DEVELOPING TECHNIQUES FOR MEASURING AND ENHANCING STUDENTS’ COGNITIVE AND METACOGNITIVE SKILLS

Wayne J. Staats\textsuperscript{1}, David Dean\textsuperscript{2}, Don Miles\textsuperscript{3}, Toni Blum\textsuperscript{4}, Ssu-yu Lu\textsuperscript{5}

Abstract - While previous work established that some self-report measures of metacognition correlate highly with programming and other CS skills \cite{2,3}, the current pilot project seeks to develop a behavioral task measure of metacognition and to use exercises modeled on this behavioral task to teach and enhance metacognitive skills. These exercises, based on work by Deanna Kuhn at Columbia University, are to be incorporated in an undergraduate computer science program for the purpose of enhancing cognitive and metacognitive skills that have been identified as important for computer scientists \cite{2,3}.

First, the pilot study established a baseline comparison between the MSI \cite{2,3,18} and one of Kuhn’s most widely used causal inferencing tasks \cite{10,11}, dubbed the “boat-races” (BOATS) task. Second, the pilot study established a baseline comparison between the BOATS model task and several domain-specific tasks developed by the researchers; these domain specific tasks were modeled after the ones developed by Kuhn. Regression analyses were performed, using the BOATS (domain independent) task, the MSI \cite{2,3} scores, and the MCSS [19] scores as predictor variables and performance on the newly created domain-specific tasks as criterion variables. While none of the individual component variables was found to be predictive of exercise task performance, the overall model was significant for at least two of the three domain-specific tasks. Continued research will use these task scores as predictors and Computer Science task performance as criterion variables.

Should the exercises prove predictive, they will be used as training tools to develop metacognitive skill.

INTRODUCTION

The authors present the results of a pilot study that is part of a long running project \cite{2,3} intended to help retain more students in computer science. The high attrition rate of computer science (CS) students during the freshman year is a persistent problem the authors have observed in different university and college CS programs. It is not far fetched to assume that this problem is not confined only to these programs, but exists nationally. To address the retention issue during the freshman year, the principle authors have begun to develop a set of exercises to augment existing introductory courses. These exercises focus on improving the cognitive and metacognitive skills of CS students.

At the College Of Charleston, it is expected that the closed lab associated with CS1 will be redesigned around these exercises. Currently, the students complete several short programming exercises during the lab that reinforce the material covered in lecture. Once a complete set of exercises are developed (1 exercise for each week of the semester), the lab will consist of working through the exercises followed by actually designing and implementing the problem in the exercise. In this fashion, the students will be taught a cognitive or metacognitive skill along with reinforcing the lecture material. By improving these skills, it is expected that students’ critical thinking abilities will increase, leading to better performance in the CS program and lowered attrition rate.

The authors present the results of a pilot study utilizing newly developed causal inferencing tasks, based on similar tasks developed by Deanna Kuhn et al. \cite{10,11}. Although the exercises are ultimately intended as developmental tools, the pilot study was conducted to determine if the exercises tapped into the metacognitive skills in a manner consistent with the authors’ expectations. These specific tasks were chosen because of the importance of causal inference in the software design process (which of these variables will affect the efficiency of my program?) and because of the importance of controlled comparisons in the program debugging process (can the failure of my program be explained by variations in this variable or in that one?). The authors developed exercises in two domains, computer science and psychology, to explore the generalizability of the approach to various scientific fields. The authors compared the scores of the pilot task exercises to those of other previously established measurement tools.

BACKGROUND

Educational researchers and practitioners have shifted their focus of late – academics and teachers are now equally likely to criticize the reliance of the educational system on “objective” standardized tests of factual knowledge and
support “education for understanding” [7]. This rare point of agreement continues to fuel interest in education for “critical thinking,” a phrase oft-cited as a goal of education at all levels though with an elusive definition. The difficulty in teaching any skill lies not in using that skill within the original context of training but in exercising the skill in other settings once support is withdrawn. This difficulty leads to performance failures that seem to suggest a lack of the metacognitive understanding necessary for skill use in all applicable contexts.

Similarly, the educational computing literature is replete with evidence that students may acquire a basic understanding of the grammar and features of a programming language, yet fail to achieve the desired level of performance in introductory programming courses [1, 15, 17]. Researchers argue that students must move beyond this syntactic level of knowledge to achieve conceptual and strategic-level knowledge [17, 23]. It is not enough to gain the procedural skills — students must also understand when and how to apply them. Studies of expertise, transfer, and other related areas in cognition have made the same determination [4,5].

