Journal of Interactive Online Learning **Volume 10, Number 2, Summer 2011**
www.ncolr.org/jiol **ISSN: 1541-4914**

Self-Explanation Prompts on Problem-Solving Performance in an Interactive Learning Environment

Kyungbin Kwon
Christiana D. Kumalasari
Jane L. Howland
University of Missouri

Abstract

This study examined the effects of self-explanation prompts on problem-solving performance. In total, 47 students were recruited and trained to debug web-program code in an online learning environment. Students in an open self-explanation group were asked to explain the problem cases to themselves, whereas a complete other-explanation group was provided with partial explanations and asked to complete them by choosing correct key-words. The results indicate that students in the open self-explanation condition (a) outperformed in a debugging task, (b) perceived higher confidence for their explanations, and (c) showed a strong positive relationship between the quality of their explanation and their performance. These results demonstrate the benefits of the open self-explanation prompts. Cognitive load of self-explanation and quality of explanation are discussed.

Self-explanation refers to a reflective activity explaining to oneself a learning material in order to understand facts from the material or to repair misunderstanding during studying worked-out examples or reading exploratory texts (Chi, Bassok, Lewis, Reimann, & Glaser, 1989). It seems obvious that a student performs better at problem-solving tasks, generates inferences which facilitate conceptual understanding, and repairs flawed mental models as well when being encouraged to use the self-explanation strategy during learning (Chi, 2000; Chi, de Leeuw, Chiu, & La Vancher, 1994). Other studies have corroborated the self-explanation effect from various domains such as mathematics (Siegler, 2002; Wong, Lawson, & Keeves, 2002), programming (Pirolli & Recker, 1994), biology (O'Reilly, Symons, & MacLatchy-Gaudet, 1998) and physics (Mayer, Dow, & Mayer, 2003).

The self-explanation effect can be explained with two fundamental reflection mechanisms: inference generation and conceptual revision (Chi, 2000). From the inference generation perspective, we expect to find learners who are able to induce information omitted from a text or explanation provided by an expert. When learners realize a gap between their current mental model and incoming information, they tend to infer new knowledge while explaining to themselves (Chi, et al., 1994). In a conceptual revision perspective, while students study an expository text with an initial flawed mental model, they may recognize a conflict between their mental model and the text. With the recognition of this violation, students intentionally take efforts to resolve the dissonance (Chi, 2000). Considering the internal process of representing an expert model and evaluating it, students are required to use metacognitive activities such as comparing their current mental model with the expert model and finding discrepancy between them.

In spite of its positive effectiveness, the self-explanation strategy needs to be supported with scaffolding to encourage students to use it in effective ways. At first, the reflection mechanisms require higher cognitive skills, but many students do not possess the skills (Chi, et al., 1989) or are reluctant to engage in learning by using their skills (Renkl, 1997). Not surprisingly, in the spontaneous self-explanation condition, only a few students (33%) engaged in generating explanations (Renkl's, 1997). According to cognitive load theory, generating self-explanation requires high cognitive load by requiring that learners monitor their understanding and represent incoming information at the same time (Sweller, 1988; Sweller, Van Merriënboer, & Paas, 1998). In order to engage and reflect on the learning process, learners need to adopt metacognitive strategies, which usually require more efforts and abilities. Cheshire, Ball, and Lewis (2005) examined self-explanation effects and found that self-explanation alone was not sufficient in analogical reasoning. Learners who were asked to generate explanations and were provided with feedback while solving problems outperformed learners who were only asked to explain themselves without feedback.

Self-Explanation Effect and Scaffolding

With concern for this issue, recently, many researchers have investigated the ways to support self-explanation. For example, Atkinson, Renkl, and Merrill (2003) implemented a fading instruction and self-explanation strategy while students learned from worked-out examples and revealed positive learning outcomes in an authentic classroom practice. The self-explanation prompts enabled students to reflect on which probability principles they used in solving the problems. Although the results supported self-explanation effects, there were some limitations in generalizing the outcomes because students in the self-explanation group received more information about the principles than the other groups.

Berthold and Renkl (2005) examined the effects of scaffolded self-explanation prompts in probability theory. In a computer-based learning environment, they manipulated three treatments: open self-explanation prompts, scaffolded self-explanation prompts, and no prompts. At the beginning, learners in the open self-explanation prompts group received a simple question prompt eliciting self-explanation (e.g., "Why do you calculate the total acceptable outcome by multiplying?"), and learners in the scaffolded self-explanation prompts group received "fill-in-the-blank" explanations (e.g., "There are ____ times ___ branches."). Next, they practiced an isomorphic example with only open self-explanation prompts. As a result, they found that learners in both the open and scaffolded self-explanation prompts groups outperformed those in a no prompts group. Moreover, scaffolded self-explanation prompts showed a notable effect on conceptual knowledge acquisition compared with open self-explanation prompts. Although it was true that the scaffolded self-explanation prompts conveyed additional information required to construct a mental model, the fading out method might enable learners to integrate the domain concept with their current mental models by reducing cognitive load in early learning phases (Renkl & Atkinson, 2003).

