6-30-2016

Training Learnings to Self-Explains: Designing Instructions and Examples to Improve Problem Solving

Lauren E. Margulieux
Georgia Institute of Technology

Briana B. Morrison
University of Nebraska at Omaha, bbmorrison@unomaha.edu

Richard Catrambone
Georgia Institute of Technology

12th INTERNATIONAL CONFERENCE OF THE LEARNING SCIENCES

ICLS 2016
SINGAPORE
20-24 June

ISBN: 978-0-9903550-9-0

Edited by
Chee-Kit Looi,
Joseph Polman,
Ulrike Cress, and
Peter Reimann

Transforming Learning, Empowering Learners

Conference Proceedings
Volume 1

Organiser

International Society of the Learning Sciences

Host

NIE NATIONAL INSTITUTE OF EDUCATION SINGAPORE
An Institute of Nanyang Technological University

Co-Host

CRADLE
Centre for Research and Development in Learning

Transforming Learning, Empowering Learners:
The International Conference of the Learning Sciences (ICLS) 2016

Volume 1

12th International Conference
of the Learning Sciences

June 20-24, 2016, Singapore
National Institute of Education,
Nanyang Technological University,
Singapore

Editors:
Chee-Kit Looi, Joseph Polman, Ulrike Cress, and Peter Reimann

Transforming Learning, Empowering Learners: The International Conference of the Learning Sciences (ICLS) 2016
June 20-24, 2014, Singapore
National Institute of Education, Nanyang Technological University, Singapore.

www.isls.org

ISBN: 978-0-9903550-9-0 (Volume 1, PDF Version)
ISSN: 1814-9316

Cite as: Looi, C. K., Polman, J. L., Cress, U., and Reimann, P. (Eds.). (2016). *Transforming Learning, Empowering Learners: The International Conference of the Learning Sciences (ICLS) 2016, Volume 1.* Singapore: International Society of the Learning Sciences.

Conference logo and cover design by Li Li Ng

Training Learners to Self-Explain: Designing Instructions and Examples to Improve Problem Solving

Lauren E. Margulieux, Briana B. Morrison, Mark Guzdial, and Richard Catrambone
l.marg@gatech.edu, bmorrison@gatech.edu, guzdial@cc.gatech.edu, richard.catrambone@psych.gatech.edu
Georgia Institute of Technology

Abstract: In this experiment, we integrated two learning methods – subgoal learning and constructive learning – to explore their interactions and effects on solving computer programming problems. We taught learners to solve problems using worked example and practice problem pairs with one of three kinds of instructional design that either did not highlight the subgoals, described the subgoals, or prompted participants to describe the subgoals for themselves. In addition, we varied the distance of transfer between the worked example and practice problem pairs. We found that instructions that highlighted subgoals improved performance on later problem solving tasks. The groups that performed best were those that received subgoal descriptions with farther transfer between examples and practice problems and those that described subgoals for themselves with nearer transfer.

Keywords: worked examples, constructive learning, subgoal learning, self-explanation

Introduction

An important instructional tool for teaching problem solving in programming, and other science, technology, engineering, and math (STEM) domains, is the worked example. In this pedagogical approach, learners receive an example problem with the solution worked out (Renkl, Stark, Gruber, & Mandl, 1998). Ideally, students will use the worked examples to develop declarative rules or schemas that guide them in future problem solving. Empirical evidence has shown that learning with worked examples is more effective for acquiring problem solving skills than solving problems. This is called the worked example effect (Sweller, van Merrienboer, & Paas, 1998). However, research on the worked example effect has shown that merely presenting worked examples is not enough to promote student schema construction (Wittwer & Renkl, 2010). 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). For instance, when studying physics worked examples, learners are more likely to remember that the example has a ramp than that the example uses Newton's second law (Chi et al., 1989). A focus on superficial features leads to ineffective organization of information that, in turn, leads to ineffective recall and transfer (Bransford, Brown, & Cocking, 2000).