Conceptual-level knowledge involves basic cognitive skills, such as grasping the principles that govern the actions executed by a program and creating a mental model of the system being implemented. These are skills used in program design. Strategic-level knowledge involves metacognitive skills, such as the acquisition of more flexible problem-solving techniques considered critical to expertise. These metacognitive skills are necessary to solve novel programming problems, test programs, and debug logic errors [17]. The authors argue, along with several other CS educators [14,15,17,23], that most introductory programming courses do not place enough emphasis upon understanding the basic concepts being used or the strategic skills needed to utilize these concepts.

In order to explore the extent to which these strategic-level skills are related to problem-solving expertise in diverse settings, the Metacognitive Skills Inventory (MSI) was developed by Blum and Staats [2,3,21,22]. Despite the promise of this assessment research, it does not reveal where or how these critical metacognitive skills are developed. Recent work in cognitive development has focused on the process by which the procedural skills and understanding necessary to use them effectively develops. In an effort to explain the process by which this strategy acquisition takes place, Kuhn et al. have developed and used several instruments that allow this study. These instruments allow students to evaluate the outcomes of a four- or five-variable system and systematically determine the causal structure, or mental model, of the system. Students are presented with introductory information about the particular paradigm in question including descriptions of each dichotomous variable. Cases are then presented where each variable is assigned a particular level. Individual cases can differ on one or more variable levels, however, conclusions can only be drawn when cases for comparison differ on only one variable, specifically the variable of interest.

Students using these instruments must recognize the need to isolate individual variables and use controlled comparison to successfully complete the task involved. Using these strategies, students are able to create a mental model of the casual structure of the system. Kuhn has used several different paradigms to investigate this developmental process, including boat races [11], floods, earthquakes, and avalanches [10]. It was the boat example upon which the computer science exercises were modeled.

**THE MODEL TASK**

The boat races task (BOATS), which was modeled to create the exercises used in this study, is described in further detail here. The task uses four variables that dictate the performance of the boat during races. The four variables are the “Shape of Boat” (wide versus narrow), the “Size of Sail” (large versus small), the “Material” of the hull (wood versus metal), and the “Depth of Water” (deep versus shallow). With these four variables, there are sixteen different configurations for a particular boat. A race consists of two boat configurations racing on the same course (depth of water is consistent for both boats). From the results of the each race, the boats are ranked as first through fourth class, which allowed students to make comparisons and possibly make inferences regarding the causal nature of each variable. In other words, students compare each case with the outcomes and try to determine whether or not each variable made significant contributions in the final outcome, based on those comparisons. Once eight total cases are presented (two presented initially, then two more on each following page), students have the opportunity to summarize what they have learned about which variables matter and which do not in the outcomes of the boat time trials. Further, they are asked to make predictions about new cases (not previously presented for study).

The three computer science exercises that were developed for the pilot study were modeled after the BOATS example already described. The authors attempted to create these exercises as direct analogies of the BOATS example, that is, each exercise presented the student with scenarios that related to the runtime performance of a given program implementation. The four variables that were selected included the “Ordering of Input Data” (sorted versus random), the “Size of Input Data” (large versus small), the “Data Structure” (binary tree versus linked list, hash table versus linked list, and hash table versus graph), and “Algorithm” (iterative versus recursive). These four variables were used to describe different types of implementations of the same problem. The descriptions of the three problems are as follows:
“The IS department of a university is asked to implement a system that calculates student GPAs given a list of all the classes a student has taken. The design considerations are how the initial data is inputted into the system, the size of the data, how the data is represented, and the search algorithm used to traverse the data structure. The goal is to come up with a system that has good performance, where good performance is defined as being fast as well as memory efficient. Performance will be ranked as 1 (fast), 2, 3, or 4 (slow).”

“A company has, as their flagship product, a word editor application. To make their product more competitive in the market, they intend to add a dictionary component to the word editor application for spell checking and looking up word definitions. The design considerations are how the initial data is inputted into the system, the size of the data, how the data is represented, and the search algorithm used to traverse the data structure. The goal is to come up with a system that has good performance, where good performance is defined as being fast as well as memory efficient. Performance will be ranked as 1 (fast), 2, 3, or 4 (slow).”

“Determining the shortest path between two cities is an important feature of a road trip planner application. Given the city of origin, the road trip planner wants to find the shortest route to the destination city. The design considerations are how the initial data is inputted into the system, the size of the data, how the data is represented, and the search algorithm used to traverse the data structure. The goal is to come up with a system that has good performance, where good performance is defined as being fast as well as memory efficient. Performance will be ranked as 1 (fast), 2, 3, or 4 (slow).”

