Interaction of Instructional Material Order and Subgoal Labels on Learning in Programming

Laura M Schaeffer (lschaeffer6@gatech.edu)
Department of Psychology, 654 Cherry Street
Atlanta, GA 30332 USA

Lauren Margulieux (l.marg@gatech.edu)
Department of Psychology, 654 Cherry Street
Atlanta, GA 30332 USA

Richard Catrambone (richard.catrambone@psych.gatech.edu)
Department of Psychology, 654 Cherry Street
Atlanta, GA 30332 USA

Abstract

Subgoal labeled expository instructions and worked examples have been shown to positively impact student learning and performance in computer science education. This study examined whether problem solving performance differed based on the order of expository instructions and worked examples and the presence of subgoal labels within the instructions for creating applications (Apps) for phones. Participants were 132 undergraduates. A significant interaction showed that when learners were presented with the worked example followed by the expository instructions containing subgoal labels, the learner was better at outlining the procedure for creating an application. However, the manipulations did not affect novel problem solving performance or explanations of solutions,. These results suggest that some limited benefit can be gained from presenting a worked example before expository instructions when subgoal labels are included.

Keywords: instructional design; STEM education; programming.

Introduction

Learners have difficulty solving novel problems, or problems that require steps that are different from worked example problems they have already encountered (Catrambone, 1995; Reed, Dempster, & Ettinger, 1985; Ross, 1987, 1989). This difficulty stems from learners tending to fixate on superficial aspects of examples as opposed to the goal structure of the problem. When learners understand the goal structure of the example problems, they become more successful at solving novel problems (e.g., Catrambone, 1995).

Subgoals are part of the task structure and organize solution steps into a meaningful hierarchy; subgoals are specific to problems within a particular domain (Catrambone, 1994; Catrambone, 1998). Subgoal labels assist learners in noticing and learning the subgoals and organizing their problem solving knowledge. This organization is demonstrated when learners who received instructions with subgoal labels tended to explain their problem solutions using the subgoals (Catrambone, 1995; Margulieux, 2013). Subgoal labels within instructions have improved transfer in many domains, including computer programming, and have been shown to be most effective when provided in both expository instructions and worked examples (Margulieux, 2013).

Expository Instructions

Expository instructions usually consist of both declarative information, such as terminology, and procedural information (Trafton & Reiser, 1993). Procedural instructions describe and explain how to carry out a task (Eiriksdottir & Catrambone, 2011). Procedural instructions are often written at a more general level than worked examples, so they can be applied to a variety of situations. The learner is equipped with the high level concepts needed to solve novel problems within the domain (Catrambone, 1990). This allows students who master procedural instructions to be able to solve novel problems better than students who receive more specific instructions (Catrambone, 1990). However, because procedural instructions do not have the same level of detail as more specific instructions, such as a worked example, more detailed information must be inferred. This inferential process is quite challenging for many learners.

Worked Examples

Worked examples demonstrate how a specific instance of a task is performed (Eiriksdottir & Catrambone, 2011). Worked examples are generally structured as a problem statement followed by the steps needed to arrive at the solution. They provide a concrete application of the problem solution's abstract concepts, rules, and general directions (Charney & Reder, 1987; Pirolli & Recker, 1994; Wiedenbeck, 1989). This allows the learner to become familiar with the task and increase their understanding of how to carry out the task (Eiriksdottir & Catrambone, 2011). Because worked examples provide detailed information, learners are able to more easily apply the same procedure to a similar problem than if they had been given more abstract information (Catrambone, 1990). Learners who use worked examples have also been shown to perform similar tasks more quickly than learners who used only procedural instructions (Catrambone, 1990).

One drawback of typical worked examples is that they do not inherently provide the learner with any general methods or reasoning behind decisions (Eiriksdottir & Catrambone, 2011). When given a worked example, the learner must infer information such as the nature of the task, the purpose of each step, rules governing the steps, subgoals, and organization (LeFevre & Dixon, 1986; Pirolli & Recker, 1994). In limited cases learners have been shown to infer general methods when several worked examples are presented, but usually guidance is needed for such connections to be made (Rumelhart & Norman, 1981). Presenting the learner with both procedural instructions and worked examples has been shown to produce the benefits associated with each type of instructional material while reducing the drawbacks. Catrambone (1995) showed that presenting procedural text with a worked example aided both initial performance and transfer.