Subgoal learning

To promote deeper processing of worked examples, and thereby improve transfer, worked examples have been formatted to encourage subgoal learning by emphasizing the subgoals, or functional parts, of problem solving procedures to highlight the structural components of the problem solving process (Catrambone, 1998). Subgoals are the building blocks of procedural problem solving, and they are inherent in procedures. Each subgoal contains one or more steps. For the example in Figure 1, initializing the variables is a subgoal of the procedure used to solve problems with loops.

Research suggests that when instructions help students learn the subgoals of a procedure, students are better able to transfer knowledge to solve novel problems. To promote subgoal learning from worked examples, subgoal labeling has been used (e.g., Catrambone, 1998; Margulieux, Guzdial, & Catrambone, 2012). Subgoal labels are functional explanations that describe the purpose of a subgoal. For instance, in Figure 1 for the subgoal that initializes variables, the subgoal label might read "Initialize variables."

Subgoal labeled worked examples have improved problem solving performance in multiple STEM domains including statistics (Catrambone, 1998) and programming (Margulieux et al., 2012). Subgoal labels are believed to be effective because they visually group the steps of worked examples into subgoals and meaningfully label those groups (Atkinson, Catrambone, & Merrill, 2003). This format highlights the structure of examples, helping students focus on structural features and more effectively organize information (Atkinson, Derry, Renkl, & Wortham, 2000; Catrambone, 1998). Giving students subgoal labels, however, is a passive form of learning, and learning is generally more effective when students learn constructively (Chi, 2009).

Self-explanation

A common and effective type of constructive learning that might help learners understand subgoals is self-explanation. Self-explanation is a learning strategy in which students use prior knowledge and logical reasoning to make sense of information and gain knowledge. A review of self-explanation studies found it is effective across a range of domains as long as the domain has logical rules with few exceptions (Wylie & Chi, 2014).

Self-explanation of a worked example's solution identifies structural features and reasons about the function of the steps (Bielaczyc, Pirolli, & Brown, 1995). This purpose is similar to that of subgoal learning. By self-explaining worked examples, learners are more likely to recognize which features are structural and which are superficial. Learners, however, do not often engage in self-explanation on their own. Many studies (e.g., Chi et al., 1989) found that 10% or less of learners self-explained examples without external prompting. Much of the time, however, learners can self-explain if they devote additional resources to the task (Wylie & Chi, 2014) if they are reminded and guided to do so. Research has found little difference in the learning outcomes of students who are internally or externally prompted to self-explain, suggesting that self-explanation itself is the cause of learning benefits rather than learner characteristics (e.g., Bielaczyc et al., 1995).

Current research

The current research explores the effect of supporting learners to constructively develop their own subgoal labels through the process of self-explanation. We taught learners to solve problems using `while` loops with instructions that either did not have subgoal labels, had subgoal labels created by an instructional designer, or had placeholders for the student to generate their own subgoal labels (see Figure 1 for an example). The instructions included three worked examples and three practice problems.

No labels	Given labels (passive)	Placeholder for label (constructive)

sum = 0
lcv = 1
 	
WHILE lcv <= 100 DO
 	
   lcv = lcv + 1
 	
ENDWHILE
	<u>Initialize Variables</u>
	sum = 0
	lcv = 1
	
	<u>Determine Loop Condition</u>
	WHILE lcv <= 100 DO
	
	   <u>Update Loop Variable</u>
	   lcv = lcv + 1
	ENDWHILE
		<u>Label 1: _____.</u>
		sum = 0
		lcv = 1
		
		<u>Label 2: _____.</u>
		WHILE lcv <= 100 DO
		
		   <u>Label 3: _____.</u>
		   lcv = lcv + 1
		ENDWHILE

Figure 1. Partial worked example formatted with no labels, given labels, or placeholders for generated labels.