Aleven and Koedinger (2002) implemented an intelligent tutoring system and Geometry Cognitive Tutor (GCT) in a high school. The GCT provided feedback on the students' solutions as well as their explanations. As a result, students in the self-explanation group were encouraged to generate more explanations and outperformed a problem-solving group, especially in more difficult test items. The feedback on self-explanation is important in that it may help them to recognize faults if students did not have a correct mental model, besides providing corrective information on the domain. In this sense, the GCT enhanced the self-explanation process very well. Considering the process of self-explanation, the GCT helped students to reflect on their understanding by providing corrective information, a process opposed to Chi's (2000) reflection mechanisms of self-explanation in which students realize the gaps between their mental model and learning materials through an internal process rather than the realization being evoked by external information.

As we consider self-learning, it is instructive to examine the internal reflection process of self-explanation. However, there are only a few studies to date that investigate the internal reflection process of self-explanation and its effects on learning (e.g. Chi, 2000). Moreover, in previous research, self-explanation was examined while studying learning materials; however, there are few studies investigating the self-explanation effect while conducting problem-solving tasks. When studying expository texts or worked-out examples, students might focus on understanding domain knowledge. Solving problems, on the other hand, requires more cognitive efforts such as identifying problems, testing hypotheses, and finding solutions. The problem-solving performance will be promoted by reflection on one's problem-solving process. This study examined the effects of self-explanation on problem-solving performance.

Purposes of Study

The purpose of the present experiment is threefold. The first goal is to examine the effect of different self-explanation prompts on conceptual understanding and enhancing problem-solving performance. Previous research has confirmed the effects of self-explanation in comparison to no self-explanation prompt or other learning strategies, but effective ways to elicit self-explanation from students have not yet been confirmed. Berthold and Renkl (2005), for example, tested two types of self-explanation prompts in learning probability theory: open self-explanation prompts and scaffolded self-explanation prompts. While the scaffolded self-explanation prompts fostered conceptual understanding better than the open self-explanation prompts, one was not superior to the other for enhancing procedural knowledge. The current study extends Berthold and Renkl's investigation by applying two different self-explanation prompts into problem-solving processes. Comparing the effects of two prompts will provide meaningful insight into the instructional design field.

The second goal is to examine whether students provided with an open self-explanation prompt exerted more cognitive efforts while generating explanations and examining problems. According to a cognitive load theory, reducing extraneous cognitive load and increasing germane cognitive load is critical to support learning. In general, generating explanations requires more time and cognitive resources than reviewing explanations. This study explores whether the efforts contributed to learning.

The third goal is to investigate the quality of explanations elicited from different prompts. If students generated incorrect explanations, it might reduce the effect of self-explanation prompts or even hinder learning. So, one can assume that the quality of explanations has a close relationship with learning gains. In this vein, if the types of prompts affect the quality of explanations, this can be significant in designing self-explanation. Considering the reflection mechanism of self-explanation, it is also very useful to investigate how well students evaluate their explanations and how confident they are with them. Theoretically, students who evaluate their understanding correctly will benefit from the self-explanation processes (Chi, et al., 1989). High confidence on good explanations reveals students' accurate monitoring skills on their understanding, and vice versa.

Method

Participants and Design

The participants were 47 college students recruited from an Introduction to Web Development course at a research university in the Midwest. They received course credit (one point out of 100) for their participation. The participants were randomly assigned to one of two experimental groups (23 in a Complete other-explanation group and 24 in an Open self-explanation group). A one-factorial quasi-experimental design with two groups was conducted. The Complete other-explanation group was provided with partial explanations that contained drop down lists, and asked to complete them by choosing the correct key words. The Open self-explanation group was asked to generate explanations for a problem and provided with a "hint" which was opened by the participant's request.

Materials and apparatus

All participants used CatchBugs for learning and testing throughout the experiment. Because CatchBugs was developed on Flex 2.0, all participants could easily use the system on any web browser installed with Adobe Flash Player 10. CatchBugs was designed to deliver learning materials of programming concepts, to provide authentic problems for practicing debugging, and to test learning achievement in the domain of HTML web programming. While using CatchBugs, participants were required to acquire basic HTML concepts and to practice debugging processes. CatchBugs consisted of three learning modules: (1) Review of HTML Rules, (2) Practice of Debugging, and (3) Final Tests. Participants followed the learning sequence linearly from the first module to the last one. The following explains each learning step in detail.

Module 1: Review of HTML Rules. In this module, participants reviewed basic HTML concepts with HTML rules and tested their understanding at the end (Figure 1). Four HTML rules were presented as follows:

- Rule1: In XHTML, all elements must be closed.
- Rule2-1: When more than two tags are opened, they must be closed in the reverse order.
- Rule2-2: Some block-level elements (e.g. <p> tag) contain only inline elements. Some block-level elements (e.g. <title> tag) are not allowed to contain block-level (or inline-level) child elements.
- Rule2-3: Inline elements cannot be placed directly inside the <body> tag. They must be nested within block-level elements.
- Rule3: XHTML documents use only lower case for both tag and attribute names.
- Rule4: All values of attributes must be quoted.

The rules were demonstrated with instructor's explanations and with both correct and incorrect HTML syntax (Figure 1-a). On the same page, an example tab contained an authentic erroneous HTML code related to the rules. The HTML code was explained with "error messages" and instructor's "interpretation of the error messages" (Figure 1-b). By studying the error messages and the interpretation, participants learned the basic HTML concept. At the end of this phase, learners took a test to assess their understanding. The test consisted of eight multiple-choice questions that asked basic HTML concepts using simple HTML syntax (Figure

1-c). When a participant submitted answers, correct and incorrect answers were revealed. The test served as a pre-test in the current study.