There is reason to believe the order in which the instructions are presented might affect the learner's ability to process them. Several lines of research suggest that students perform and learn better when given a worked example followed by procedural texts (Alfieri, Nokes-Malach, & Schunn, 2013; Anderson, 1990; Dale, 1946). Dale (1946) argued that when learning math, students should first be introduced to concrete objects (e.g., five fingers as opposed to an abstract five), and then work up to semi-concrete ideas. If the material does not relate to a student's experience with the items in the equation, the formula will not mean anything (Dale, 1946). Dale (1946) concluded that the role of the teacher is to take the student from concrete experiences to significant and important generalizations. Other studies also suggest that it is better to give people principles for the concept or procedure that they are trying to learn after they view the cases (Alfieri, Nokes-Malach, & Schunn, 2013).

Another theory, from the inductive teaching research literature, suggests that worked examples provide the "why" behind the principles and procedure (Prince & Felder, 2006). The specifics from worked examples cause the learner to generate a need for more information, such as the rules, procedures, and principles. This curiosity then motivates the learner to incorporate and apply the instructions.

It has been noted that new information is best learned when the learner has a knowledge base to support the information, and they are unlikely to learn if the new information has few apparent connections to what they already know. Advance organizers have been used to provide such a foundation (Ausubel, 1968; Novak, 1977). Advance organizers can be used as an effective way to bridge the gap between the novice's knowledge and the basis on which the instructions function (Ausubel, 1968). When presented at a suitable level for the learner, advance organizers activate the learner's prior knowledge making the new information more familiar and meaningful, which decreases dependence on sheer memorization in favor of a meaningful understanding of the information.

A worked example might serve a similar function as an advance organizer because it gives the learner a base on which to apply the latter expository information. A worked example introduces the learner to the type of situation to which the expository information is applicable, mobilizing the learner's prior knowledge. Therefore, instructional materials might be more effective if the worked example is presented before the expository information.

Alternatively, presenting the worked example first might be disadvantageous. According to Ausubel (1968), instructions aid mental organization better when progressing from abstract ideas to specific details because this organization better fits our cognitive structure. Additionally, presenting specific details first, such as those found in the worked example, might cause the learner to focus on applying the expository instructions to problems that are very similar to the worked example. Consequently, the learner might have a more difficult time generalizing the instructions to other situations. Because of this, presenting the worked example first might hinder the learner's ability to use the abstract principles when solving novel problems. However, subgoal labels might help learners compensate for this effect because they explicitly provide the higher level functions found within the worked example and the expository instructions.

Present Study

The present study investigated the effects of instructional material order and subgoal labels in learning computer programming. Participants were taught how to use the programming language (Android App Inventor) to create a Fortune Teller application (app). The App Inventor programming environment uses a drag-and-drop interface to create apps for Android devices.

Drag-and-drop programming is ideal for novices because instead of writing code, the learners drag components from a menu and fashion them together like puzzle pieces. Creating code in this way has been shown to be easier for novices to comprehend than other types of programming environments (Hundhausen, Farley, & Brown, 2009).

Videos were used to convey the App Inventor instructions because videos have been shown to be a natural and efficient way for learners to gain knowledge of direct-manipulation interfaces (Palmiter & Elkerton, 1993; Palmiter, Elkerton, & Baggett, 1991). Participants also used a practice problem guide to practice creating the Fortune Teller app before being tested. Trafton and Reiser (1993) showed that learners who study and practice newly learned material are better able to apply the material than learners who are not given the opportunity to practice.