Kuhn has found some discrepancies between adolescents and adults in their abilities to successfully perform these types of tasks. However, even adults seem to have some difficulty using the right strategies to successfully complete these tasks [9]. This suggests that there is much more than the procedural skills themselves that need to be developed to enable students to construct new knowledge. As opposed to routine cognitive development prior to adolescence, the more complex processes and structures that are described here are not necessarily realized. Kuhn has made frequent use of this principle to develop techniques for training interventions to increase metacognitive awareness and understanding.

**METHOD**

While countless researchers make the case for including assessment and training of metacognition in the sciences and other educational arenas [16,20], relatively few have developed effective strategies for accomplishing these tasks in anything but a very general way. The current study was designed as a pilot to determine the effectiveness of the exercises to measure metacognitive skills. The goal of the study was to help determine if the exercises could actually be used to build metacognitive skills as part of an introductory programming course.

**Participants**

Twenty-four participants in the computer science operating systems course and fifteen participants from the Research Methods in Psychology class completed all phases of the study. Participants for the overall study ranged in age from 19 to 37, with a mean age of 23.9. The mean age of the CS group was 25.9, while the mean age of the psychology group was 20.5. There were 22 males and 18 females. However, the composition varied across the two groups, with the CS group predominately male (79%), while the psychology group was predominately female (87%). The samples were predominately white (92.3%), and ranged in education from sophomores to seniors. Approximately 71% of the CS sample were employed, primarily part time, while 33% of the psychology sample were employed, all part time.

**Materials**

In order to assess metacognitive awareness and skills, the MSI [2,3] was utilized. The inventory assesses the extent to which people have confidence in their problem-solving ability (Confidence subscale), as well as their awareness of their own planning and evaluation processes during problem solving (Decomposition subscale). In order to utilize another measure for assessing metacognitive strategies, a 12-item scale called the Metacognitive Computer Skills Scale (MCSS) was used [19]. This scale assesses various approaches to problem solving strategies as well as the trainee’s awareness of these strategies and how well the strategies seemed to be working. These items are answered using a five point Likert-style scale, with a response of ‘1’ indicating "rarely true of me" and a response of ‘5’ indicating "almost always true of me." A sample item is: "When I approach any problem, I try to break the problem down into smaller parts." The coefficient alpha for this scale is .92.

In order to assess general cognitive ability of the participants, the Shipley Institute of Living Scale was utilized. This test is a general cognitive ability test designed as a general screening instrument that provides an accurate estimate of intellectual functioning [24]. The scale consists of two subscales, a 40-item vocabulary test and a 20-item abstract-thinking test. Split-half reliabilities for the scales are .87 for vocabulary, .89 for abstraction, and .92 for the entire scale. Test scores from the combined test can be used to estimate full-scale IQ scores, with a median correlation of .79 with scores on the Wais Full Scale across 11 validation samples.

In addition to these self-report measures, one of the causal inference tasks developed by Kuhn [10] was used to attain a behavioral measure of cognitive ability. For the...
purposes of this pilot study, the BOATS example was used. Both the number of valid inferences on the exercise and the number of correct predictions on the final analysis page were measured. The computer science domain-specific stimulus materials that were developed are based on the general design of these BOATS example as already described.

The problems dealing with research methodology and statistical problems consist of word problems containing multiple components that have an effect on the choice of an appropriate analytic technique. As stated previously, components include identification of the independent variable and dependent variable, the scales of measurement for each variable, the research design, and whether testing of mean differences is appropriate. The participants select the best analytic technique based upon the specific combination of those components presented within each scenario. Five exercises were developed, utilizing analytic techniques including correlated samples t test, independent samples t test, 1-way ANOVA, and correlation. Scores were obtained on the correct identification of the components, as well as on the correct choice of the statistical technique for the five problems. An aggregate score was obtained by summing the scores from the five problems.

**Procedure**

Preliminary data were gathered using a demographic survey, the MSI and the MCSS, and the Shipley Institute of Living Scale. Next, the participants were presented with the problem solving exercises. The first exercise that was administered was the BOATS exercise. Following completion of the BOATS exercise, the domain-specific problem-solving exercises were administered (either CS or Psychology).