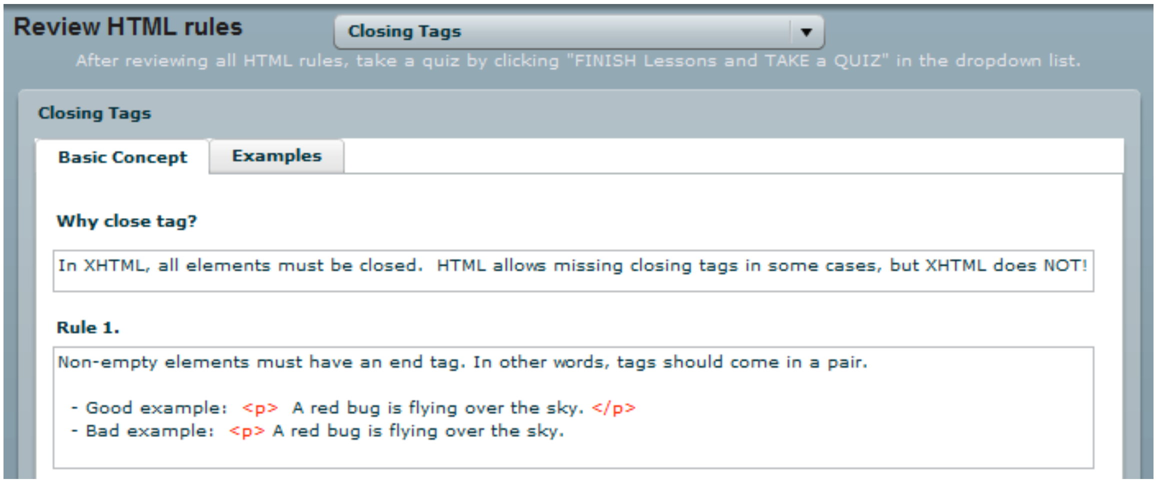

Figure 1-a. Example of HTML rules in CatchBugs.

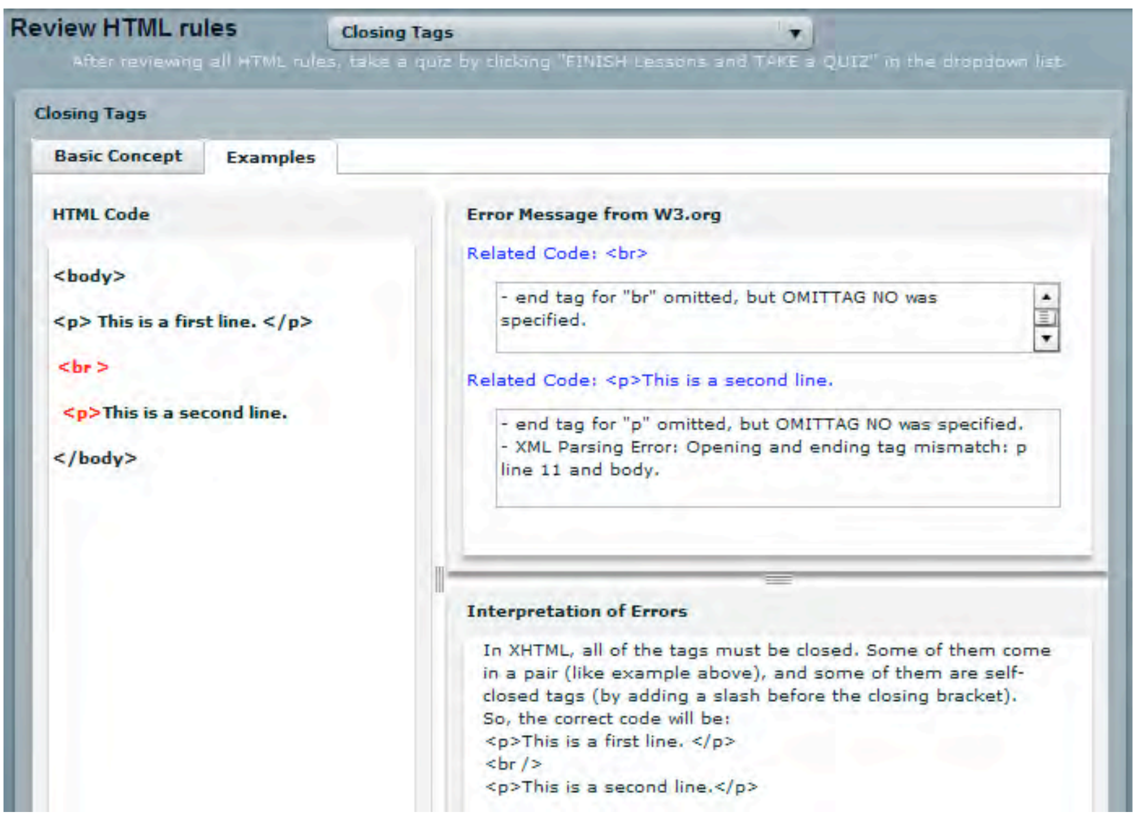

Figure 1-b. HTML code with “error messages” and “interpretation of the error messages.”