Method

Participants

Participants were 132 undergraduate students from the Georgia Institute of Technology compensated with course credit. The sample consisted of 68 females and 64 males. The mean age was 19.3 years with a standard deviation of 1.93. Participants were excluded if they had taken more than one computer science course or had experience with App Inventor. These qualifications were necessary because the instructional materials were designed for novices.

Design

The experiment was a two-by-two, between subjects, factorial design with 33 participants per cell. The first independent variable was the order participants received the instructional materials: expository followed by worked example or worked example followed by expository. The second independent variable was presence of subgoal labels: present or absent. The dependent variables consisted of performance on three assessment tasks to determine organization of domain knowledge and problem solving performance.

Procedure

Each session lasted between 60 and 90 minutes. Participants were randomly assigned to one of four conditions. All participants first completed the demographic questionnaire. Next, participants began the instructional period where they watched both instructional videos (the expository video and the worked example video) before using a practice problem guide to practice creating an app. The expository instructional videos contained general procedural instructions and declarative information, such as definitions, necessary for creating an app in App Inventor. The worked example video demonstrated how to create a specific app, the Fortune Teller app. Subgoals were created by Margulieux (2013) using the Task Analysis by Problem Solving (TAPS) method developed by Catrambone et al., (2012).

The videos used callouts to present the subgoal labels. These were text boxes containing the subgoal labels appearing on screen while the narration continued explaining the steps needed to achieve the subgoal.

The final instructional material was the practice problem guide, which was a scaffolded worked example. The stages of scaffolding can vary (Pea, 2004), but in the present study the practice problem guide provided learners with the steps necessary for creating the Fortune Teller app without giving them guidance on how to carry out the steps (e.g., where in the menus to find blocks). The scaffolded example used the same Fortune Teller app presented in the worked example video.

After the instructional period, the participants began the assessment period. During the assessment period, the participants were not able to use the materials from the instructional period. However, they were able to use the App Inventor website and refer to the app they created during the instructional period as an aid to problem solving (Margulieux, 2013). The first assessment consisted of four problem solving tasks in which participants were instructed to add or modify features of their Fortune Teller app. This assessment was broken into two parts where the participants were first asked to modify the app directly in App Inventor, and later asked to write down the necessary steps. The written portion allowed participants to demonstrate their knowledge of steps even if they did not know how to correctly execute the steps in App Inventor. This assessment measured participants' problem solving performance on novel tasks using App Inventor.

The second assessment was the explanation task. Correct solutions to the four problem solving tasks were given to the participants. Participants were asked to group steps of the problem solving task solution. They were then asked to label their groups by describing what goal was met for each grouping. This assessment measured how well participants could group steps based on structural similarity, and how well they could explain the solutions.

The final assessment was the generalization task that asked participants to describe the general procedure that they would use to create an app within a given set of constraints. A correct response to this task included the fundamental steps needed to make the app while excluding unnecessary details. This assessment was used to measure how well the participants could use abstract principles to outline the task procedure they learned earlier in the session.

Results

Demographic information such as age, GPA, college major, and experience with computer science were collected but were not correlated with performance on any of the following assessments.

General Procedure Task

The general procedure asked participants to describe the general process they would use to create an app. One point was awarded for each structurally necessary feature the participant described, for up to a maximum score of 6. ICC(A) for this assessment was .99. There was no main effect of instructional material order, *F* (1, 132) = 0.58, *p* = .45 There was also no main effect of subgoal labels, *F* (1, 132) = 1.31, *p* = .26 (see Table 1).

Table 1. Score on Task for Describing General Process to Create an App

Worked Example First		*Expository First*	
Subgoals	*No labels*	*Subgoals*	*No labels*
M (SD)	*M (SD)*	*M (SD)*	*M (SD)*
2.85 (1.46)	2.03 (1.22)	2.12 (1.47)	2.40 (1.29)

Note: Score out of six possible points.

However, there was a significant interaction between the instructional material order and subgoal labeling, F (1, 132) = 5.49, p = .02. Simple main effects analysis showed that participants who received subgoal labels were able to provide more steps of the general process for creating an app than those who did not receive subgoal labels when presented with the worked example before the expository instructions, p = .02, but there were no differences between the subgoal labeled group and the group without subgoal labels when the expository instructions were presented before the worked example, p= .40.

