Using Subgoal Learning and Self-Explanation to Improve Programming Education

Lauren E. Margulieux (l.marg@gatech.edu)
Georgia Institute of Technology, School of Psychology
Atlanta, GA 30332-0170 USA

Richard Catrambone (richard.catrambone@psychology.gatech.edu)
Georgia Institute of Technology, School of Psychology
Atlanta, GA 30332-0170 USA

Abstract

The present study explored passive, active, and constructive methods of learning problem solving procedures. Using subgoal learning, which has promoted retention and transfer in procedural domains, the study compared the efficacy of different methods for learning a programming procedure. The results suggest that constructive methods produced better problem solving performance than passive or active methods. The amount of instructional support that learners received in the three different constructive interventions also affected performance. Learners performed best when they either received hints about the subgoals of the procedure or received feedback on the subgoal labels that they constructed, but not when they received both. These findings suggest that in some cases constructing subgoal labels is better than passively or actively engaging with subgoal labels. There is an optimal level of instructional support for students engaging in constructive learning and that providing too much support can be equally as detrimental as providing too little support.

Keywords: subgoal learning; self-explanation; worked examples; computing education.

Introduction

Students in higher education need to be able to learn independently, at least in part. As the number of students pursuing bachelor's and advanced degrees increases, so does the ratio of students to instructors and the number of online courses. These factors make direct interaction between students and instructions increasingly limited and self-guided learning increasingly valuable. To help students be more independent learners, support from researchers and instructional designers is needed. The present research examined a new strategy to support independent learning: the integration of subgoal learning and self-explanation.

Subgoal Learning

Subgoal learning refers to a strategy used predominantly in STEM fields that helps students to deconstruct problem solving procedures into subgoals to better recognize the structural components of the problem solving process (Atkinson, Catrambone, & Merrill, 2003; Catrambone, 1998). Subgoals are functional pieces of procedures used to solve problems that contain one or more individual steps, such as solving for a variable in a calculus problem.

Research suggests that when instructions help students learn the subgoals of a procedure, students are better able to transfer knowledge to solve novel problems. Catrambone and Holyoak (1990) found that when instructional materials highlighted the subgoals of a procedure, learners were more likely to correctly apply it to problems that used the same procedure but had different contextual features (e.g., problems about birthdays versus those about football) or had modified or new steps. Subsequent studies (Catrambone, 1994, 1996, 1998; Margulieux & Catrambone, 2014; Margulieux, Guzdial, & Catrambone, 2012) have consistently found that subgoal-oriented instructions improved problem solving performance across a variety of STEM domains.

Subgoal-oriented instructions are typically implemented as worked examples. Worked examples give learners concrete examples of the procedure being used to solve a problem. Because problems necessarily include a context, such as birthdays or football, worked examples include context-specific information. Eiriksdottir and Catrambone (2011) argued that, when studying examples, learners tend to focus on superficial features rather than the structural features because superficial features are easier to grasp and novices do not have the necessary domain knowledge to recognize the structural features of examples (Chi, Bassok, Lewis, Reimann, & Glaser, 1989). A focus on superficial features leads to ineffective organization and storage of information that, in turn, leads to ineffective recall and transfer (Bransford, Brown, & Cocking, 2000).

To promote deeper processing of worked examples and, thus, improve retention and transfer, worked examples have been manipulated to promote subgoal learning. Subgoal labeling is a technique used to promote subgoal learning that has been used to help learners recognize the structural structure of the procedure being exemplified in worked examples (e.g., Catrambone, 1994, 1996, 1998). Subgoal labels are function-based instructional explanations that describe the purpose of a subgoal to the learner.

Catrambone (1998) found that learners who received labels that were abstract (e.g., Ω) and had greater prior knowledge performed better than those who received labels that were context-specific (e.g., isolate x) on problem solving tasks that were given after a week-long delay or that required using the procedure differently than demonstrated in the examples. Catrambone (1998) argued that learners with sufficient prior knowledge were able to correctly explain to themselves the purpose of the subgoal. He argued that prompting self-explanation of the subgoal by providing a label that did not explain the subgoal's function was more effective than providing an informative label.