**RESULTS**

Means and standard deviations for key variables can be found in Table 1. Preliminary analyses were conducted to compare the two samples (CS and Psychology) regarding cognitive abilities and metacognitive skills. There were no differences on scores of the cognitive ability test, nor for the two subscales (verbal and abstraction). There were also no differences between the samples on their scores for either of the measures of metacognition, the MSI and MCSS.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Computer Science</th>
<th>Psychology</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N = 24</td>
<td>N = 15</td>
</tr>
<tr>
<td>Pre-MSI</td>
<td>M 136.17</td>
<td>133.40</td>
</tr>
<tr>
<td></td>
<td>SD 14.76</td>
<td>13.16</td>
</tr>
<tr>
<td>Post-MSI</td>
<td>M 137.29</td>
<td>131.33</td>
</tr>
<tr>
<td></td>
<td>SD 16.45</td>
<td>13.04</td>
</tr>
<tr>
<td>Pre-Confidence</td>
<td>M 25.00</td>
<td>23.73</td>
</tr>
<tr>
<td></td>
<td>SD 3.59</td>
<td>2.55</td>
</tr>
<tr>
<td>Post-Confidence</td>
<td>M 25.04</td>
<td>23.00</td>
</tr>
<tr>
<td></td>
<td>SD 3.86</td>
<td>2.85</td>
</tr>
<tr>
<td>Pre-Decomp</td>
<td>M 72.58</td>
<td>71.33</td>
</tr>
<tr>
<td></td>
<td>SD 9.03</td>
<td>7.76</td>
</tr>
<tr>
<td>Post-Decomp</td>
<td>M 72.88</td>
<td>70.07</td>
</tr>
<tr>
<td></td>
<td>SD 8.87</td>
<td>8.40</td>
</tr>
<tr>
<td>Shipley</td>
<td>M 117.29</td>
<td>116.23</td>
</tr>
<tr>
<td></td>
<td>SD 5.09</td>
<td>4.02</td>
</tr>
<tr>
<td>Pre-MCSS</td>
<td>M 42.50</td>
<td>41.27</td>
</tr>
<tr>
<td></td>
<td>SD 16.44</td>
<td>13.05</td>
</tr>
<tr>
<td>Post-MCSS</td>
<td>M 43.42</td>
<td>39.07</td>
</tr>
<tr>
<td></td>
<td>SD 9.78</td>
<td>6.40</td>
</tr>
<tr>
<td>Boat Total</td>
<td>M 13.38</td>
<td>9.13</td>
</tr>
<tr>
<td></td>
<td>SD 4.31</td>
<td>2.13</td>
</tr>
<tr>
<td>CS Exercise</td>
<td>M 7.54</td>
<td>3.98</td>
</tr>
<tr>
<td>(Max = 20)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RM Exercise</td>
<td>M 38.93</td>
<td>7.16</td>
</tr>
<tr>
<td>(Max = 65)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Shipley scores for 2 Participants from psychology group were unavailable.

**Differences**

An independent samples t test was conducted to compare scores of the two groups on the boat exercise. There was a significant difference, with \( t(37) = 3.54, p < .01 \), with the mean number of correct inferences higher for CS students, \( M = 13.38 \) compared to \( M = 9.13 \) for psychology students. In order to investigate whether the various metacognitive skills scales are predictive of cognitive performance, a series of regression analyses were conducted

**Regression Models**

In order to investigate whether the various metacognitive skills scales were predictive of cognitive performance, a series of regression analyses were conducted. The first regression model tested utilized the BOATS exercise score as the criterion, with four predictor variables entered as a block. Using the scores on the Shipley, as well as pre-exercise scores on the decomposition and confidence scales of the MSI and the pre-exercise scores on the MCSS, the regression approached significance, with \( F(4,32) = 2.57, p = .057 \), with adjusted \( R^2 = 15\% \).
The second regression model that was tested dealt with the computer science students. For these students, the combined score for the “word editor” and “path between cities” exercises was used as the criterion. When scores on the boat exercises, pre-exercise scores of the confidence subscale and the MCSS scale were entered as predictors, the overall regression was significant, with $F(3,20) = 3.36$, $p<.05$. None of the individual predictor variables created statistically significant coefficients. This model explained approximately 24% of the variance.

When regression analyses were conducted on the two problems separately, slightly different and more complex results were found. Using the results from the problem dealing with addition of a dictionary component to a word editor as the criterion, scores on the boat exercise, pre-exercise scores on the decomposition and confidence scales of the MSI and the pre-exercise scores on the MCSS were entered as predictors. The overall regression was significant, with $F(4,19) = 2.86$, $p<.05$. None of the individual predictors created statistically significant coefficients. This model explained approximately 25% of the variance.