The worked examples and practice problems were interleaved so each worked example was paired with a similar practice problem. The practice problems either had isomorphic or contextual transfer from the worked examples. Isomorphic transfer meant that the worked example and practice problem were the same except for the values in each problem. For example, one worked example showed a program that would find the average tip amount for a restaurant server with the values \$15, \$5.50, \$6.75, etc. The paired practice problem with isomorphic transfer asked participants to find the average tip amount with the values \$20, \$8.25, \$9.75, etc. Alternatively, contextual transfer meant that the worked example and practice problem followed the same procedural steps but had different contexts. For example, for a worked example that found the average tip amount, the paired practice problem with contextual transfer asked participants to find average rainfall.

Giving learners practice problems to practice applying the procedure, even if the problems have minimal transfer from examples, allows students to monitor their learning and identify concepts that they superficially understand (Trafton & Reiser, 1993). The contextual transfer was intended to be harder for participants to map concepts from the worked example to the practice problem. More difficult mapping can improve learning by reducing illusions of understanding caused by shallow processing thus inducing deeper processing of information (Bjork, 1994; Eiriksdottir & Catrambone, 2011; Palmiter, Elkerton, & Baggett, 1991). However it can also increase cognitive load and potentially hinder learning by overloading cognitive resources (Sweller, 2010).

We hypothesized that students who generated subgoal labels would solve novel problems better than those who were given the subgoal labels, and both groups would solve problems better than those who had no subgoals at all. We also hypothesized that learners whose practice problems required contextual transfer would solve problems better than learners whose practice problems required only isomorphic transfer.

Methods

Materials

All participants received three worked examples and three practice problems. The examples demonstrated using `while` loops to solve problems that found the average amount of tips a restaurant server received (from an array of tip amounts), counted the number times a pair of dice rolled a 7 (from an array of dice rolls), and counted the number of prime numbers between 1 and 100. The isomorphic-transfer practice problems were in the same contexts, but they asked for the average tip amount (from a different array), the number of times a 2 was rolled (from the same array), and the number of prime numbers between 100 and 200, respectively. The contextual-transfer practice problems asked for the average amount of rainfall (from an array), the number of restaurants within 3 miles (from an array), and the number of unique phone numbers in a contact list (from an array), respectively.

Each participant received one of three formats for the worked examples. The first format did not highlight the subgoals of the procedure. The second format grouped individual steps of the example into subgoals and provided meaningful labels that described the function of each subgoal. This format is typical in subgoal label research (e.g., Catrambone, 1998; Margulieux et al., 2012). The third format grouped steps of the example into subgoals and provided a placeholder for participants to write their own labels. For this condition, each of the groups of steps was numbered as "label 1," "label 2," etc., and groups of steps that represented the same subgoal had the same number. For instance, groups that represented the "initialize variables" subgoal were called "label 1" regardless of where in the example they appeared. At the beginning of the session, participants who generated subgoals were told that each of the worked examples would have the same subgoals, and they were encouraged to update and improve upon their generated labels as they learned more about the procedure.

Mimicking the format of the worked examples, participants who received subgoal-oriented examples also received subgoal-oriented practice problems. If participants were given or generated subgoal labels in the examples, then the area in which participants solved practice problems was also structured with the given or generated subgoal labels, respectively. Instead of having a completely blank space to write the practice problem's solution, like in the non-subgoal-oriented conditions, the subgoal-oriented conditions had several small blank spaces headed by subgoal labels or placeholders for labels. This design is typical of subgoal label research that uses practice problems (e.g., Margulieux et al., 2012) and was intended to support learners in initial problem solving and highlight connections between the examples and practice problems.

Participants assigned to generate their own subgoal labels received training on how to create subgoal labels. The training included expository instructions about generating subgoal labels, an example of a subgoal labeled worked example, and activities in which participants practiced generating subgoal labels and received feedback on their labels. The feedback was the same for all participants and asked them to compare the labels that they made to the labels that an instructional designer made. Participants who were not assigned to generate their own subgoal labels did not receive this training because it might have prompted them to generate their own labels, which would confound the results. Instead, these participants received training to complete verbal analogies (e.g., water : thirst :: food : hunger). Verbal analogies were considered a comparable task to subgoal label training because they both require analyzing text to determine an underlying structure.

