructional Factors

While the hypothesis testing portion of the program gave ample opportunity for the students to propose sets and receive feedback, the quiz did not adequately test attainment of the rule. As mentioned above, the students were often able to pass the quiz without knowing the rule. This resulted from distractors that were randomly chosen, rather than being selected to test a particular rule. In fact, CK consistently checked only the first two numbers in the given set to see if they followed the rule and was never penalized for not checking the third number.

The clues, given to help students who had received several no responses, were discovered by few of the students. Most did not try enough sets to reach the clue threshold before giving up or discovering the rule. The clues on the first two levels rarely seemed to provide additional information that resulted in solving the rule. Occasionally, the clue would be the same as the given set. While CK and FK continued to use their clue generating strategy, they generally could not go on to solve the rule after seeing the clue.

Student interaction. The students each worked at their own computer, but interacted in loosely defined pairs. Generally, these pairs were matched by the level they worked on. While the pairs AK and BK, and CK and EK worked quietly side by side, DK and FK had a more vocal interaction. AK and BK seemed to only discuss the number of tokens each had. CK and EK often agreed to work together, each waiting for the other to enter sets at the same time, even though they had different rules. CK would occasionally compare tokens with FK. When AK was absent, BK moved next to DK and talked to him about tokens and which rules each had been given. Rarely did AK, BK, CK, or EK leave their chairs.

DK and FK did not sit next to each other and were often out of their chairs, looking over each other's shoulder. They frequently asked each other for help, refused, then gave in. They regularly compared token status and went to great lengths to justify a lack of tokens ("I would have had four by now, if I hadn't lost that quiz." "Well, I could have had some chess pieces, but I went back to level 2."). Each was very aware of what the other was saying to themselves, the other students, or to the experimenter and often joined in the conversation.

The students were requested to talk about what they were thinking as a practice for the think-aloud protocol session. Except for one conversation about a basketball game that had just ended, the topics discussed centered on The King's Rule. Frequently, the conversation was self-directed. The students talked themselves through difficult problems or expressed how they were feeling. These feelings ranged from disappointment at having lost their tokens to unabashed pride. When FK encountered a rule that she thought was tricky, she announced that she would figure it out because she was "the master." When the students talked to each other, it was likely to be about the number of tokens each had received or lost, or to ask for or give help.

While the program was designed to use a discovery method of instruction, the students attempted to turn it into guided discovery. They were told that they could help each other as long as they did not tell the answer or do the problem for the student. Students were observed helping each other in almost every session. This behavior increased as the sessions progressed. The help took the form of asking leading questions, explaining the directions, and giving helpful hints. CK and EK helped each other by discussing what the rule might be. CK tried to help EK figure out a multiplication rule, but EK was unsuccessful. AK received help from DK and FK. They usually asked him questions about the set or gave him hints, like working from right to left for a dividing rule.

Helping also occurred incidentally, as a result of the students thinking aloud. The other students would appear to not be paying attention, but later they would try out the same sets and/or strategies.

Safari Search

General Observations

The students using Safari Search seemed to enjoy the program. They remained on task and involved throughout most of the study. The students were generally reluctant to leave at the end of each session.

As expected, the program was reliable and easy to use. The students quickly learned how to operate the program and before long were successfully locating animals. The students enjoyed the graphics, but the sound was too loud and distracting so it was turned off after the second session.

Safaris searched. The students were allowed to choose any safari they wished, but were encouraged to try them all. All of the students reported or were observed trying each of the one-animal safaris. All of the students tried at least one of the two-animal safaris, though few regularly selected two-animal safaris before the fourth session. ES missed the third, fourth, and fifth sessions.

Target Strategies

Three domain-general strategies were identified in the software analysis as the intended products of the problem solving activities in the software. They are the ability to (a) make inferences, (b) invent problem solving tactics, and (c) collect, organize, and use data. An analysis of the observations related to each strategy follows.

Making inferences. If the students had learned to make inferences, they would be expected to be able to examine clues in light of their meaning for a particular safari and determine where the animal(s) is/are located. This behavior was observed in all students. Everyone was able to successfully infer the location of some of the animals, but no one was able to successfully infer the location of animals in every safari.

The optimal strategy identified in the teacher's guide was not followed consistently by any student. While all of the students were observed taking advantage of the review to examine their clues, many clues were ignored, particularly those that conveyed that the animal was not seen. Rather than looking at all of the clues and determining where an animal could and could not be, the students were more likely to look for particular clues or only consider the last one or two clues. Which clues they looked for was determined by the safari the students were playing. For example, in the sixth one-animal safari (donkey), students would look for the box with negative clues in all directions. In the second two-animal safari (rhinos), the students would look for the two boxes that contained a zero.

It became apparent that some safaris were more likely than others to encourage the students to consider more clues. Only the first, fourth, and fifth two-animal safaris (kittens, kangaroos, and cats) are difficult to solve without examining all of the clues. The other safaris may all be solved by locating one or two particular clues. The inferences made in these puzzles are so intimately tied to the rules for each that, what was intended to be a single, domain-general strategy looks much more like a series of domain-specific strategies. For example, the seal may be found by continually choosing a box that is the correct distance away. It is not necessary to eliminate all the possible locations and consider multiple clues, the seal will eventually be found even when all but the current clue are ignored.

Finding the kittens is not so easily accomplished. While the same strategy may be followed, the kittens will not be found unless the clues are reviewed and considered together. Since the clues do not indicate to which kitten they refer, the students must find the two boxes that account for all of the clues.