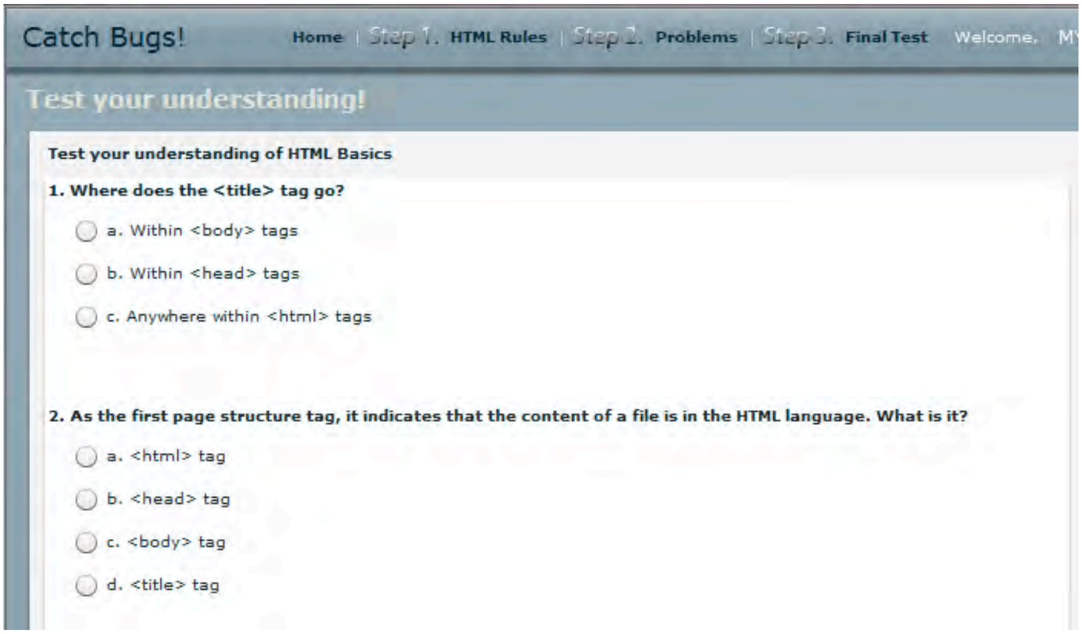

Figure 1-c. Test of basic HTML concepts.

Module 2: Practice of Debugging. In this module, participants applied the HTML concepts in authentic problems through debugging processes. There were four cases to study, so participants repeated the following procedure four times recursively. The debugging practice consisted of two activities: error detection and correction. In the error detection phase, the participants examined an HTML document and defined errors with the help of “error messages” explaining the error’s location and the reasons. Experimental difference took place in this phase by providing different self-explanation prompts. On the first page, participants were given a problem context and were presented with a screen shot illustrating the problem’s result (Figure 2-a.). Actual erroneous HTML code and error messages followed on the next page (Figure 2-b). The HTML code contained bugs that a novice programmer (an introductory level student) often commits when writing an HTML document. Error messages that were presented on the same page explained where the bugs were and why the errors occurred on the basis of HTML concepts. According to each participant’s experimental condition, Catchbugs ran one of two modes that reflected the conditions of this experiment.

In the Open self-explanation condition, participants received self-explanation prompts that asked them to explain the meaning of error messages and reasons for the errors as follows: “What does the error message mean?” “Why does the error cause problems?” Participants were required to type their explanations in an input textbox (see Figure 3-a.). After typing the explanations on the errors, participants were guided to monitor their competency. A metacognitive prompt asked how confident the participants felt about their explanation of the problems with a five-point Likert scale. When participants responded to the prompt, the error correction page followed.

In the Complete other-explanation condition, participants were presented with explanations generated by an instructor. The explanation contained a few drop-down lists consisting of keywords. Participants were asked to select one correct keyword from the drop-down list in order to complete the explanation (see Figure 3-b). Through the process, they might engage in learning as much as those in the Open self-explanation group. After completing the explanations, participants were guided to monitor their competency as the Open self-explanation group did and proceeded with the error correction phase.

In the error correction phase, participants were presented only HTML code without error messages and asked to fix the erroneous code. In this phase, no debugging aids were presented, so participants needed to re-locate bugs and correct them on the basis of their understanding.

The components of module 2 are described below. Problem statement and screen shots: A problem statement described the context of the current problems. From the problem statement, participants recognized symptoms of problems and the goal for correcting them. A screen shot showed the outcome of HTML code, which was displayed in a web browser. This visualization helped participants to understand the current problem status well (see Figure 2-a). HTML code window: It showed HTML code in a plain text mode (see Figure 2-b).

Error message window: It showed error messages that were generated from a w3 web validation tool (http://validator .w3. org/). One HTML error could cause multiple error messages. In this case, the messages were grouped in a chunk when they were presented in the error message window (see Figure 2-b).

Solution window: In this window, participants wrote final solutions. To help learners easily compare the two codes (problematic HTML and Solution), the solution window was accompanied by the HTML code window.