After finishing the instructions (i.e., training, worked examples, and practice problems), participants completed novel programming tasks. The tasks asked participants to solve four novel problems using loops. Two of these problems required contextual transfer, meaning that they followed the same steps found in the instructions but had a different context (i.e., the same type of transfer as in contextual-transfer practice problems). The other two problems required both contextual and structural transfer, which is farther transfer than contextual. In these problems the context was new *and* the solution to the problem required a different structure than presented in the instructions. For example, the instructions included problems for averaging values, and the assessment included problems for averaging the first and second half of a list separately.

Design

The experiment was a 3x2, between-subjects, factorial design. Format of examples and practice problems (unlabeled vs. given subgoal labels vs. generate subgoal labels) was crossed with transfer distance between worked examples and practice problems (isomorphic vs. contextual transfer). The dependent variables were problem solving performance, quality of generated labels when applicable, and time on problem solving tasks.

Participants

Participants included in the final analyses were 120 students, 20 in each condition, from introductory programming courses in two technical universities in the Southeast United States (see Table 1 for demographics). Students were offered credit for completing a lab activity or extra credit as compensation for participation. All students from these courses were allowed to participate, regardless of prior experience. To account for prior experience, participants were asked about their prior programming experience in high school and college and whether they had experience using `while` loops. Other demographic information collected included gender, age, academic major, high school GPA, college GPA, number of years in college, reported comfort with computers, expected difficulty of the programming task, and primary spoken language. There were no statistical differences among the groups for demographic data, which is expected because participants were randomly assigned to treatment groups. Participants also took a multiple-choice pre-test to measure problem solving performance for using `while` loops. Average scores on the pre-test were low, 1.6 out of 5 points, with 23% of participants scoring zero points. There were no correlations between pre-test score and format of worked examples and practice problems, $\rho = .07$, $p = .45$, or the transfer distance between them, $\rho = .01$, $p = .96$.

Table 1. Participant demographics

Age	Gender	GPA	Major
M = 21.6 years	71% male	M = 3.2/4	52% CS major

Many participants did not complete all tasks of the experiment. Participants received compensation regardless of the amount of time or effort that they devoted to the experiment, which might have caused low motivation in some participants. Participants who did not attempt all tasks (n = 43) were excluded from analysis. Participants who answered more than two questions correctly out of the five on the pre-test (n = 12) were also excluded from analysis because the instructions were designed for novices. Of the 175 students that participated in the experiment, 120 were included in final analyses.

Procedure

At the beginning of the session, participants completed the demographic questionnaire and pre-test. The pre-test had multiple choice questions about `while` loops from previous Advanced Placement Computer Science exams. Next, participants began the instructional period, which started with the subgoal label or analogy training. After the training, participants received the three worked example and practice problem pairs to help them learn to use `while` loops. When participants finished the instructions, they were asked to complete a 10 item survey designed to measure cognitive load while learning programming skills (Morrison, Dorn, & Guzdial, 2014). The placement of the survey at this point was to ensure measurement of cognitive load during the learning process and not during the assessments. Participants next completed the assessments. The assessments included four types of tasks, but the results of only the problem solving tasks, which were administered first, are discussed in this paper. Throughout the procedure, time on task was measured. Performance on activities in the subgoal label or analogy training and on practice problems was collected to ensure participants were completing tasks. The subgoal labels that participants generated were also recorded.

Results and discussion

Problem solving performance

Participants received a problem solving score based on the accuracy of their solutions. Participants earned one point for each correct line of code that they wrote, allowing for more sensitivity than scoring solutions as wholly right or wrong. If participants wrote lines that were conceptually correct but contained syntax errors (e.g., missing a parenthesis), they still received points. We scored logic errors (having < rather an <=) as incorrect. We considered scoring for conceptual and logical accuracy as more valuable than scoring for absolute accuracy because participants were in the early stages of learning. Participants could earn a maximum score of 44.