Inventing tactics. If the students had learned to invent problem solving tactics, they would be expected to demonstrate specific search strategies and ways of interpreting clues in order to locate the animal(s). These behaviors were observed in all students. At first, the students seemed to search randomly, often reopening boxes several times. By the second session, planned search strategies began to develop. The object of these search strategies was generally a target clue. The search strategy would be used until the target clue was found, then more specific strategies related to that safari would be employed to locate the animal(s). For example, a common strategy used to locate a flamingo or a loon was to start in the upper left-hand corner and open boxes in a clockwise spiral until a hot, warm, or yes clue was found. Then the students would stop searching in a spiral and look for the animal in that area.

Several search strategies were invented to solve random guessing, proximity, and rows and columns safaris. Several of the students used a spiral search strategy. Students would start in the upper left-hand box and open every box as they moved across that row, down the right-hand column, back across the bottom row, and then up the left-hand column, continuing in this manner until they found the clue they were looking for. As they progressed, one or two of the students modified this strategy by skipping every other box.

Another common search strategy was to search row-by-row. The first row of boxes would be opened from left to right, then the next row from right to left, continuing in this manner down the grid. FS adc;-ted a variation of this strategy. He opened all of the boxes in the center column, the right-hand column, and then the left-hand column.

The row-by-row search strategy was occasionally used with distance and direction puzzles. The purpose was to quickly open all of the boxes and then study the clues with the review. The clues were rarely studied as the boxes were opened. This strategy was observed less often in the later sessions. As the clues and the rules of each safari became better understood, the students paid more attention to the clues and were able to solve some of the safaris with fewer clues than before.

Another search strategy used to solve distance and direction puzzles was to begin at either the upper-left hand or center box and use that clue to determine where to go next. In the distance puzzles, the number of boxes indicated in the clue would be counted off in a straight or stair-stepped line. In the direction puzzles, the clues would be followed in the direction indicated by the yes feedback.

In one variation of this follow-the-clue search strategy, CS counted the boxes to the next clue as if she were reading. She proceeded to the left, then, at the end of the row, returned to the left end of the next row, where she began to count again. The use of this strategy indicates that she did not understand the concept of rectilinear distance. She persisted in this strategy throughout the study, despite repeated explanations, because it was successful. By following the clues around, she would eventually find the hidden seal.

The students invented target-clue strategies for locating the animals in half of the safaris. These strategies involved looking for a clue that would lead directly to an animal. Each target clue was closely related to the rules for that safari, and, in most cases, appeared to be simply a logical extension of the rules. It was difficult to observe who invented each strategy because, in most cases, they were quickly adopted by the rest of the students. This dissemination of strategies seemed to occur when (a) students talking aloud to themselves were overheard, (b) students with the strategies helped others, and (c) students agreed to work on problems in the same safari, side-by-side.

A target-clue strategy was invented for both of the proximity puzzles. The target clue for the flamingo was either warm or hot. Finding either clue meant that the flamingo was in an adjacent box. The target clue for the loon was the block of yes clues. The loon was the yes that was completely surrounded by other yes clues.

Target clues were also found for the rows and columns puzzles. In both, the students looked for the lines of 1s. The dragon is always hiding where they cross. Since there are four of these lines that form a rectangle in the snails safari, the students also looked for the two 2s that are always on opposite corners of that rectangle. The snails can always be found on the other two corners.

Target clues were identified for only one distance and one direction puzzle. The rhinos are located in the two boxes that have zero as one of their two clues. The donkey is located in the box where the answer to all of the directions is no. Similar target clues could have been developed for the cats and the llamas safaris, but were not observed.

A strategy for retaining the location of the animal(s) in short-term memory was used by all of the students. The students would review their clues, determine where the animal(s) was/were located, and mark the box(es) with their finger(s). That way, when the clues disappeared, they could remember the location(s) long enough to declare their answer.

<u>Collecting, organizing, and using data</u>. If the students had learned to collect, organize, and use data, they would be expected to obtain clues, determine which animal the clue referred to (when applicable), and be able to interpret the clues in order to determine where each animal was located. Because the students were all able to locate animals, they seem to have acquired this strategy, at least to some degree.

All of the students were able to collect data or clues by simply following the basic program instructions. Few seemed able to truly understand all of the information provided by the clues. Negative sounding clues, such as cold, no, or 0, were treated as though they provided less information than positive clues. For example, a yes in the loon safari tells that the loon is in one of nine boxes. A no tells that the loon is not in any of nine boxes. Each provides the same amount of information, but the information provided by a no clue was consistently ignored when that no was next to a yes. It took CS and DS all seven sessions to consistently be able to locate a loon. Both used the target-clue strategy of locating the box surrounded by yes clues, but would also select a box that was next to a no.

Similarly, when a target-clue strategy was used, the students sometimes ignored clues that were close to the target value. ES was looking for zeros in the rhino safari. When he found a one paired with another number, he used the other number to determine the next box to open. He did not seem to realize that the clue with the one meant that the box was just one away from the searched-for zero. The simplified strategies adopted for searching for clues are almost all successful, eventually, but they are not necessarily accompanied by a deep understanding of the information contained in the clues.

While all of the students were able to interpret the data well enough to solve the easier safaris, little was observed to indicate that the students were able to organize and interpret the information received from the more difficult clues. The kitten and kangaroo safaris were the safaris with the hardest clues to interpret and organize. They could not easily be solved using target-clue or other strategies. Of the three students who attempted the kitten safari, only AS was observed to be successful. She found both kittens once. Only BS was observed searching for kangaroos. He uncovered all of the clues and still could not figure out where the kangaroos were.

The record sheets provided in the teacher's guide were made available to the students. AS and FS were the only ones to use them. The students did not use the sheets for organizing data, planning strategies, determining possible animal locations, or next boxes to open. Instead, both AS and FS only used the sheets to copy their clues from the review.