The findings from Catrambone's (1998) research align with a growing body of evidence that learning is more effective when students actively or constructively engage with content rather than passively receive content. This body of evidence is summarized by Chi (2009) and used to support her Interactive-Constructive-Active-Passive (ICAP) framework. In this framework, Chi (2009) characterized four types of learning based on students' engagement with content: interactive, constructive, active, and passive (see Figure 1 for definitions and examples).

Passive	**Active**	**Constructive**	**Interactive**
Receiving information without physical activity (e.g., listen to a lecture)	Receiving information with physical activity (e.g., take notes on a lecture)	Individually producing information beyond that which is provided (e.g., connect concepts to prior knowledge)	Collaboratively producing information beyond that which is provided (e.g., discuss concepts)

Figure 1. Definitions of passive, active, constructive, and interactive learning based on the ICAP framework (Chi, 2009).

Using this framework to compare the learning outcomes from various learning activities, Chi (2009) found that interactive and constructive learning were the most effective, active learning was the second most effective, and passive learning was the least effective. Most research about subgoal learning, besides Catrambone (1998), has provided meaningful subgoal labels that explain the function of subgoals to learners. Providing labels to learners promotes passive learning, which is the least effective method of learning. The present study explored whether more engaging methods of learning subgoals, such as self-explanation of the subgoals of a procedure, would improve novel problem solving.

Present Study

The present study prompted participants to learn the subgoals of a procedure through a worked example that either encouraged passive, active, or constructive learning. The problem solving domain for the present study was programming. Because participants were novices, the present study used a drag-and-drop programming language to teach programming concepts. Drag-and-drop programming languages are more easily understood by novice learners because they can select and drag pieces of code from a menu, which does not require learning the syntax and semantics of a programming language (Hundhausen, Farley, & Brown, 2009). The programming language used in the present study was Android App Inventor, which is used to create applications (apps) for Android devices. Participants used App Inventor to create an app that has buttons that play sounds when pressed (see Figure 2 for excerpt).

The subgoals of the procedure were identified using the Task Analysis by Problem Solving (TAPS) procedure (Catrambone et al., 2016) that has been used in prior research (e.g., Margulieux & Catrambone, 2014). In the passive learning condition, participants were given subgoal labels created by the experimenters, as is conventional in prior subgoal research (e.g., Catrambone, 1998). These subgoal labels will also be created through the TAPS procedure (Catrambone et al., 2015).

In the active learning condition, participants were given the worked example grouped by subgoals and asked to select a subgoal label from a list of labels that matched the purpose of the group. The list contained only labels that were viable options, meaning the list did not include distractor items. This active method of self-explaining was equivalent to the active self-explanation methods used by Aleven and Koedinger (2002) and Conati and VanLehn (2000). The method matches Chi's (2009) definition of active learning as a method that requires activity from the learner but not construction of new information.

In the constructive learning conditions, participants were asked to create their own subgoal labels to explain the subgoals of the procedure. To train participants to construct their own subgoal labels, they were given subgoal label training. Only the constructive groups received this training. The passive and active groups received a comparable task: analogy training. Training for analogies (e.g., water : thirst :: food : hunger) was considered equivalent because both analogies and subgoal labeling requires people to consider the underlying relationship between words and come up with a new word that describes that relationship.

The three constructive learning conditions prompted participants to construct their own subgoal labels. They differed on the amount of guidance that participants received while constructing labels. In the guided constructive conditions, participants were given the worked example with the solution steps grouped by subgoal, and the example indicated which subgoals achieved the same functions. For instance, all of the subgoals denoted as "Label 1" achieve the same function though the contexts are different (see Figure 2). In the guided constructive with hints condition, participants were given hints about the similarities among different instances of the same subgoal. In the guided constructive without hints condition, participant did not receive these hints. In the unguided constructive condition, participants received a worked example that did not indicate which steps belonged to which subgoals. Participants in this condition had to identify the subgoals for themselves and create labels for them.