When scores for the road trip planner exercise were used as the criterion, scores on prediction portion of the boat exercise and pre-exercise scores on the decomposition scale of the MSI were entered as predictors. This model was significant, with $F(2,21) = 3.95$, $p<.05$. Again, none of the individual predictors resulted in significant regression coefficients, although the prediction portion of the boat exercise approached significance. This model explained approximately 20% of the variance.

For the psychology students, the aggregate score on the exercises was used as the criterion, with scores on the boat exercises, pre-exercise scores of both the confidence and decomposition subscales of the MSI and the MCSS scale entered as predictors. The overall regression model was significant, $F(4,10) = 7.82$, $p<.01$. For this model, approximately 69% of the variance was explained, and both Shipley scores and decomposition scores resulted in significant regression coefficients within the overall model.

**DISCUSSION**

Differences between the CS and Psychology students on the BOATS task parallel the difference found between CS and non-CS students on MSI scores in previous studies. The findings provide further evidence that the systematic decomposition of problem structure is a crucial component of computer science skill. Both self-reports of these skills and behavioral measures of these skills show similar trends. Still, these data are clouded by gender differences, making the distinction less clear. An expanded sample is needed to test the full model and include gender as a factor in the analysis.

The first regression analysis was intended to explore the value of including a behavior-based assessment for metacognition in addition to our current self-report measures. The model revealed that, while the MSI and MCSS accounted for some of the variability in problem-performance, the BOATS exercise tapped into a new dimension of metacognitive skill and should be included as a separate predictor variable in future models. Future research should assess the extent to which performance on this task predicts computer science course performance and/or programming ability.

Another intention of the pilot study was to test the utility of the newly developed domain-specific exercises. The intent was to verify that some variables previously known to predict performance levels in Computer Science and Science education are valuable predictors of problem-solving skill in the domain-specific problems. The second regression analysis suggested that the aggregate of the predictor variables accounted for much of the variance, but none of the individual predictors (BOATS, MSI, MCSS) were useful alone. However, when the two CS exercises were modeled separately, somewhat more complex relationships are found. These findings suggest that the problem dealing with addition of a dictionary component to a word editor is more directly parallel to the BOATS example and students were better able to transfer skills from one to the other. However, the road trip planner exercise may have tapped more heavily into the prediction component in the boat exercise, as well as activating more of the student's decomposition skills. Clearly, additional training in the component skills gleaned from each task could enhance transfer to new domains. Future studies will focus on such techniques.

Ultimately, the use of these tasks as predictors cannot be established without comparing these scores to actual programming performance and course performance in CS classes. These data analyses are underway and the authors hope to report them by the time of the November conference. If they prove predictive, then the domain-specific tasks can be used as training exercises, rather than simply measuring tools.

The final regression analysis, focusing upon the Psychology Research Methods example, revealed that the assessment variables were an overwhelming success in predicting performance upon the problem-solving task in this domain. In particular, the MSI decomposition scale and the Shipley scale were effective predictors of performance. These exercises may have relied upon greater levels of verbal ability in addition to the problem solving skills. Although conceptually similar to the BOATS exercises, the format for these exercises is somewhat different. It is difficult to determine the extent to which these exercises can be used in training of metacognition until a more thorough longitudinal study is completed.
During a future longitudinal study, to be conducted once the entire computer science lab course is designed, the authors intend to break the problem sets down in class and discuss each component, particularly the reasons for including or excluding a particular variable in a causal inference. The authors hope to appropriately model the cognitive processing steps needed in this type of task. It is hoped that the development of models for decomposing the task will lead to appropriate decomposition skills in later tasks. The lack of change from pretest to post test of any of the assessment scales is clear evidence that students who have not appropriately parsed the problem from the beginning will not learn to do so via mere repetition. Indeed, the transfer of these skills from one problem to the next seems unlikely to occur without explicit training.

It will be critical to explore not only the valid inferences reached by the students completing the exercise, but also the reasons why each inference (valid or invalid) was made. The analysis of these additional data may provide more insight into the metacognitive skills and abilities of the students than does the preliminary data.

ACKNOWLEDGMENT

We would like to thank Tameika Jackson and Phil Balem, students at the College Of Charleston, for their work in data collection and in the research process. We would also like to thank the psychology and computer science students involved in the pilot study discussed in this paper.

REFERENCES