Insert Table 4 about here

Instructional Factors

As expected, the one random guessing puzzle, Intuit the Iguana, was useful for showing the lack of power associated with a trial and error strategy. DS and FS were thrilled to see how easy it was to find an iguana, at first. FS soon pointed out that, while it was an easy safari, it was hard, too, because it could take so long to find the iguana. The students only chose this safari occasionally, and less often as the study progressed.

The directions were the only new information presented. While no learning guidance was provided, the directions were not expected to cause problems. Unfortunately, this was not the case. While the students were able to read the directions, they often chose not to do so. Those students who did read the directions did not always understand them, as shown by their lack of understanding of the meaning of the clues.

As the students became familiar with the program and tried new safaris, they would assume that the clues in the new safari meant the same as those in the previous safari. The surface similarities between the clues allowed the misunderstanding to continue. The students would typically play the new safari as if it were the previous safari. When they couldn't find the animal(s), the students either gave up or asked for help. This problem was encountered throughout the study, as students tried new safaris.

Those safaris that required the students to state the location of the animal(s) seemed to adequately assess student performance. The students would have to interpret the information in the clues in order to be able to infer the location of the animal(s). It is unlikely that students would often be able to locate the animal(s) by chance.

Student interaction. The students each worked at their own computer, but were free to interact with each other. Only CS and DS always worked as a pair. Occasionally, AS would join them, but most of the time she worked alone. BS always worked alone. At first, he was very quiet, but gradually started to share his successes with the rest of the students. ES missed the

ERIC

middle three sessions. When he was in attendance, ES tended to work alone, but interacted with FS occasionally. FS worked on his own puzzles alone, but frequently interacted with all but BS.

The students were allowed to help each other as long as they didn't tell the student they were helping where the animal was hiding. As a result, helping consisted of short descriptions of the strategy that should be used. While this facilitated the spread of strategies through the group, it inhibited much of the development of individual strategies. The discovery method of instruction intended by the software was turned into guided discovery or direct instruction by the students with regard to the specific strategies. The underlying principles involved in the puzzles were not discussed.

Think-Aloud Protocols

Think-aloud protocols were collected twice during the observational period. The first protocols were collected during the second session, after the students had had the opportunity to learn to use the software, but before they were likely to be consistently solving problems. The second set of protocols were collected after the last observational session. Each student was asked to think out loud while they solved at least one problem presented by the software.

In order to analyze the think-aloud protocols, a coding system was developed (Ericsson & Simon, 1984). This coding system was based on the target strategies, and the behaviors described in the previous sections, that would show that those strategies were being used. The tape and notes made during each protocol were transcribed. When the students exhibited the expected behaviors, it was an indication that that strategy was being used. When the exhibited behaviors did not match the expected behaviors, they were analyzed to determine which other strategies the students were using. This analysis resulted in a description of the strategies the students used to solve problems at the beginning and the end of the observational period.

During the software analysis, the problem solving strategies explicitly or implicitly taught by the software were identified. The strategies the students used were compared to those the software was intended to teach to the students. The degree to which students exhibited these strategies or others was noted.

If the software was constructed to require the students to use a specific strategy, then the students would be expected to employ that strategy during both think-aloud sessions. While greater proficiency would be expected during the second think-aloud session, the students should have employed that strategy to some degree from the beginning. If the students were not using the predicted strategies, this would indicate that those strategies were not being learned. The results of both analyses follow.

The King's Rule

Domain-General Strategies

The domain-general strategy identified in the software analysis was that of hypothesis testing. Those students who had adopted this strategy would be expected to (a) try several possible rules for a set, (b) test sets that would eliminate competing hypotheses, and (c) try sets that should not fit the rule.

While all of the students tested at least one hypothesis, alternate hypotheses were only tested if the first one failed. None of the students developed sets to eliminate competing hypotheses, but one student tested a set that he thought should not fit the rule.

During the first think-aloud session, only two students exhibited the expected strategy. CK and FK proposed alternative hypotheses after their first hypothesis failed. DK solved his problem on the first attempt and did not need to develop alternatives. AK, BK, and EK each proposed only one hypothesis. They gave up when it failed.

Use of the hypothesis testing strategy increased during the second think-aloud session. CK, DK, EK, and FK all proposed alternative hypotheses after their first hypothesis failed. In addition, DK tested a set that he said should not work if his hypothesis was correct. AK and BK each tried one hypothesis and then gave up.

Domain-Specific Strategies

The domain-specific strategy identified in the software analysis was that of recognizing numerical patterns and relationships. Students who had adopted this strategy would be expected to examine the given sets for a variety of patterns and relationships.

During the first think-aloud session, each student was asked to solve a level 1 rule. FK was the only student who did not recognize the equal intervals between the numbers in the given set. While she tried addition, subtraction, multiplication, and division, calculation errors prevented her from being successful. Only two of the others, CK and DK, were able to construct sets to test their hypothesis and solve the rule. Both AK and BK made a set of the differences between the numbers in the given set. EK was not able to construct a set.

The students were much more successful in determining numerical patterns and relationships during the second think-aloud session. AK, BK, DK, and EK were each able to recognize and solve an addition or subtraction rule. CK was able to recognize and solve a multiplication rule. When presented with multiplication or division rules, AK, BK, and EK continued to apply addition and subtraction strategies to determine the relationship between the numbers. When presented with level 3 rules, CK, DK, and FK each tried a number of strategies to find the relationship between the numbers. Only DK was successful.