Problem Solving Tasks

The following assessments were scored following the method developed by Margulieux et al. (2012), which has been shown to have high statistical power (due to partial scoring methods discussed later) and high interrater reliability. Two raters scored each of the assessments; interrater reliability was measured with an intraclass correlation coefficient of absolute agreement (ICC(A)).

Performance in App Inventor. For this task, participants were asked to modify or add different features of an app. They were awarded one point for each correct action in App Inventor taken towards the problem solutions for up to a maximum score of 22. ICC(A) for this assessment was .89. Visual inspection of the data revealed that the data were not normally distributed (see Figure 1). The residuals were not normally distributed, violating the normality assumption of the ANOVA. Therefore, a Kruskal-Wallis H test was used to determine if there were differences in the performance score among the four instructional groups. The mean rank of performance scores was not statistically significantly different among groups, $\chi^2(3) = .789$, $p = .852$ (see Table 2).

This was unexpected because prior research suggests that subgoal labels benefit problem solving by helping learners to represent their problem solving knowledge in a way that allows more flexible transfer (e.g. Catrambone, 1998; Margulieux, 2013). For the main effect of subgoal labels, the present study showed $\eta^2_p = 0.003$, and the observed power was 0.09 compared to $\eta^2_p = .38$ found in Margulieux's (2013) study. The present study saw a very small effect size that would have needed a much larger sample to reveal any significant differences.

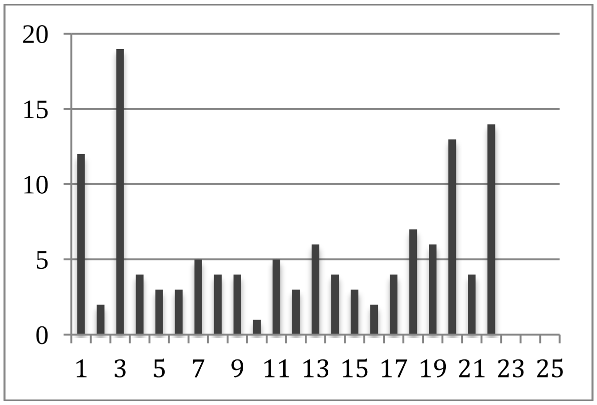

Figure 1. Distribution of scores for
Problem Solving Task: Performance in AppInventor.

Written Performance. Participants were awarded one point for each correct step written towards achieving the problem solution for up to a maximum score of 22, and the ICC(A) for this assessment was .91. Visual inspection of the data revealed that the data were not normally distributed. The residuals did not have a normal distribution, violating the normality assumption of the ANOVA. Therefore, a Kruskal-Wallis H test was used to determine if there were differences in written performance score among the four instructional groups. The mean rank of the written performance scores was not statistically significantly different between groups, $\chi^2(3) = 1.64$, $p = .65$. These results did not support the hypothesis that instructional order and subgoal labels would affect the declarative knowledge concerning how to modify and add features to an app in App Inventor.

Explanation Task

In order to measure how well participants could organize and explain problem solutions, participants were given the solutions and instructed to meaningfully group and label the solution steps. Participants were awarded one point for each group that contained only structurally similar steps, for up to

Table 2. Descriptive Statistics for Problem Solving Task

	Worked Example First		*Expository First*	
	Subgoals	*No labels*	*Subgoals*	*No labels*
	M (SD)	*M (SD)*	*M (SD)*	*M (SD)*
App Inventor Performance	11.41 (7.43)	10.25 (6.44)	10.75 (7.99)	10.37 (8.44)
Written Performance	10.91 (7.13)	10.15 (6.82)	8.67 (6.68)	10.42 (6.67)
Attempted Subgoals	6.41 (3.61)	6.06 (3.05)	5.75 (3.51)	6.22 (3.32)