For problem solving performance among conditions, see Figure 2. We found a main effect of format of examples and practice problems, F (2, 114) = 5.07, MSE = 176.5, p = .008, est. $\omega^2 = .08$, f = .21. To explore this result, we conducted a post-hoc analysis with the LSD test because it is the most powerful for comparing three groups. We found that both subgoal-oriented formats (i.e., given or generate subgoal labels) performed better than the unlabeled group, mean difference = 7.8, p = .01, and mean difference = 8.6, p = .005, respectively. Both subgoal-oriented formats performed equally, mean difference = .78, p = .80. For transfer distance between examples and practice problems, we found no main effect, F (2, 114) = 0.42, MSE = 176.5, p = .52, est. $\omega^2 = .004$. These findings are tempered by an interaction between the two interventions.

We found a small, but interesting, interaction between the format of worked examples and practice problems and the transfer distance between them, F (2, 114) = 2.71, MSE = 176.5, p = .071, est. $\omega^2 = .05$, f = .15.

Though this interaction does not pass the threshold for statistical significance in the null hypothesis significance testing framework, the size of the effect makes it worth discussing. We found three levels of performance, as can be seen in Figure 2. The best performing groups were those that were given subgoal labels with contextual transfer ($M = 25.3$) and generated subgoal labels with isomorphic transfer ($M = 25.8$). The middle groups were those that received no subgoal labels with isomorphic transfer ($M = 16.9$), received labels with isomorphic transfer ($M = 18.9$), or generated subgoal labels with contextual transfer ($M = 19.9$). The worst performing group received no subgoal labels with contextual transfer ($M = 11.7$).

Each level of performance is separated by about seven points, or 16% of the total score. The difference between the middle and best level of performance was not statistically significant but had a medium effect size, as shown by the t-test comparing groups that were given subgoal labels (middle bars in Figure 2), $t(38) = 1.45$, $p = .15$, $d = .46$. Similarly, the difference between the middle and worst level of performance was not statistically significant but had a medium effect size, as shown by the t-test comparing groups that received labels with isomorphic transfer and that did not receive labels with contextual transfer (second and third bars from the left in Figure 2), $t(38) = 1.73$, $p = .09$, $d = .65$. Given these effect sizes, we would expect these differences to be statistically different with a sample size that was larger than 20 participants per group.

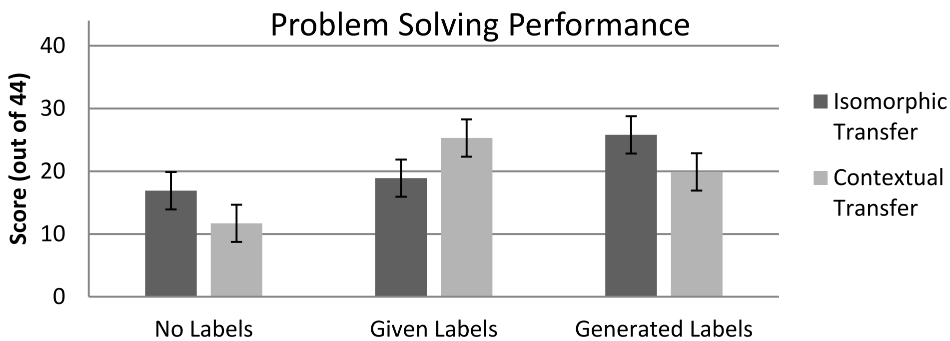

Figure 2. Performance for each group on problem solving tasks.

In summary, participants who received isomorphic transfer practice problems performed better than those who received contextual transfer unless they were given subgoal labels created by an instructional designer. This finding might be due to participants' mapping between worked examples and practice problems. In the isomorphic transfer conditions, it was obvious how the practice problems resembled the worked examples, allowing learners to easily apply the procedure from the example to solving the practice problem. Participants who received contextual transfer might have had difficulty mapping the example to the practice problem, which ultimately hindered learning unless they received subgoal labels that guided this transfer.