Unanticipated Strategies

Unanticipated strategies were observed in two students when those students could not solve a rule. During both think-aloud sessions, AK had two favorite sets that he entered when his first hypothesis failed. The sets, 3, 6, 9 and 4, 8, 12, had worked at other times, and he apparently wanted to check them just in case they worked this time. EK, during the first session, and AK, during the second session, both explained how a yes could be obtained by copying the given set. EK studied the given set for a few seconds and then announced that you could get a yes by copying the given set. When asked if she wanted a yes, she said that she did because that is how you get the rule. AK explained about the copying strategy after he was unable to find a rule that worked.

Safari Search

Three domain-general strategies were identified in the software analysis: (a) making inferences, (b) inventing problem solving tactics, and (c) collecting, organizing, and using data. Each was observed in the students during both think-aloud sessions. The first strategy, making inferences, would be demonstrated if the students were able to locate the animals by examining the clues. All of the students successfully located at least one animal during both sessions.

Only CS and DS were not able to solve every safari they attempted during the first session. Both had difficulty interpreting the clues. CS first located a seal, but, when she tried to find a loon, she was unable to guess where it was, even though she had found enough clues to tell her where it was. When DS tried to find a seal, she opened a box with a four inside. Instead of trying a box four boxes away, she opened an adjacent box. She opened adjacent boxes five times, ignoring the fours and fives inside, before she said that she didn't understand and gave up.

During the second session, AS selected a safari she had never played. She used a strategy that had been successful with a different safari and was unable to locate the animals. The students were able to solve more difficult safaris during the second think-aloud session, indicating that they had learned to make inferences.

If students had invented problem solving tactics, they would be expected to demonstrate specific search strategies and ways of interpreting clues that allowed them to locate animals. While it is difficult to determine which students invented particular tactics, all students were observed using a variety of tactics during both think-aloud sessions.

All of the students used a random search strategy during the first think-aloud session. This strategy was still being used by four students during the second session. During the first session, AS, BS, and CS were the only ones to use a clue to determine which box to open next. All of the students used this strategy during the second session. A target-clue strategy was another strategy to be used by just three students during the first session, and all of the students during the final session. Only BS and FS were able to explain the reason a target clue worked. BS explained that the donkey was not above, below, or to the left of the box, and "I couldn't go right because I'd hit the wall (edge of the screen) and so it must be here." FS was able to explain that the snails were located where the rows and columns containing the twos crossed. The others, when questioned, just restated the clue and CS said that it was in the directions. CS was the only student to use a search pattern during the first session. She counted the boxes from left to right, across the rows. In the second think-aloud session, four students used a search pattern. AS and DS both used a clockwise spiral pattern, BS began by looking in all four corners, and CS used the same left to right pattern that she used in the first think-aloud session.

Two unique strategies were observed. During the first think-aloud session, ES adopted a strategy of counting eight boxes between every clue. He apparently took this strategy from a practice item on the pretest. By the second think-aloud session, FS had adopted the strategy of always opening the center box first because that box "could see the most."

Those students who could collect, organize, and use data would be expected to be able to obtain and interpret clues. The optimal strategy described in the teacher's guide of determining where each animal could and could not be and then choosing boxes to eliminate locations was never observed, though two students came close. The students took several clues into consideration at once, combining the information provided to decide where to look next for the target clue. For example, FS, looking for snails, was trying to find twos because "sometimes they tell you where the snails are and sometimes they don't." Rather than searching over the whole grid, he found rows and columns that contained ones and followed them to where they crossed. He was looking for a pattern in the clues, rather than just the targeted twos. Similarly, BS seemed to eliminate blocks of boxes as he searched for a donkey. He looked at the below clues until he found a no, then looked to the right of the boxes in that row until he received another no. Once he had eliminated an area, he did not return to it. Both BS and FS were observed using these strategies during the second think-aloud session in safaris they were very familiar with.

All of the students were able to obtain clues, but not all clues were correctly interpreted. It was common for all or part of the information provided by the clues to be ignored during both sessions. For example, AS ignored the information provided by a warm clue to continue looking for hot clues. She found the flamingo, but several clues later than she should have, had she interpreted all of the clues correctly. Similarly, DS looked in an adjacent box after receiving a warm clue when she should have looked in one that only shared a corner with it. She found the flamingo after she found a "hot" in the next box. When looking for kangaroos, AS had isolated the area where they were by the seventeenth clue, but did not realize this. She could not find the target clue she was expecting, based on the donkey safari, so she began searching outside that area and finally gave up after over 30 clues.

CS was so intent on finding a zero in the rhino safari that she ignored the other clues that could have lead her to one. She searched randomly, ignoring the information provided by the non-zero clues. When she found a box with the clue 3:0, she opened an adjacent box rather than looking for the second zero three boxes away. Even though she found the first zero in the first box she opened, it took her 23 clues and two tries to find the rhinos.

ES was able to use more of the clues than CS to find rhinos, but still ignored much of the information provided. He usually used the higher number in the clue to determine which box to open next. For example, if the clue was 4:1, he would look for his next clue four boxes away, ignoring the fact that he was just one box away from a rhino.

DS also looked for a loon in the second think-aloud session. She found a block of six yes clues that was surrounded by no clues, but her first guess was in the corner, a location that did not account for two of the yes clues. She guessed correctly on the second try.

Measurement Instruments

The pretest and posttest results were analyzed in order to determine the degree to which the students in each group could solve (a) problems similar to those included in the software, (b) new problems in the same domain, and (c) related types of problems in other domains. The degree to which the students were able to solve these problems is another indication of the effectiveness of the software. The pretest was given the week before the observational sessions. The posttest was given after the conclusion of the observational sessions and during the same session the think-aloud protocols were collected. The results of both analyses follow.

The King's Rule

The problem solving skills test was intended to assess student ability to solve problems similar to those in the software. The pretest and posttest results for the problem solving skills test indicated a small increase in student performance on this measure. The pretest mean was 1.5 (S.D. = 1.4) and the posttest mean was 2.2 (S.D. = .98). Both tests had four items each. Two of the six students achieved a gain of two points from the pretest to the posttest. The other four students achieved within one point of the same score on both tests.