Table 3. Descriptive Statistics for Explanation Task

	Worked Example First		*Expository First*	
	Subgoals	*No labels*	*Subgoals*	*No labels*
	M (SD)	*M (SD)*	*M (SD)*	*M (SD)*
Grouping	3.90 (1.88)	3.87 (1.78)	4.03 (2.01)	3.83 (1.91)
Explanations	1.46 (1.72)	1.24 (1.80)	1.26 (1.87)	1.45 (1.83)

Note: Scored out of a possible ten points.

a maximum of 10 points. ICC(A) for this assessment was .98. There were no significant differences on grouping structurally similar solution steps based on instructional material order, *F* (1, 116) = 0.02, *p* = .89, subgoal labels, *F* (1, 116) = 0.11, *p* = .74, or interaction , *F* (1, 116) = 0.06, *p* = .81 (see Table 3).Labels were scored for whether they described the function of the group of steps. For each label, participants earned one point if the explanation identified the purpose of the grouped steps. There were no significant differences based on instructional material order, *F* (1, 136) = 0.00, *p* = .98, subgoal labels , *F* (1, 136) = 0.00, *p* = .97, or the interaction , *F* (1, 136) = 0.47, *p* = .50. The hypothesis that the order the materials were presented, labeling of subgoals, and the interaction would effect performance on organizing and explaining problems solutions was not supported.

Discussion

The present study showed limited evidence that the instructional material order and subgoal labels affect a learner's performance in computer programming. This study suggests that similar learning occurs regardless of whether the worked example is presented before or after the expository instructions. The exception to this is that when asked to provide a general outline for creating an app, participants whose instructions contained subgoal labels and received the worked example before the expository instructions performed better than the other groups.

The reasoning behind presenting the worked example before the expository instructions was partly based on the literature about advance organizers. The benefit of an advance organizer lies on relating the new information to the existing cognitive structures. However, it is possible that the given instructions were not aligned with the participants' cognitive structures. The distribution of scores for the problem solving task in Figure 1 show that although some students did well, many performed poorly. It is plausible that the instructions might have been at an appropriate level for the high performers, but not for the low performers. For the participants who did not do well, the worked example might not have been able to bridge the gap between what the learners already knew and what they were about to learn. Instead, the instructions might have primarily been new information that was not easily anchored to existing cognitive structures. The inductive teaching literature shows that learners are unlikely to learn new information when there are few apparent connections to what the learner already knows. If the instructions were not at the proper level for the learner, then it follows that presenting the worked example first would have no added benefit.

Contrary to previous research such as Margulieux (2013), subgoal labels did not affect problem solving performance. There are several possible reasons that results in this study differed from results of previous research on subgoal labels. The main difference in research materials between this study and Margulieux (2013) is the media used for the expository instructions. Margulieux (2013) used a text document to convey this information, whereas the present study narrated the text document during a video. The use of a text document might have reduced the cognitive load as well as ambiguity of these instructions because the learner did not need to mentally transpose the text information to the App Inventor interface. Additionally, auditory information is more transient than text on a piece of paper; each piece of auditory information lasts for only a short period of time compared to text information that is continually present. Instructions presented through videos tend to be processed at a more superficial level than text instructions (Palmiter & Elkerton, 1993). Therefore, the subgoal labels in the videos might not have been processed to the same extent as when they were presented in a text document. As discussed previously, subgoal labels are thought to provide a framework for problem solving and aid in the creation of mental representations. However, if the information was not presented for a long enough duration, or processed to the necessary extent, the learner would not be able to form these connections..

Further Work

Further research should investigate the effectiveness of subgoal labels in videos compared to subgoal labels in text instructions, since the lack of a subgoal effect in the present study was surprising. Future research should also broaden the sample to include groups other than undergraduates to increase generalizability to other student groups. Additionally, this study focused on performance on the same day the task was learned. Testing after a delay would reveal how well the instructions were incorporated and applied long term. Much instruction aims to teach knowledge and skills that will be used not just on tasks on the day of instruction, but on future tasks. Investigating knowledge that is retained days and weeks after instruction is more reflective of the real-world application of this type of instruction.