In general, participants who received subgoal-oriented instructions performed better than those who did not, suggesting that highlighting the subgoals of the procedure supported student learning. Which type of transfer was better for subgoal-oriented instructions depended on whether learners received or generated subgoal labels. Participants who received subgoal labels performed better with contextual transfer than with isomorphic transfer. This result might be due to contextual transfer allowing participants to build a more context independent understanding of the procedure, and receiving subgoal labels allowed participants to more easily map between the example and practice problems. In contrast, participants who generated subgoal labels performed better with isomorphic transfer than with contextual transfer. This result might be due to isomorphic transfer allowing participants to understand the connections between the examples and practice problems, making it easier to self-explain the subgoals of the procedure. Generating labels with contextual transfer might have overload cognitive resources enough to hinder learning.

Quality of learner-generated labels

We examined the subgoal labels that learners generated to explore the quality of labels that they produced. We used an iterative qualitative analysis in which we read a sample of participant responses to identify common themes then coded the data based on those themes (Braun & Clarke, 2006). We found there were two general types of labels: those including details that were specific to the worked examples and practice problems and those independent from the context. For instance, for the subgoal that initialized variables, two labels that were specific to the example in which participants calculated the average tip for a restaurant server are, "Establish container to hold tips" and "Create variable of tip values." Labels for the same subgoal that were not specific are, "Create variables," and "Define function and variables." We found that, overall, twice as many participants generated specific labels (n = 27) than general labels (n = 13). However, a larger percentage of participants who received contextual transfer (40%) generated general labels than those who had isomorphic transfer (25%).

We explored how the specificity of generated labels affected problem solving performance and interacted with the transfer distance manipulation. The following results include data from only the generate subgoal label groups (i.e., the rightmost groups on Figure 2). We found that participants who generated general labels (M = 25.8, SD = 13.4) performed better than those who generated specific labels (M = 19.8, SD = 13.7), F (1, 36) = 5.23, MSE = 144.6, p = .028, est. ω^2 = .13, f = .36. Similar to the general problem solving performance results, no main effect of transfer distance was found, F (1, 36) = .92, MSE = 144.6, p = .35, est. ω^2 = .02. Again, these results are tempered by an interaction, this time between specificity of labels and transfer distance.

We found an interaction between specificity of labels and transfer distance (see Figure 3), F (1, 36) = 5.52, MSE = 144.6, p = .024, est. ω^2 = .13, f = .37. There is not a difference between participants in the isomorphic transfer groups based on specificity of labels, t(18) = .04, p = .97. For the contextual transfer groups, however, participants who generated specific labels (M = 12.1, SD = 9.4) performed much worse than those who generated general labels (M = 31.4, SD = 10.7), t(18) = 4.22, p = .001, d = 1.9. To put these results in context, if we compare the scores of these two contextual transfer groups to the general problem solving performance results, the group that made specific labels performs on par with the lowest performing group (i.e., the unlabeled, contextual transfer group). In contrast, the group that made general labels performs six points (or 14% of the total score) better than the highest performing groups (i.e., the given labels, contextual transfer group and the generate labels, isomorphic transfer group).

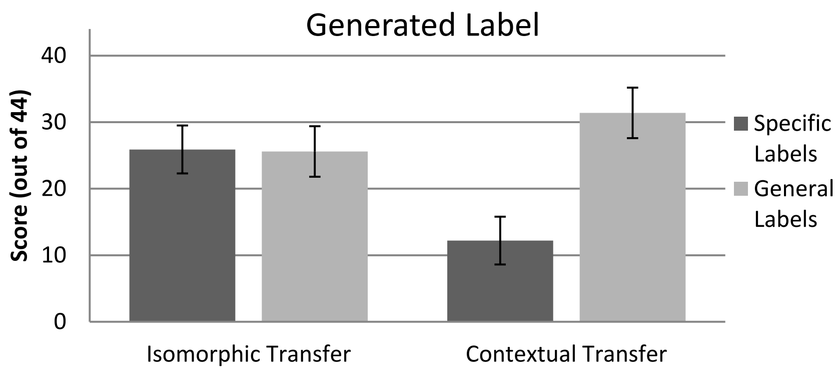

Figure 3. Performance for groups that generated subgoal labels on problem solving tasks split by transfer distance and specificity of generated labels.