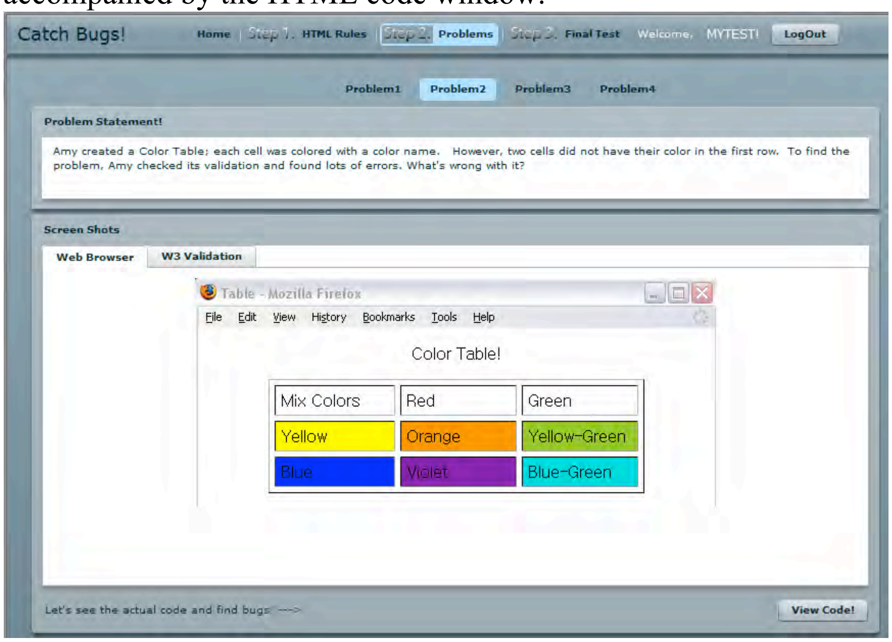

Figure 2-a. Problem context that presented with a screen shot illustrating the problem result.

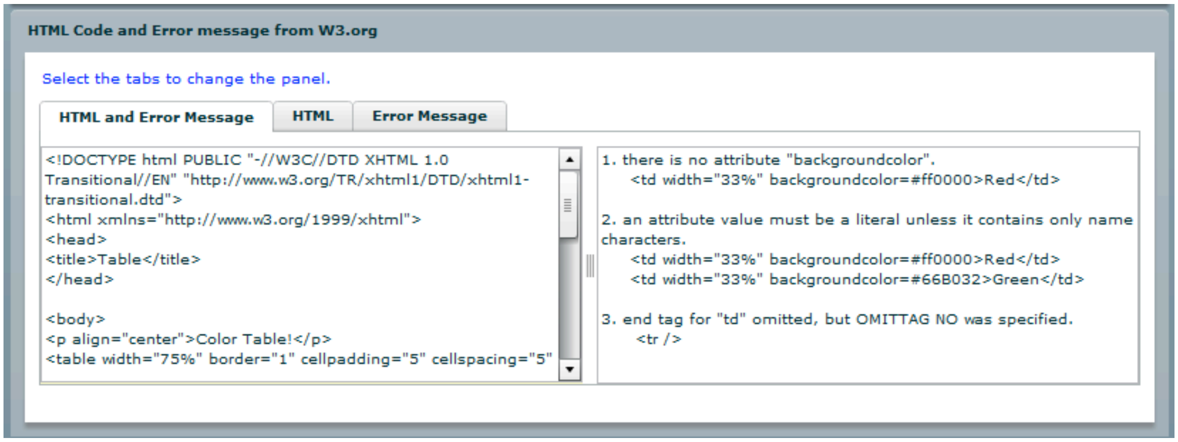

Figure 2-b. HTML code and error messages.

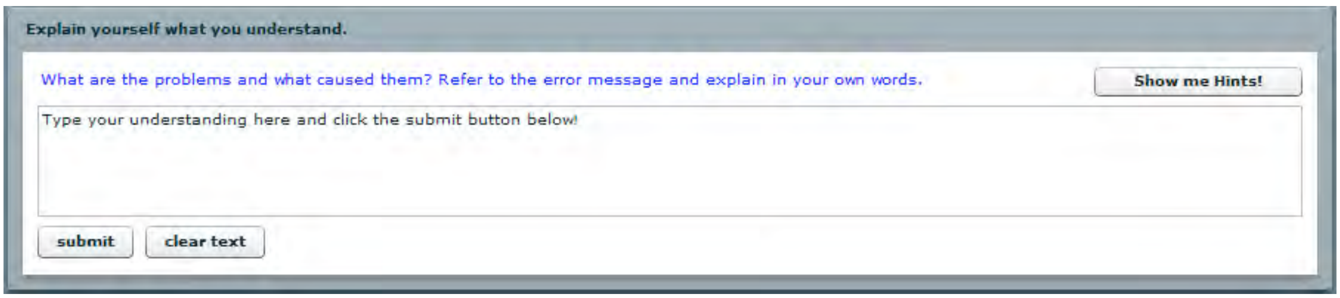

Figure 3-a. Example of Open self-explanation prompt.

Figure 3-b. Example of Complete other-explanation prompt.

Module 3: Final Tests. Two types of tests were implemented as a midterm examination of the course in which participants were enrolled. The tests served as a post-test in the current study. The post-test consisted of two tests: a multiple-choice quiz assessing domain knowledge and a debugging task evaluating debugging skills as a transfer test. The debugging task was evaluated with a rubric developed before the experiment. Details of the final tests will be introduced in the measurements section.

Procedure

All tests and learning took place as a part of coursework. Students studied the domain, basic HTML concepts and usages during the half semester (seven weeks) before the experiment. The experiment involved four phases: review domain knowledge, pre-test, debugging practice (experimental treatment), and post-test. All the phases were done within a web-based learning application, CatchBugs, and participants could regulate their learning pace within a one-week period.

First, an instructor informed participants that they would use CatchBugs to complete the midterm examination. One of the authors generated IDs and passwords and sent them to participants through a private emailing system in their online learning environment. A user manual of CatchBugs was provided as a PDF file before the experiment began. Second, when participants logged into CatchBugs, they were guided to follow the four learning phases as introduced in the previous section. All activities were done by the participants. The instructor/ teaching assistants helped participants only with the issues regarding the learning environment and did not answer questions regarding learning materials. Participants could monitor their learning progress on the CatchBugs web site. Thus, they could log out of the system and revisit it later to complete the task within the one-week period.