The amount of guidance that participants received during instruction differed based on whether they received feedback. Some studies in the self-explanation literature have found that feedback supports self-explanation because it reinforces correct explanations and reduces floundering

(Aleven & Koedinger, 2002; Conati & VanLehn, 2000). Conversely, other studies have found that feedback creates overreliance on instructional information provided via feedback and, overall, hinders self-explanation (Schworm & Renkl, 2006). Based on this conflicting evidence, feedback was considered an important feature to vary in the present study. Instructions for participants who received feedback had another copy of the worked example that included subgoal labels created by the experimenters. For the passive condition, this copy was exactly the same as the initial worked example. For the active and constructive conditions, the copied example with experimenter-created subgoal labels provided feedback to the participants about whether they selected the correct labels or created similar labels. Participants who received feedback were asked to compare their labels to those created by the experimenter to prompt them to reflect on the similarities or differences between the two. Instructions for participants who did not receive feedback included only the worked example with the passive, active, or constructive interventions. These participants were asked to re-read the example to make time on task more similar to that of participants who received feedback. The exception was that participants in the passive and no feedback condition were not asked to re-read the example to make their experience different from those in the passive with feedback condition. Due to this difference, the time on task was different, providing some insight into whether time on task affects performance for this task.

Given Labels (Passive)	Placeholder for Label (Active and Constructive)
Problem: Create an app that plays a drum sound when the image of a drum is touched. Handle Event Click on "My Blocks" to see the blocks for components created Click on "clap“ Drag out a when clap.Touched block Set Output Click on “clapSound” Drag out call clapSound.Play Connect it after when clap.Touched	Problem: Create an app that plays a drum sound when the image of a drum is touched. Label 1: __________ Click on "My Blocks" to see the blocks for components created Click on "clap“ Drag out a when clap.Touched block Label 2: _________ Click on “clapSound” Drag out call clapSound.Play Connect it after when clap.Touched

Figure 2. Worked example formatted with given labels or placeholders for labels.

Because the worked example was long, participants received only one worked example. Giving one worked example provided a unique opportunity to ensure that participants in the feedback condition did not overly rely on feedback. Participants were not told that they would receive feedback until they completed the task, meaning that they did not know to expect feedback.

The guidance provided by feedback was expected to interact with subgoal learning method. Withholding feedback can lead to incorrect explanations and floundering, but giving feedback can hinder self-explanation and lead to overreliance on feedback (Renkl, 2002; Schworm & Renkl, 2006). Learners making self-explanations can flounder because self-explanation, especially constructive explanations, requires some insight, meaning that learners have to recognize connections between pieces of information that are not necessarily apparent from the instructions (Wylie & Chi, 2014). Durso, Rea, and Dayton (1994) found that insight resulted from mental restructuring of knowledge that made connections between previously disjointed pieces of information. Durso et al. (1994) also found that if participants were given the solution to the problem at hand, mental restructuring did not occur. Durso et al.’s (1994) findings can explain why receiving information that could have been constructed through self-explanation does not allow for mental restructuring. Therefore, extra guidance from feedback on self-explanations was not always expected to lead to better learning outcomes, especially when learners received high levels of guidance during self-explanation.

Method

Participants

Each of the 10 conditions had 20 participants (N = 200). Participants were students at a mid-sized, southeastern, technical institute. Participants were required to have no experience with App Inventor and could not have taken more than one computer science or programming course. They also completed a multiple-choice pre-test to ensure that they did not have prior knowledge of the procedure. The majority of participants (91%) scored a zero on the pre-test, and no participants scored higher than one point. Participants completed a demographic questionnaire. Demographic variables and pre-test scores were analyzed to determine that groups were equivalent, but this analysis will not be reported due to space limitations.

Design

The experiment was five-by-two factorial, between-subjects design: subgoal learning method (passive, active, guided constructive with hints, guided constructive without hints, or unguided constructive) was crossed with feedback (no feedback or feedback). Dependent measures that are included in this paper were performance on the problem solving tasks and the subgoal labels that participants construct. Other measures are not discussed in this paper due to space restrictions.

Procedure

Sessions took between 80 and 110 minutes, depending on how quickly participants complete each of the tasks. First, participants completed the demographic questionnaire and pre-test. Then participants started the instructional period, which took 40 to 55 minutes. All manipulations occurred within the instructional period.