In summary, participants who generated labels with isomorphic transfer performed relatively well, regardless of whether they created context-specific or general labels. For participants who generated labels with contextual transfer, however, their performance depends on whether they created specific or general labels. Those who created specific labels performed as poorly as the worst performing group, those who received no subgoal labels with contextual transfer. Participants in these groups were likely unable to discern the similarities between the examples and practice problems, which hindered their learning. On the other hand, participants who created general labels with contextual transfer performed better than any other group. This condition gave participants the most freedom to figure out the subgoals of the procedure for themselves, and if they were able discover the context-independent subgoals, then they were better able to solve new problems.

We explored whether these higher achieving participants were simply those students who perform well regardless of the learning conditions. We found that 40% of participants in the generate subgoal labels with contextual transfer created general subgoal labels. This percentage is higher than the typical 10% of students who perform well in all learning conditions (e.g., Chi et al., 1989). We also found that students were more likely to create general labels if they had a high college GPA, r_s = .44, p = .008, or high school GPA, r_s = .44, p = .01. Based on these results, we concluded that higher achieving students were more likely to be successful in this condition but in higher numbers than would be expected if the instructional intervention did not affect learning.

Time on task

We measured the amount of time that participants spent completing the problem solving tasks during the assessment. Those who received contextual transfer in the instructions completed the tasks faster than those who received isomorphic transfer (M = 16.9 minutes, SD = 10.8), F (2, 114) = 4.18, MSE = 78.9, p = .043, est. ω^2 =

.04, f = .18. There was no main effect for format of instructions, F (2, 114) = 0.38, MSE = 78.9, p = .69, est. ω^2 = .007. There was an interaction between the two manipulations, F (2, 114) = 4.11, MSE = 78.9, p = .019, est. ω^2 = .07, f = .18.

The pattern of results for time on task was almost identical to the pattern of results for problem solving performance. Participants who performed better took longer to complete the tasks. For example, participants who received contextual transfer finished the tasks more quickly and performed worse, except for those who received subgoal labels (see Table 2). The exception to this similar pattern was that participants who received no labels and isomorphic transfer took longer than other groups who performed better (i.e., groups that received labels with contextual transfer and that generated labels with isomorphic transfer). We examined the data for outliers that might skew the means, but we found no participants who spent a very short (i.e., less than 50% of the mean) or very long (i.e., more than 150% of the mean) amount of time on the tasks.

Based on these results, we conclude that completing the problem solving tasks correctly necessarily took longer than completing them incorrectly, but those who completed the tasks incorrectly devoted sufficient time attempting to achieve the correct answer. Alternatively, receiving contextual transfer during the instructions might have helped participants to apply their knowledge more quickly to the problem solving tasks, resulting in less time on task. Because these participants tended to perform worse on the tasks, we do not find this explanation likely.

Table 2. Time spent on problem solving tasks for each group (in minutes)

Group	No Labels	Given Labels	Generate Labels
Isomorphic Transfer	M = 20.0, SD = 12.9	M = 13.3, SD = 8.3	M = 17.5, SD = 9.9
Contextual Transfer	M = 10.7, SD = 6.2	M = 15.4, SD = 5.9	M = 14.7, SD = 8.0

Conclusions and implications

One of the biggest challenges in constructive learning is providing learners with enough support so they do not flounder but not so much support that they miss opportunities to construct knowledge. We found two types of instructional design that supported learning better than the others. Learners who received contextual transfer between worked examples and practice problems performed best when they were given meaningful subgoal labels. The labels likely guided the learners to recognize the similarities of the procedure between the two contexts. Learners who received isomorphic transfer performed just as well when they were guided to generate their own subgoal labels. This finding may be due to minimal transfer requiring less cognitive effort, allowing spare working memory capacity to be devoted to developing subgoal labels and improved learning.