The far transfer test was intended to assess student ability to apply the skills and knowledge addressed in the software to solve related types of problems in other domains. Since The King's Rule and Safari Search were designed to teach similar skills, the far transfer test for The King's Rule is the same as the problem solving skills test for Safari Search. The pretest and posttest results for the far transfer test also indicated a small increase in student performance on this measure. The pretest mean was .17 (S.D. = .41) and the posttest mean was 1.0 (S.D. = 1.55). Both tests had four items each. This rise in standard deviation was due to two students who achieved a gain of two and three points from the pretest to the posttest. The rest of the students were unable to answer any questions correctly on either pretest or posttest.

The near transfer test was intended to assess student ability to use the skills, knowledge, and problem solving abilities taught by the software to solve similar problems in the same domain. The results of the posttest for near transfer indicated that all but one of the students were able to use the problem solving abilities taught by the software to solve new problems in a related domain. The mean score on the five-item posttest was 3.5 (S.D. = 1.9). This test was also administered to the students using Safari Search in order to assess the reliability of the test. A mean of 3.0 (S.D. = 1.67) for the Safari Search group indicates that both groups performed nearly the same. No near transfer pretest was developed.

Safari Search

The problem solving skills test was intended to assess student ability to solve problems similar to those in the software. The pretest and posttest results for the problem solving skills test indicated no increase in student performance on this measure. The pretest mean was .67 (S.D. = .82) and the posttest mean was 1.17 (S.D. = 1.17). Both tests had four items each. One of the six students achieved a gain of two points from the pretest to the posttest. The other five students achieved the same score, or within one point of the same score, on both tests.

The far transfer test was intended to assess student ability to apply the skills and knowledge addressed in the software to solve related types of problems in other domains. Since The King's Rule and Safari Search were designed to teach similar skills, the far transfer test for Safari Search was the same as the problem solving skills test for The King's Rule. The pretest and posttest results for the far transfer test also indicated no increase in student performance on this measure. The pretest mean was .83 (S.D. = .75) and the posttest mean was .83 (S.D. = .98). Both tests had four items each. All of the students achieved the same score, or within one point of the same score, on both tests.

The near transfer test was intended to assess student ability to use the skills, knowledge, and problem solving abilities taught by the software to solve similar problems in the same domain.

The results of the posttest for near transfer indicated a wide range in student achievement on this measure. Half of the students answered four of the five questions correctly. Two of the students only answered one question correctly and one student was not able to answer any questions correctly. The mean score on the five-item posttest was 2.3 (S.D. = 1.9). This test was also administered to the students using The King's Rule in order to assess the reliability of the test. A mean of 2.7 (S.D. = 2.25) for the The King's Rule group indicates that both groups performed nearly the same. No near transfer pretest was developed.

DISCUSSION

The results of the observations of the software in use, the think-aloud protocols, and the tests were reported separately for each piece of software then compared in order to determine how various characteristics of software seemed to affect student learning behaviors, student acquisition of problem solving strategies directly addressed by the software, and student ability to transfer those problem solving strategies to new problems.

Prerequisite Knowledge

Problem solving requires the application of prior knowledge and strategies to the current problem in order to produce new knowledge and/or strategies. Prerequisite knowledge was only identified for The King's Rule. The first three levels of the program were intended as an introduction to the program and as a review of the prerequisite basic math facts. Progress through these levels seemed to be related to the automaticity of the math facts. Only half of the students possessed skills automatic enough to allow them to pass the first two levels consistently.

The problem of a lack of automaticity could have been overcome had the data management capabilities of the computer been used to allow the students to request the computer to perform a calculation for them. This could have occurred on a different screen or the students could have been allowed to enter formulas when they were entering their sets. For example, the set 5, 15, 45 would be entered as 5, 3x5, 3x15, with the computer performing the calculation as it is entered so the student would be able to use the resulting number in the next formula. In its current form, The King's Rule will not perform calculations.

Domain-General Strategies

The domain-general strategies each piece of software claims to teach are very similar. In these programs, testing hypotheses and making inferences both involve examining data (sets or clues) in light of a set of rules (of math or of a game) to determine patterns and relationships. The optimal strategies described in the teacher's guides both involve identifying all the possible rules or locations based on the available information and testing the feasibility of those rules or locations with sets or clues designed to eliminate competing rules or locations.

For example, if The King's Rule students were using the optimal strategy and received the given set, 10, 20, 30, they would think of several hypotheses that would fit it: (a) add 10, (b) multiples of 10, (c) any numbers in numerical order, (d) even numbers, (e) sum is divisible by 10, and (f) A + B = C. The students would then design a set that would test several hypotheses at once. For example, if the set, 6, 4, 10, followed the rule, it would eliminate the first three hypotheses.

Students using Safari Search to solve a distance puzzle would look at their first clue, 4 for example, and figure out all of the locations (four spaces away from the box) where the animal could be. All other locations would be automatically eliminated. The next box opened would be selected so that, whichever number was revealed, it would eliminate some of the possible locations identified by the 4.

The optimal strategies were rarely used by either group of students. Students were able to test hypotheses in The King's Rule, but, rather than testing several at once, they tested a series of hypotheses, one at a time. Because they did not develop multiple hypotheses, the students using The King's Rule also were not observed testing sets to eliminate competing hypotheses. Similarly, students using Safari Search to solve distance puzzles never considered all of the locations where an animal might or might not be located. They picked one location, tried it out, and used the clue in that box to determine where to go for the next clue, usually ignoring all of the information provided by the previous clues.