Acknowledgments

Thank you to Wendy Rogers and Frank Durso for their guidance, and to Lien Nguyen and Sarah Nay for their work on this project.

References

Alfieri, L., Nokes-Malach, T. J., & Schunn, C. D. (2013). Learning through case comparisons: a meta-analytic review. *Educational Psychologist*, *48*(2), 87-113.

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Lawrence Erlbaum.

Ausubel, D. P. (1968). *Educational psychology: A cognitive view*. New York: Holt, Rinehart and Winston.

Catrambone, R. (1990). Specific versus general procedures in instructions. *Human-Computer Interaction, 5,* 49–93.

Catrambone, R. (1994). Improving examples to improve transfer to novel problems. *Memory & Cognition, 22*(5), 606-615.

Catrambone, R. (1995). Aiding subgoal learning: Effects on transfer. *Journal of Educational Psychology, 87*(1), 5-17. doi: 10.1037/0022-0663.87.1.5

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. (2012). Task Analysis by Problem Solving (TAPS): A Method for Uncovering Expert Knowledge. Unpublished manuscript, School of Psychology, Georgia Institute of Technology, Atlanta, GA.

Charney, D. H., & Reder, L. M. (1987). Initial skill learning: An analysis of how elaborations facilitate the three components. In P. Morris (Ed.), *Modelling cognition* (pp. 135–165). Chichester, UK: Wiley.

Dale, E. (1946). *Audiovisual Methods in Teaching.* Dryden Press, New York, NY.

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 directed manipulation lower the barriers to computer programming and promote transfer of training?: An experimental study. *ACM Transactions in CHI, 16*(3). doi: 10.1145/1592440.1592442

LeFevre, J.. & Dixon, P. (1986). Do written instructions need examples? *Cognition and Instruction, 3,* l-30. doi: 10.1207/s1532690xci0301_1

Margulieux, L. E. (2013) *Subgoal Labeled Instructional Text and Worked Examples in STEM Education.* Unpublished master's thesis, Georgia Institute of Technology, Atlanta, Georgia.

Margulieux, L. E., Guzdial, M., & Catrambone, R. (2012). Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research*, 71-78. doi: 10.1145/2361276.2361291

Novak, J.N. (1977). *A Theory of Education*. New York: Cornell University Press.

Palmiter, S., & Elkerton, J. (1993). Animated demonstrations for learning procedural computer-based tasks. *Human-Computer Interaction, 8*(3), 193-216. doi:10.1207/s15327051hci0803_1

Palmiter, S., Elkerton, J., & Baggett, P. (1991). Animated demonstrations versus written instructions for learning procedural tasks: A preliminary investigation. *International Journal of Man-Machine Studies, 34,* 687-701. doi: 10.1016/0020-7373(91)90019-4

Pea, R. (2004). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. *Journal of the Learning Sciences, 13*(3), 423-451. doi: 10.1207/s15327809jls1303_6

Pirolli, P., & Recker, M. M. (1994). Learning strategies and transfer in the domain of programming. *Cognition and Instruction*, *12*, 235–275.

Reed, S. K., Dempster, A., & Ettinger, M. (1985). Usefulness of analogous solutions for solving algebra word problems. *Journal Of Experimental Psychology: Learning, Memory, And Cognition*, *11*(1), 106-125. doi:10.1037/0278-7393.11.1.106

Ross, B. (1987). This is like that: The use of earlier problems and the separation of similarity effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13,* 629-639.

Ross, B. (1989). Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15,* 456-468.

Rumelhart, D. E., & Norman, D. A. (1981). Analogical processes in learning. In J. R. Anderson (Ed.), Cognitive skills and their acquisition (pp. 335-360). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Trafton, J. G., & Reiser, B. J. (1993). *Studying examples and solving problems: Contributions to skill acquisition*. Technical report, Naval HCI Research Lab, Washington, DC, USA.

Wiedenbeck, S. (1989). Learning iteration and recursion from examples. *International Journal of Man-Machine Studies, 30*(1), 1–22.