The instructional period started with an overview video of the App Inventor interface that was the same across all participants. The video did not include information about the procedure being taught, but it was intended to help participants familiarize themselves with the problem solving space in which they would be working. After the introductory video, participants received either subgoal label or analogy training. Next, participants received the worked example. The worked example listed the steps taken to create a Music Maker app that plays musical sounds when images of musical instruments are pressed or the device is shaken. The format of the worked example depended on participants' assigned method of subgoal learning. The passive method gave participants subgoal labels (see passive condition in Figure 2), and the other methods gave participants spaces to fill in subgoal labels (see constructive condition in Figure 2), except for the unguided constructive condition, which had only the listed steps. For the active method, participants had a word bank with labels that they could select. In the guided constructive with hints condition, additional text highlighted similarities between all subgoals called "Label 1." This guidance was given for each subgoal.

When participants finished the first pass through the worked example, they were either prompted to re-read the example for the no feedback condition, or they were given the worked example with the experimenter-created subgoal labels for the feedback condition. Participants in the feedback condition were told that the subgoal labels in the second copy of the worked example were created by subgoal label experts. Then they were asked to compare the labels that they made or selected to those given in the second example. To ensure that participants paid attention to the worked example and could complete tasks in the App Inventor interface, they were asked to complete practice problems before finishing the instructional period.

Following the instructional period, participants completed problem solving tasks that measured learning. During this assessment period, participant did not have access to the instructional materials. They were told of this restriction at the beginning of the session. The problem solving tasks asked participants to modify or add components to their Music Maker app. Of the five tasks, two required contextual transfer from the worked example, meaning that the superficial features of the app components were different (e.g., exchange a drum sound for a cymbal sound) but the procedural steps used to create them were the same. The remaining three tasks required procedural transfer from the worked example, meaning that the individual steps used to create the app components were different but the procedure used to create them was structurally the same. For instance, the worked example showed steps to make a sound play when an image is clicked, and a problem solving task asked participants to make a label display text when an image is clicked. Participants had up to 25 minutes to complete the problem solving tasks.

Results and Discussion

For the problem solving tasks, participants received a score for number of correct steps taken towards problem solutions. For each correct action, such as adding code to play a drum sound, participants earned one point. Because the tasks involve multiple steps, scoring based on steps rather than whole answers provided more sensitivity. The maximum possible score was 25. Performance on the problem solving tasks depended on the interaction of subgoal learning method and feedback, $F(4, 190) = 3.39$, MSE = 23.6, $p = .01$, partial $\eta^2 = .067$. Due to the disordinal nature of this interaction, the main effects will not be reported to avoid confusion in interpreting the results (Maxwell & Delaney, 2004). Simple main effects comparisons were used to determine the effect of feedback on each method of learning subgoals. This analysis found that feedback affected the guided constructive groups, but it affected them in different ways (see Table 1). Participants in the guided constructive with hints conditions performed statistically better when they did not receive feedback than when they did, whereas participants in the guided constructive without hints conditions performed statistically better when they received feedback than when they did not.

Table 1: Simple main effects analysis of subgoal learning methods on problem solving performance. * indicates statistical significance at the .05 level.

Learning Method	Mean for No Feedback	Mean for Feedback	Mean Difference	Std. Error
Passive	15.5	17.4	-1.90	1.54
Active	18.0	16.1	1.95	1.54
Guided Constructive with Hints	21.0	17.5	3.50*	1.54
Guided Constructive without Hints	18.0	21.5	-3.54*	1.54
Unguided Constructive	18.0	18.3	-0.30	1.54

A simple main effects comparison was used for the feedback variable to explore the relative efficacy of different methods of learning subgoals,. Method of learning subgoals affected performance for groups that received feedback, $F(4, 190) = 3.54$, MSE = 23.6, $p = .008$, partial $\eta^2 = .069$, and groups that did not receive feedback, $F(4, 190) = 3.27$, MSE = 23.6, $p = .013$, partial $\eta^2 = .064$. Based on pairwise comparisons within the two types of feedback groups, those in the guided constructive with hints and without feedback condition performed statistically better than those in the passive condition without feedback, Mean Difference = 5.55, $p = .004$. Furthermore, participants in the guided constructive without hints condition and with feedback performed statistically better than those in the active condition with feedback, Mean Difference = 5.45, $p = .005$. These results suggest that, within both the feedback and no feedback groups, the best performing conditions scored statistically significantly better than those in the worst performing conditions. The other conditions that scored in the middle were not statistically better or worse than the best or worst performing conditions.