The strategy the students adopted is more consistent than the optimal strategies with information-processing theories of problem solving. According to these theories, the problem solver searches for a possible solution, tries it out, and only continues to search for more solutions if that one fails (Gick, 1986). But this strategy is also very basic. Perhaps the students would have learned to use a more complex strategy if this basic strategy had not been so successful.

Students using Safari Search were also expected to use the domain-general strategies of (a) inventing tactics and (b) collecting, organizing, and using data. Both of these general strategies were observed, to varying degrees, in all of the students, though several of the tactics they invented did not follow the directions for the safaris or were not very useful (e.g., always counting eight boxes to the next clue, counting across one row then back across the next, or opening all of the boxes to look for one specific clue and ignoring all the others). Unfortunately, many of these inappropriate tactics were successful, especially in the first three one-animal safaris, where the students could solve the problem without having to state where they thought the animal was. CS never did learn to count rectilinear distances, but she always found her seal.

Domain-Specific Strategies

The specific tactics and data management strategies the students in the Safari Search group used coincided with how students in The King's Rule group used domain-specific strategies. The specific strategies students in both groups used were often inappropriately applied. Students using The King's Rule had difficulty recognizing new number patterns and relationships because they continued to apply the same strategies, useful in one level of the program, to other, inappropriate levels. They failed to see that old strategies no longer applied. Students using Safari Search had trouble with some puzzles because they tried to apply target-clue strategies appropriate for one safari to other safaris.

The difficulties the students experienced in moving between levels and safaris seems directly related to the surface similarities between the problems in each program. When structural similarities and differences are not made explicit in the presence of strong surface similarities, novices and experts, alike, have difficulty solving problems (Novick, 1988). Had the differences between levels and safaris been made more explicit, the students should have had fewer difficulties moving to the next level or safari.

Students in both groups had trouble interpreting the given information. Many of the clues in Safari Search were ignored, especially those that sounded negative, such as cold, no, or 0, or were not the specific clue that the students were looking for. Ignoring the information provided by negative or non-target clues was apparently caused by the students failing to recognize that all clues provide the same amount of information. Knowing where an animal is not located can be just as informative as knowing where it might be.

In The King's Rule, while one facet of the optimal strategy was for the students to test sets they thought would receive a no response, ignoring negative information was encouraged by the program. Each no was accompanied by the same negative sound that accompanied failing a quiz. It is not surprising that students tried to avoid receiving a no. Also, additional sets are only given as clues after a number of negative responses have been received. Might not students who had received several yes responses and had not solved the rule be just as much in need of an additional set? And wouldn't there be a benefit in providing those students with a set that did not follow the rule?

Software-specific strategies. While the goal of both pieces of software was to teach the specified problem solving strategies, the students had another goal in mind: to win. Ideally, these two goals should be compatible. The students should only be able to win by following the optimal strategy. This was obviously not the case. The students in both groups were able to develop several strategies to bypass features in the software.

To deal with frustration caused by difficult rules or safaris, the students often quit the rule or safari they were in by trying to guess the animal's location or by taking the quiz. The students using Safari Search were rarely successful in guessing the one or two correct locations out of the 25 possibilities. Nonetheless, they often tried to do so just to get to an easier safari. The students using The King's Rule, on the other hand, were often successful by guessing. Whether they were successful or not, the students using The King's Rule also knew that the chances that they would get an easier rule the next time were high.

Transfer of Training

One measure of problem solving ability is the degree to which the students are able to apply the strategies taught in the problem solving activities to other, unrelated problems (far transfer) or to new problems in the same domain (near transfer). No evidence of transfer was found with either piece of software. Students attained the same score, or nearly the same score, on the far transfer posttests as they did on the far transfer pretests. Students in both software groups attained nearly the same score on their near transfer posttest as students who had not used that software.

This lack of transfer was expected because both The King's Rule and Safari Search present problems in a single context. Transfer of training is less likely to occur when problems are presented in a narrow range of contexts than when they are presented in a wide variety of contexts (Vye & Bransford, cited in Bransford, Sherwood, Vye, & Reiser, 1986).

A second factor that may have had an effect on transfer was the instructional method used. Discovery is less effective than guided discovery in promoting transfer because students may or may not discover the underlying principles that are involved in the solution, making it unlikely that they will be able to recognize other circumstances where the solution will be applicable in the future (Gagne & Brown, 1961; Kittell, 1957).

Guided discovery could have been incorporated into The King's Rule in several ways. For example, at the beginning, students could have been shown how one set could follow several rules and shown how to select a second set to eliminate some of the rules. It would have been even more effective if, rather than having to wait for help, the students could have requested help at any point in the hypothesis testing portion. In addition to giving additional sets that follow the rule, this help could have given a set that did not follow the rule, or suggested questions that the students could ask about the set that would help them recognize patterns and relationships. For example, some patterns become clear when you ask: What happened to the first number to turn it into the second, and to the second to turn it into the third? Other patterns are easier to see if you ask: What do these numbers have in common?

In Safari Search, as part of the directions for each safari, guidance could have been added by demonstrating how the clues are interpreted as part of the directions. A feature of this program that is available on other computers, but not on the Apple, is the ability to review the safari's directions while looking for clues. The combination of being able to review the directions and having a demonstration of how to interpret the clues would have eliminated many of the problems associated with students misinterpreting or failing to read the directions. When the students had realized that they did not understand, they could have gone back to the directions without having to quit the safari.

Another possibility would be for the program to ask questions about the possible locations of the animals when the students request a review of their clues. This could be in the form of allowing the students to make notes on the review screen. The students would be asked to mark all of the boxes where the animal could and could not possibly be located. If they were wrong, the program could prompt them to reexamine their clues for errors.