Only a subgroup of another condition performed better than participants in these conditions: learners who received contextual transfer and generated subgoal labels that were not specific to the context of the worked examples and practice problems. This finding suggests that allowing learners who are capable of generating abstract labels with minimal guidance to do so is best for their learning. However, learners who were not capable of generating abstract labels performed as poorly as those in the lowest scoring condition. This result makes giving students this minimal amount of guidance risky, and until the factors that would predict success better are better understood, we do not recommend using this particular instructional design.

We recognize some limitations that affect the generalizability of our results. We did not measure learners' cognitive fatigue throughout the experiment, but we expect that it could be high, especially for those generating labels. Cognitively demanding tasks, such as constructing knowledge, could result in learners taking breaks during the learning process or becoming de-motivated, which might affect performance. Because we did not measure learner fatigue, break times, or related constructs, we do not know how they affected the results. In addition, for our analysis the quality of subgoal labels generated by participants, the sample size within each of those subgroups is small. For example, the number of people in the contextual transfer condition who generated context-independent labels was eight. Though the difference between these groups was large, it is possible that it is unreliable and that the actual effect size is smaller.

For future research, we plan to explore the factors that make students more or less successful in this paradigm that mixes subgoal learning and constructive learning. Perhaps we could improve the training for generating subgoals and, in turn, improve the learning of people who create their own labels. In addition, perhaps we could predict before learning begins which type of instruction will help the student to be most successful. For example, perhaps students with lower working memory capacity would perform the best when given subgoal labels and contextual transfer and students with higher working memory capacity would perform best when allowed to generate labels with isomorphic practice problems. Until we discover these predictive variables, we can conclude that learners can be successful when they generate their own subgoal labels, but only if they receive enough guidance from the instructions to support their constructive learning.

References

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.

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of the Educational Research, 70*(2), 181.

Bielaczyc, K., Pirolli, P. L., & Brown, A. L. (1995). Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem solving. *Cognition and Instruction, 13*(2), 221-252. doi:10.1207/s1532690xci1302_3

Bjork, R. A. (1994). Memory and metamemory considerations in the training of human beings. J. Metcalfe & A. P. Shimamura (Eds.). *Metacognition: Knowing about Knowing*. MIT Press: Cambridge, MA.

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

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, *3*(2), 77-101.

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. doi:10.1037/0096-3445.127.4.355

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.

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

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: Association for Computing Machinery.

Morrison, B. B., Dorn, B., & Guzdial, M. (2014). Measuring cognitive load in introductory CS: adaptation of an instrument. In *Proceedings of the Tenth Annual Conference on International Computing Education Research*, 131–138.

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

Renkl, A., Stark, R., Gruber, H., & Mandl, H. (1998). Learning from worked-out examples: The effects of example variability and elicited self-explanations. Contemporary Educational Psychology, 23, 90-108.

Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational Psychology Review*, *22*(2), 123-138.

Sweller, J., van Merrienboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational Psychology Review, 10*(3), 251-296.

Trafton, J. G., & Reiser, B. J. (1993). The contributions of studying examples and solving problems to kill acquisition. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society* (pp. 1017-1022). Boulder, CO.

Wittwer, J., & Renkl, A. (2010). How effective are instructional explanations in example-based learning? A meta-analytic review. Educational Psychology Review, 22, 393-409.

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.

Acknowledgments

We would like to thank the students who participated in the study and their instructors who gave us their time. We also thank the reviewers who supplied comments which improved this paper. This work is funded by the National Science Foundation under grant 1138378. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.