Overall, this pattern of results matched the expected pattern of results, suggesting that there is an optimal level of support for learning subgoals. In particular, the disordinal effect of feedback on the guided constructive groups suggests that learners perform best with just enough support and providing too much support hinders learning. Based on these results, it was concluded that providing hints for learners constructing subgoal labels and providing feedback on constructed labels are both techniques that can help learners to perform better on later problem solving, but providing both types of support could hurt performance.

Participant-Created Labels

To determine the quality of participant-created labels, they were qualitatively analyzed. Each label was analyzed as one unit (i.e., each word within a label was not analyzed individually), and each participant was categorized based on all of their labels collectively. In nearly all cases, all of the labels that a participant created fell into one of the following categories. The coding scheme included categories for whether labels were context-specific, context-independent, or incorrect. Context-specific labels included information about the specific instance of the subgoal and, therefore, could be applied only to that one instance. For example, the participant-created label “name and add picture to image sprite” could be applied only to the steps that named and added a picture to an Image Sprite. For a participant’s labels to be classified as context-specific, at least 80% of their labels had to include information about the context. Context-independent labels, on the other hand, did not contain any information about the specific instantiation of that subgoal. For example, the participant-created label “add properties to app” is context-independent because it can be applied to any property, such as the name and picture of an Image Sprite, that is being added to the app. To be classified as context-independent, at least 80% of labels had to not include information about the context. Context-specific labels were considered to be of a lower quality than context-independent labels because they cannot be applied to novel problems. Context-independent labels indicate a more conceptual understanding of the procedure that is more easily applied to solving new problems.

Incorrect subgoal labels were those that were execution-based instead of function-based, such as “click on menu,” or those that did not describe the correct function. To be classified as incorrect, more than one label had to meet either of these criteria. For the unguided constructive conditions, many of the subgoals that participants identified included many more steps than the subgoals created by experimenters. For example, some subgoals that participants grouped were more than 20 steps long, whereas the longest experimenter-grouped subgoal was seven steps. In all cases, the participant-created labels for these higher level subgoals were context-specific. For example, one participant identified a subgoal that was 24 steps long and labeled it “make the correct sounds play according to whatever input is received.” To distinguish these labels from the other context-specific labels, these labels were classified as higher-level context-specific labels. The higher-level context-specific labels were considered lower quality subgoal labels than the context-independent or –specific labels. One of the benefits of learning the subgoals of a procedure is that subgoals break up long procedures into functional pieces that are easier to adapt to novel problems. The higher-level subgoals were not identifying these functional pieces but instead describing the procedure that was being executed.

Most participants in the guided constructive with hints conditions created context-independent labels (69%). Some of these participants created context-specific labels (22%) or incorrect labels (8%). Many participants in the guided constructive without hints conditions created context-independent labels (49%). A proportion of these participants created context-specific labels (27%) or incorrect labels (24%). The majority of participants in the unguided constructive conditions created higher-level context-specific labels (79%). A small number of these participants created context-independent labels (9%), context-specific labels (9%), and incorrect labels (3%).

Most of the participants in the guided constructive with hints conditions created subgoal labels that were similar to the experimenter-created labels, meaning that they created labels that aligned with those created through an intensive task analysis with a subject-matter expert. Therefore, it is not surprising that these participants did not benefit from feedback (i.e., experimenter-created labels) and performed well on the novel problem solving tasks. In fact, the condition that did not receive feedback performed better that the condition that did. Because participants created high quality labels, comparing their labels to the experimenter-created labels in the feedback might not have been as beneficial as reviewing the labels that they constructed, as participants in the no feedback condition did. Comparing labels might have caused participants to unjustifiably question or doubt their understanding of the procedure, whereas reviewing their own labels would reinforce the mental representations that participants developed.