Think-Aloud Protocols

As expected, using think-aloud protocols with young students to determine which strategies they used to solve problems was difficult. During the think-aloud protocols, the researcher had to rely not just on what the students said, but on what they did. Notes taken for The King's Rule included the given sets and the sets the students tested. Notes taken for Safari

Search included the boxes the students opened and the order in which they were opened, using the record sheets provided in the teacher's guide.

This additional information was necessary because most of the students were very quiet. Only one or two in each group spoke loudly enough for the tape to easily record their voices. Most had trouble thinking aloud. Long periods of silence were common, punctuated by "um"s and "hum"s. Well over half of the statements made by students were made in response to questions.

Conclusions

While the strategies each piece of software claimed to teach were observed to varying degrees in all students, none of the students were observed using the optimal strategies described in the teacher's guides. The main cause of students not using the optimal strategy seems to lie in the construction of the software. The software does not require the students to use the desired strategies to succeed in the program. The students quickly found ways to be successful without using the optimal strategy.

A second cause of the difficulties experienced by the students in solving problems seemed to be the lack of explicit explanations of the surface and structural similarities and differences between the problems in different safaris and levels. Surface similarities existed in both pieces of software, from the way each problem was presented, to the feedback and clues received. However, there were many structural differences between the various safaris in Safari Search, as well as between and within the various levels of The Kings Rule. These similarities and differences were never pointed out or explained to the students. Students were left on their own to discover the similarities and differences.

While the discovery teaching method was intentionally used, it is unlikely that all students will learn the intended strategies without guidance designed to point out surface and structural similarities and differences, and the underlying principles associated with the problems. This guidance could come from revisions to the software similar to those suggested above, or from the teacher or other adult responsible for monitoring the software use.

Guidance could also come from other students. In this study, students in each group generally worked together, with the exception of BS. They were eager to help and share what they had learned. In both cases, students shared information actively, by helping and being helped, and passively, by listening as other students thought aloud while solving problems. This interaction was to be expected. Computers tend to encourage rather than discourage interaction between students (Flake, McClintock, & Turner, 1985). However, based on the fact that most of the students in this study were unable to use the optimal strategies, it is unlikely that the guidance provided by student interaction would have been sufficient.

Recommendations

Several questions remain unanswered by this study. Even though The King's Rule was based in a school subject, neither it nor Safari Search was designed to teach domain-specific knowledge. Would the presence of knowledge-related objectives enhance or deter the acquisition of problem solving abilities? If The King's Rule had been designed to teach math skills or if Safari Search had been designed to teach map skills in addition to problem solving strategies, would they have been more effective? It is recommended that similar studies be conducted to examine the effects of problem solving software designed to teach knowledge as well as problem solving strategies.

Both pieces of software allowed the students to succeed in the program without using the desired strategies. How would the results have been different if the students had only been allowed to succeed by following the desired strategies? It is recommended that similar studies be conducted to examine the effects of problem solving software designed to require the use of the desired problem solving strategies.

Neither piece of software provided learning guidance to assist the students to acquire the desired problem solving strategies. What would have been the effect of adding guidance to the software? If the students had received guidance in the form of the examples, explanations, or questions as described above, would the students have chosen to use the desired strategies and decreased their use of other, unintended strategies? It is recommended that similar studies be conducted to examine the effects of problem solving software that includes learning guidance for the desired problem solving strategies.

The role of the teacher in the use of problem solving software was outside the scope of this study, but is not a factor that can be ignored. With other types of software, it is up to the teacher to supply the missing instructional events (Wager, et al., 1989). This should be no less true with problem solving software. What happens to the effectiveness of problem solving software when a teacher provides missing instructional events, as well as explanations for the surface and structural similarities and differences between problems? Would the effectiveness of problem solving software be enhanced if the discovery methods used in the software were supplemented with learning guidance provided by the teacher? It is further recommended that the role played by the teacher in implementing problem solving software into the classroom be examined.

Implications

This study has provided a detailed picture of how students interact with problem solving software and how the nature of that interaction may vary from what was intended. By using techniques like those employed in this study, it is possible to find out more about how students actually use problem solving software and what skills, knowledge, and problem solving strategies result. Future research of this type should provide additional valuable clues for the design of effective problem solving software.

REFERENCES

Adams, M. J. (1989). Thinking skills curricula: Their promise and progress. Educational Psychologist, 24, 25-77.

Andre, T. (1986). Problem solving in education. In Phye, G. D. & Andre, T. (Eds.), Cognitive classroom learning (pp. 169-204). Orlando, FL: Academic Press, Inc.

Bransford, J., Sherwood, R., Vye, N., & Reiser, J. (1986). Teaching thinking and problem solving. American Psychologist, 41, 1078-1089.

Bransford, J. D., & Stein, B. S. (1984). The ideal problem solver. New York, NY: W. H. Freeman and Company.

Duffield, J. A. (1989). Problem solving software: What does it teach? Unpublished doctoral dissertation, The Florida State University.

Ericsson, K. A., & Simon, H. A. (1984). Protocol Analysis. Cambridge, MA: The MIT Press.

Flake, J. L., McClintock, C. E., & Turner, S. V. (1985). Fundamentals of computer education. Belmont, CA: Wadsworth Publishing Co.

Gagne, R. M. (1985). The conditions of learning (4th ed.). New York: Holt, Rinehart and Winston.

Gagne, R. M., & Brown, L. T. (1961). Some factors in the programing of conceptual learning. Journal of Experimental Psychology, 62, 313-321.

Gick, M. L. (1986). Problem-solving strategies. Educational Psychologist, 21, 99-120.