Measurements

Pre-test. An 8-item multiple choice quiz was developed to assess participants' entry level of domain (HTML rules) knowledge. The quiz consisted of three recall questions assessing the knowledge of HTML, three comprehension questions asking about correct usages of HTML, and two analysis questions requiring one to find errors in a chunk of HTML code (Bloom, 1956).

Confidence level. During the "Practice of Debugging" phase, participants were asked to report how confident they were regarding their understanding immediately after identifying errors. A five-point Likert scale was used to measure each participant's confidence level. The value served as an indicator illustrating the participant's perception of certainty regarding his or her learning.

Score of self-explanation. The explanations generated by participants who received Open self-explanation prompts were saved and assessed. The quality of explanation was evaluated with regard to the correctness of the explanation. One point was awarded for each category of errors that the participant correctly explained.

Score of explanation completion. The selection of keywords in the Complete other-explanation group was assessed. One point was awarded for each correct keyword.

Solution. The solution was assessed based on the correction of the erroneous HTML code. One point per error was awarded for the correct solution. Instructors developed the rubric of correct solutions before the study began.

Time. Time spent by each participant was saved as a log file. Using the log file, an experimenter could measure total time taken to complete each module.

Post-test. A 10-item multiple-choice quiz, which was identical to the pre-test in contents, was used to assess conceptual knowledge of HTML rules. The possible range of the score was 11 (0 to 10). A debugging task was developed to evaluate participants' debugging skills as a transfer test. The debugging task consisted of one HTML document containing twenty unique problems. The debugging task was similar to the problems presented in the "Practice of Debugging" module; however no debugging aids were presented in the test. The possible range of the score was 20 (0 to 19).

Results

Learning performance

Pre-test. In order to check prior knowledge of the participants, an analysis of variance (ANOVA) on the mean scores of the groups in the pre-test was carried out. Because all participants had already learned basic HTML concepts before the experiment, most students seemed to master basic HTML concepts and there were no differences in pre-test scores across the two conditions as illustrated in Table 1.

Table 1
Pre-test and post-test performance for Open self-explanation and Complete other-explanation groups

Measure (possible score)	Open self-explanation (n = 24)		Complete other-explanation (n = 23)			
	M	*SD*	*M*	*SD*	*F*-value	Effect size
Pre-test (8)	6.71	1.042	6.83	.650	.214	.138
Post-test						
-Debugging (19)	17.83	1.274	16.09	3.604	4.988*	.644
-Quiz (10)	9.13	1.116	8.96	1.364	.216	.136

Note 1. *, $p < .05$.
Note 2. Effect sizes calculated following Cohen (1988)

Immediate Learning through the Practice Debugging phase. During the Practice Debugging Process phase, participants solved four problem cases. Immediate learning performance was measured by assessing the solutions submitted for each problem case. An ANOVA on each case was performed on participants' solutions. Overall, there was no statistical difference between groups, $F(1,45) = .027$, $p > .05$. Table 2 illustrates means and standard deviations of each case across the groups.

Table 2
Learning performance through the Practice Debugging phase

Measure (possible score)	Open self-explanation (n = 24)		Complete other-explanation (n = 23)			
	M	*SD*	*M*	*SD*	*F*-value	*p*-value
Case 1(2)	1.33	.917	1.09	.848	.912	.345
Case 2(5)	3.21	2.245	3.17	2.103	.003	.957
Case 3(6)	3.54	2.502	3.35	2.347	.075	.786
Case 4(5)	2.96	2.136	3.09	2.130	.043	.837
Total(18)	11.04	7.410	10.70	6.983	.027	.870

Learning performance in the Debugging task and the multiple choice quiz. Participants' problem-solving skills as well as conceptual knowledge acquisition were measured in the Final Test phase. The test scores for the groups are shown in Table 1. These results indicated that there was a positive impact of the open self-explanation prompts on students' problem-solving performance. Although the groups achieved similar scores on the multiple choice quiz which assessed conceptual knowledge, their debugging performance differed substantially. The Open self-explanation group (M = 17.83, SD = 1.27) solved more errors correctly than the Complete other-explanation group (M = 16.09, SD = 3.6), $F(1,45) = 4.988$, $p = .03$ with a medium effect.

Time spent in learning phase and testing phase. Time spent in the Practice Debugging Process as well as the Final Test was measured by analyzing a log file. Table 3 shows the time spent in the phases across the groups. ANOVAs revealed that overall there was no substantial difference between the groups in time spent. In other words, the Open self-explanation group did not significantly spend more time than the Complete other-explanation group in either phase.

Table 3
Time (in seconds) spent in the Practice Debugging process and the Final Test phase

Measure	Open self-explanation (n = 24)		Complete other-explanation (n = 23)			
	M	*SD*	*M*	*SD*	*F*-value	*p*-value
Practice Debugging						
-Case 1	482.33	329.917	449.17	388.899	.100	.75
-Case 2	590.39	510.562	378.13	292.730	2.992	.05
-Case 3	702.08	444.316	681.78	721.240	.014	.91
-Case 4	577.63	466.177	653.43	387.749	.366	.55
Final Test						
-Debugging	3984.79	2730.587	4322.39	3636.458	.130	.72
-Quiz	688.88	883.777	590.52	360.134	.246	.62

Explanation quality and Confidence

Participants' explanations either generated or selected in the Practice Debugging Process phase were evaluated per error in each case. Because the possible scores of each case across the groups were not identical, the percentage of correct scores was used for comparison of the two groups. Overall, there was no significant difference between groups on the correctness of their explanations (see Table 4). In both conditions, students generated or selected incorrect explanations about one and one-half times out of ten chances.