In summary, the results suggest that constructive methods of learning subgoals (i.e., self-explaining subgoals) are the most effective, but they require some instructional support. Either receiving feedback on constructed labels or receiving hints while constructing labels, but not both, led to the best problem solving performance. Participants who received hints while constructing labels were more likely to construct high quality labels than participants who did not receive hints. These participants performed better when they did not receive feedback than when they did, suggesting that the feedback provided too much instructional support to promote constructive learning. In contrast, participants who did not receive hints performed better when they received feedback than when they did not, suggesting that the feedback was necessary for the best performance when participants did not receive hints.

Acknowledgments

Our thanks to the reviewers who helped to improve this paper. We also thank Mark Guzdial and Briana Morrison for their help in guiding this research and Wesley Hughes and Ben Freed for their help in collecting and coding the data.

References

Aleven, V. A. W. M. M., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science, 26*, 147-179.

Atkinson, R. K., Catrambone, R., & Merrill, M. M. (2003). Aiding transfer in statistics: Examining the use of conceptually oriented equations and elaborations during subgoal learning. *Journal of Educational Psychology, 95*(4), 762-773.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.) (2000). How people learn: Brain, mind, experience, and school: Expanded edition. Retrieved from http://www.nap. edu/catalog.php?record_id=9853

Catrambone, R. (1994). Improving examples to improve transfer to novel problems. *Memory and Cognition, 22*, 605-615. doi:10.3758/BF03198399

Catrambone, R. (1996). Generalizing solution procedures learned from examples. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 22*, 1020-1031. doi:10.1037/0278-7393.22.4.1020

Catrambone, R. (1998). The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of Experimental Psychology: General, 127*, 355-376.

Catrambone, R., Gane, B. D., Adams, A. E., Bujak, K. R., Kline, K. A., & Eiriksdottir, E. (2016). Task analysis by problem solving (TAPS): A method for uncovering expert knowledge. Under review.

Catrambone, R. & Holyoak, K. J. (1989). Overcoming contextual limitations on problem-solving transfer. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*(6), 1147-1156.

Chi. M. T. H. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science, 1*(1), 73-105.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science, 13*, 145-182.

Conati, C., & VanLehn, K. (2000). Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation. International *Journal of Artificial Intelligence in Education, 11*, 389-415.

Durso, F. T., Rea, C. B., & Dayton, T. (1994). Graph-theoretic confirmation of restructuring during insight. *Psychological Science, 5*(2), 94-98.

Eiriksdottir, E., & Catrambone, R. (2011). Procedural instructions, principles, and examples: How to structure instructions for procedural tasks to enhance performance, learning, and transfer. *Human Factors, 53*(6), 749-770. doi:10.1177/0018720811419154

Hundhausen, C. D., Farley, S. F., & Brown, J. L. (2009). Can direct manipulation lower the barriers to computer programming and promote transfer of training?: An experimental study. *ACM Transactions in CHI, 16*(3).

Margulieux, L. E. & Catrambone, R., (2014). Improving programming instruction with subgoal labeled instructional text. In P. Bello, M. Guarini, M. McShane, & B. Scassellati (Eds.) Proceedings of the 35th Annual Conference of CogSci (pp. 952-957). Austin, TX: Cognitive Science Society.

Margulieux, L. E., Guzdial, M., & Catrambone, R. (2012). Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. Proceedings of the Ninth Annual International Conference on ICER (pp. 71-78). New York, NY: ACM. doi:10.1145/2361276.2361291

Maxwell, S. E., & Delaney, H. D. (2004). Designing experiments and analyzing data: A model comparison perspective (2nd ed.). New York, NY: Psychology Press.

Renkl, A. (2002). Worked-out examples: Instructional explanations support learning by self-explanations. *Learning and Instruction, 12*, 529-556.

Schworm, S., & Renkl, A. (2006). Computer-supported example-based learning: When instructional explanations reduce self-explanations. *Computers & Education, 46*, 426-445.

Wylie, R., & Chi, M. T. H. (2014). The self-explanation principle in multimedia learning. In R. Mayer (Ed.) The Cambridge Handbook of Multimedia Learning, 2nd Edition (pp.413-432). Cambridge University Press.