Greeno, J. G. (1980). Trends in the theory of knowledge for problem solving. In Tuma, D. T., & Reif, F. (Eds.), Problem solving and education: Issues in teaching and research (pp. 9-24). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kittell, J. E. (1957). An experimental study of the effect of external direction during learning on transfer and retention of principles. Journal of Educational Psychology, 48, 391-405.

Larkin, J. H. (1980). Teaching problem solving in physics: The psychological laboratory and the practical classroom. In Tuma, D. T., & Reif, F. (Eds.), Problem solving and education: Issues in teaching and research (pp. 25-38). Hillsdale, NJ: Lawrence Erlbaum Associates.

Mayer, R. E. (1987). Educational Psychology. Boston, MA: Little, Brown and Company.

Norton, P., & Resta, V. (1986). Investigating the impact of computer instruction on elementary students' reading achievement. Educational Technology, 26(3), 35-41.

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

O'Brien, T. C. (1983). The king's rule [Computer program]. Pleasantville, NY: Sunburst Communications, Inc.

O'Brien, T. C. (1985). Safari search [Computer program]. Pleasantville, NY: Sunburst Communications, Inc.

Rubenstein, M. F. (1980). A decade of experience in teaching an interdisciplinary problem-solving course. In Tuma, D. T., & Reif, F. (Eds.), Problem solving and education: Issues in teaching and research (pp. 25-38). Hillsdale, NJ: Lawrence Erlbaum Associates.

Wager, W. W., Wager, S. U., & Duffield, J. A. (1989). Computers in teaching. Cambridge, MA: Brookline.

Woodward, J., Carnine, D., & Gersten, R. (1988). Teaching problem solving through computer simulations. American Educational Research Journal, 25, 72-86.

Table 1
Type of Rules Presented at Each Level of The King's Rule

Level	Type of Rule	Example Rule	Example Set
1	Addition	+3	7, 10, 13
	Subtraction	-4	19, 15, 11
2	Multiplication	x2	3, 6, 12
	Division	÷3	27, 9, 3
3	Rules From Levels 1 or 2		
	Order	Low to High	2, 39, 173
	Equation	A x B = C	4, 3, 12
	Common Characteristic	Odd	13, 5, 27
		Divisible by 4	16, 8, 20
		End in 9	29, 49, 9
4	Equation	A + B + 4 = C	3, 5, 12
		(A + 1) x B = C	10, 2, 22
5	Rules similar to	Low to High	5, 10, 15
	those in other levels.[a]	A + B = C	10, 20, 30
6	All numbers may	2 #s Are Equal	4, 4, 16
	not be used.[a]	Sum Is Even	2, 4, 6

[a]The given sets are disguised so that they appear to follow at least one other rule.

Table 2
Type of Clues Presented in Each Game of Safari Search

Game	Type of Clues Given in Each Box
One-Animal Safaris	
1. Iguana[a]	Tells if it is in the box (Not Here).
2. Flamingo[a]	Tells if it shares an edge (Hot), a corner (Warm), or not touching (Cold).
3. Seal[a]	Tells how many boxes away it is (numbers 0-8).
4. Loon	Tells if it shares an edge or corner (Yes) or not (No).
5. Dragon	Tells how many it sees in that row and column (0, 1).
6. Donkey	Tells if it is above, below, left, or right (Yes, No).
Two-Animal Safaris	
1. Kittens	Tells how far away one kitten is (numbers 0-8).
2. Rhinos	Tells how far away both are (number pairs, 3:2, 1:6).
3. Snails	Tells how many it sees in that row and column (0,1,2).
4. Kangaroos	Tells if both are above, below, left, right (Yes, No).
5. Cats	Tells if any are above, below, left, right (Yes, No).
6. Llamas	Tells total distance to both (numbers 0-12).

[a]Students are immediately informed when box with animal is opened.

Table 3

Results Summary for The King's Rule

Anticipated outcomes	Evidence
Domain-general strategy	
Hypothesis testing	*Observations*: All students tested hypotheses. Half tested multiple hypotheses. One tested a set expected to be negative. None tested sets to eliminate competing hypotheses. *Think-aloud protocols*: All students tested hypotheses. Four tested multiple hypotheses. One tested a set expected to be negative. None tested sets to eliminate competing hypotheses. *Tests*: Did not examine this outcome.
Domain-specific strategy	
Recognizing patterns and relationships	*Observations*: All recognized some patterns and relationships. Only two looked for a wide variety. *Think-aloud protocols*: All recognized level 1 relationships by second protocol. Half could look for level 3 relationships, only one found them. *Tests*: Did not examine this outcome.
Near and Far Transfer	*Observations*: Did not examine this outcome. *Think-aloud protocols*: Did not examine this outcome. *Tests*: No supporting evidence was found.

Table 4

Results Summary for Safari Search

Anticipated outcomes	Evidence
Domain-General Strategies	
Making inferences	*Observations*: All located some animals, none located all animals. Many clues were ignored or misinterpreted. *Think-aloud protocols*: All were able to solve puzzles. Performance improved in second protocol. Most solved harder puzzles than in first protocol. *Tests*: Did not examine this outcome.
Inventing tactics	*Observations*: All used tactics. Difficult to determine who invented them. Target clues were found for half the safaris. *Think-aloud protocols*: All used tactics. Only two could explain why they worked. More tactics were used in the second protocol. *Tests*: Did not examine this outcome.
Collecting, organizing, and using data	*Observations*: All collected and used data. Most ignored negative data. Few understood meaning of clues. Few used more than last one or two clues. *Think-aloud protocols*: All collected and used data. Only two came close to using optimal strategy. *Tests*: Did not examine this outcome.
Near and Far Transfer	*Observations*: Did not examine this outcome. *Think-aloud protocols*: Did not examine this outcome. *Tests*: No supporting evidence was found.