Participants reported the confidence level on each explanation. A one-way ANOVA revealed that the Open self-explanation group showed higher confidence for their explanations (M = 18, SD = 2.43) than the Complete other-explanation group (M = 16.3, SD = 2.16), $F(1, 45) = 6.36$, $p < .05$.

Table 4
Percentage correct of explanation (or selecting keywords) and scores of confidence on the explanation (or the selection)

Measure	Open self-explanation (n = 24)		Complete other-explanation (n = 23)			
	M	*SD*	*M*	*SD*	*F*-value	Effect size
Percentage correct	86	15	85	10	.012	.079
Confidence scores	18.00	2.432	16.30	2.162	6.362*	.739

Note 1. *, $p < .05$.
Note 2. Effect sizes calculated following Cohen (1988)

In order to examine to what extent the two factors -- explanation and confidence -- were associated with the learning performance, a correlation analysis was conducted on the immediate learning scores computed in the Practice Debugging phase and the proportion of the correct explanations as well as the scores of confidence. Overall, the Open self-explanation group showed strong positive relationship between the correctness of explanation and learning performance, $r(23) = .82$, $p = .001$. However, the Complete other-explanation group did not show a significant relationship between them. Interestingly, the confidence level did not show significant relationship with learning performance in either group.

Table 5
Correlation matrix for correctness of explanation, confidence, and learning performance

	1	2	3
Open self-explanation (n = 24)			
1. Correctness of explanation	-	-.02	.82**
2. Confidence		-	-.04
3. Learning performance			-
Complete other-explanation (n = 23)			
1. Correctness of explanation	-	.23	.41
2. Confidence		-	-.20
3. Learning performance			-

Note 1. **, $p < .01$.

Discussion

This study investigated the different effects of self-explanation prompts on learning web programming concepts and enhancing problem-solving performance. The main results of the present study can be summarized as follows: (1) Generating an explanation to oneself was more effective than completing an explanation provided by an expert on problem-solving performance. (2) Generating an explanation did not require higher cognitive efforts compared to completing an expert's explanation that omitted key words. (3) Student who explained reasons for problems to themselves perceived more confidence in their learning than those who completed an expert's explanations that omitted key words. (4) When students generated more correct explanations on problems, they solved the problems more accurately.

Superiority of open self-explanation

The results revealed the superiority of "open self-explanation" over completing an expert's partial explanation. When students freely generated explanation of problems, they could apply their skills into solving a new problem more effectively than students who were provided with a pre-made explanation. There may be a considerable difference between "self-explanation": generating an explanation and judging its validity and "completing other-explanation": receiving another's explanation and understanding it. Although the current study could not fully explain the reason for superiority of the "open self-explanation", the reflection mechanism of self-explanation may shed light on interpreting the results (Chi, 2000). We cannot deny the fact that both the open self-explanation and complete other-explanation strategies allow students to actively engage in the problem-solving process. However, the approaches of the two methods were opposed: whereas, the "open self-explanation" allowed students to engage in finding reasons for problems by themselves and to reflect on their explanations, the "completing other-explanation" had them examine the pre-made explanation first and required them to understand the problems on the basis of that explanation. According to Chi's (2000) description of internal self-explanation processes, the "open self-explanation" might encourage students to generate more inferences to make sense of their explanations and the inferences could be elements of new knowledge. However, the "completing other-explanation" might force students to make sense of a given explanation and thus hinder inference generation. Therefore, the outperformance of the open self-explanation group in the new problem-solving tasks may be due to the inference generation activities.

At a glance, the completing other-explanation method seemed to hinder learning. However, we cannot generalize the results hastily because there were a few limitations. First, there was no feedback on students' learning activities: selecting correct or incorrect keywords. When students chose incorrect key words, they could not recognize that the answer was not correct. In contrast with the current results, Berthold and Renkl (2005) revealed the effect of pre-made explanation on conceptual knowledge acquisition. However, they provided multiple-representational solution procedures, which guided the student to fill in the blanks of an incomplete explanation. Thus, the effect of pre-made explanations could be mainly ascribed to the conceptual representation rather than the "fill-in-the-blank" activities. Second, selecting key words might be a more challenging task than we intended. The pre-made explanation was straightforward and the key words were clear to find the correct answer. However, overall 15% of key words selected by participants were incorrect. Although the main purpose of the "select-key words" activities was to engage students in the learning process, it might hinder them from understanding given explanations. The benefit of engagement would be lost if students selected wrong key words and then proceeded with the next phase without correcting their misconceptions. Simple corrective feedback might have provided a chance to reflect on the incorrect answers.

Cognitive load of self-explanation

According to the results, the open self-explanation did not require higher cognitive load than the complete other-explanation did. Considering the intrinsic cognitive load, the interactions within the problems were considerable because one program error might be related to others. Students in the open self-explanation condition might suffer from the higher interactions between the complexity of problems and the demands of monitoring their understanding. On the other hand, providing a partial explanation could reduce extraneous cognitive load by reducing the size of the problem space (Van Merriënboer & Sweller, 2005). Therefore, we assumed that students who were required to generate explanations without being given partial explanations, would spend a great deal of time in completing the process. However, in contrast to the expectation, the results revealed that students in the open self-explanation condition spent as much time as those in the complete other-explanation group. The results supported the possibility that self-explanation activities were not as demanding as instructors assumed. As a critical limit, however, we could not measure the cognitive load directly. A future study measuring the cognitive load of self-explanation might explore and confirm the results.

Relation between quality of explanation and learning performance

The quality of self-explanations during a learning phase affect students' learning performance later. When students spontaneously generate more self-explanation during studying learning materials, they can solve more problems (Chi, et al., 1989; Pirolli & Recker, 1994). Even when a student is enforced, or trained to explain to oneself, he or she will have the benefits of self-explanation (Bielaczyc, Pirolli, & Brown, 1995; Chi, et al., 1994). Appropriate frequency of generating explanations while studying is a necessary but not sufficient condition of learning gains. An effective self-explanation refers to a principle which can elaborate domain knowledge (Renkl, 1997). Thus, students should explain learning materials with consideration of underlying principles and the corresponding sub-goal to engage in meaningful learning. The current study supported these arguments in the domain of web programming and extended the findings by assessing the correctness of self-explanation. The more a student generated correct explanations while studying a problem case, the better he or she solved the problem. Students also revealed higher confidence on their explanation from the Open self-explanation condition. Especially, the contents of explanations generated by students not "given and completed" represented students' understanding well. Thus, the quality of self-explanation could be a good indicator assessing a student's conceptual model.

Conclusion

The current study has educational significance in that self-explanation strategies were utilized in an online learning environment, which typically requires that students be self regulated and centered in the learning process. Additionally, the study examined the self-explanation effects in a condition where students were asked to write explanations rather than verbalize them. Although the study provides answers to the research questions, there are still many significant questions to be addressed in future research. For example, are written explanations superior to verbal explanations and, if so, in what ways? More generally, how can we effectively embed support of student metacognition in online courses by use of self-explanation strategies? Considering the unique challenges of online distance learning environments and the characteristics of learning activities, students can benefit by being taught and guided in using the self-explanation strategy in effective ways. Course designers and online instructors might do well to support student learning by including work that integrates this type of metacognitive exercise as they remain cognizant of emerging research in the field.

References

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

Atkinson, R. K., Renkl, A., & Merrill, M. M. (2003). Transitioning From Studying Examples to Solving Problems: Effects of Self-Explanation Prompts and Fading Worked-Out Steps. *Journal of Educational Psychology, 95*(4), 774-783.

Berthold, K., & Renkl, A. (2005). *Fostering the understanding of multi-representational examples by self-explanation prompts.* Paper presented at the 27th Annual Conference of the Cognitive Science Society.

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.

Bloom, B. S. (1956). *Taxonomy of educational objectives. Handbook I: Cognitive domain. Handbook II: Affective domain.* New York: David McKay.

Cheshire, A., Ball, L. J., & Lewis, C. N. (2005). *Self-explanation, feedback and the development of analogical reasoning skills: Microgenetic evidence for a metacognitive processing account.* Paper presented at the 27th annual conference of the cognitive science society.

Chi, M. T. H. (2000). Self-explaining Expository Texts: the dual processes of generating inferences and repairing mental models. In R. Glaser (Ed.), *Advances in instructional psychology* (pp. 161–238). Mahwah, NJ: Lawrence Erlbaum Associates.

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*(2), 145-182.

Chi, M. T. H., de Leeuw, N., Chiu, M.-H., & La Vancher, C. (1994). Eliciting Self-Explanations Improves Understanding. *Cognitive Science, 18*, 439-477.

Mayer, R. E., Dow, G. T., & Mayer, S. (2003). Multimedia Learning in an Interactive Self-Explaining Environment: What Works in the Design of Agent-Based Microworlds? . *Journal of Educational Psychology, 95*(4), 806-812.

O'Reilly, T., Symons, S., & MacLatchy-Gaudet, H. (1998). A Comparison of Self-Explanation and Elaborative Interrogation. *Contemporary Educational Psychology, 23*(4), 434-445.

Pirolli, P., & Recker, M. (1994). Learning Strategies and Transfer in the Domain of Programming. *Cognition and Instruction, 12*(3), 235-275.

Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. *Cognitive Science, 21*(1), 1-29.

Renkl, A., & Atkinson, R. K. (2003). Structuring the Transition From Example Study to Problem Solving in Cognitive Skill Acquisition: A Cognitive Load Perspective. *Educational Psychologist, 38*(1), 15-22.

Siegler, R. S. (2002). Microgenetic studies of self-explanations. In N. Granott & J. Parziale (Eds.), *Microdevelopment: Transition processes in development and learning* (pp. 31-58). New York: Cambridge University.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science, 12*(2), 257-285.

Sweller, J., Van Merriënboer, J. J. G., & Paas, F. (1998). Cognitive Architecture and Instructional Design. *Educational Psychology Review, 10*(3), 251-296.

Van Merriënboer, J. J. G., & Sweller, J. (2005). Cognitive Load Theory and Complex Learning: Recent Developments and Future Directions. *Educational Psychology Review, 17*(2), 147-177.

Wong, R. M. F., Lawson, M. J., & Keeves, J. (2002). The effects of self-explanation training on students' problem solving in high-school mathematics. *Learning and Instruction, 12*(2), 233-262.