Why Do Elaborative Dialogs Lead to Effective Problem Solving and Deep Learning?

Robert G. M. Hausmann (bobhaus@pitt.edu)
Department of Psychology and the Learning Research and Development Center
University of Pittsburgh, 3939 O'Hara Street, Pittsburgh, PA 15260

Abstract

Previous research has demonstrated a correlation between learning and elaborative dialogs. To better understand that relationship, the present experiment contrasted performance in three conditions: individuals, a control condition that did not receive communication training, and an elaboration condition. After a pretest, participants were asked to solve a conceptual engineering problem, which required them to optimize a pre-existing bridge structure. Participants iteratively edited their design, analyzed its cost and effectiveness, and discussed their analyses to formulate their next modification. Immediately thereafter, a posttest measuring both shallow and deep learning was administered. The elaborative condition generated better optimized designs and learned more deep knowledge than the control condition. The elaborations led to shorter negotiations about which design modification to try next. This may explain the increased learning in the elaborative condition because more appropriate learning events occurred. I argue that the positive outcomes associated with elaborative dialogs can be attributed to the efficient production of collaborative contributions at a relatively high level of specification.

Keywords: collaboration, elaboration, learning, problem-solving

Introduction

Past research on collaborative problem solving and learning has painted a fairly consistent picture of both its costs and benefits. For instance, collaboration seems to be an effective educational intervention; however, not all collaborative dialogs lead to positive outcomes. Instead, only certain dialog patterns tend to result in strong learning gains. For example, generating explanations tends to be associated with understanding (Chi, Bassok, Lewis, Reimann, & Glaser, 1989) while paraphrasing does not (Hausmann & Chi, 2002). Therefore, one of the goals of the learning sciences is to identify dialog patterns that tend to produce strong learning gains (Dillenbourg, Baker, Blaye, & O'Malley, 1995). One such candidate is elaborative dialogs (Brown & Palincsar, 1989). Elaboration is a likely candidate because it has been shown to support individual learning. For instance, elaboration has been shown to enhance memory of sentences (Stein & Bransford, 1979), increased comprehension of prose (Anderson & Reder, 1979) and instructional texts (Reder, Charney, & Morgan, 1986).

Given elaboration's utility for individual learning, it seems natural to extend it to collaborative learning. For example, van Boxtel, van der Linden, and Kanselaar (2000) investigated the effects of collaboratively producing a concept map. Although they found a significant correlation between the definition posttest score and elaborative episodes, elaborative episodes did not correlate with a deeper measure of learning (i.e., essay questions). One way to explain their results is to look at the task demands. Concept maps lend themselves to discussions of the links between individual concepts, which suggests that the individuals would have formed detailed representations of the concepts (i.e., definitions) through elaborative activities.

Stark, Mandl, Gruber, and Renkl (2002) extended these results by providing evidence of deep learning as a result of producing elaborations. They trained participants to give elaborations while studying worked-out examples. Individuals were categorized into different elaboration profiles based on their think-aloud protocols. Stark et al. found that individuals who displayed an active, meta-cognitive orientation to the elaboration activity performed better on deep, far transfer problems than individuals with profiles that were more passive and superficial. This finding is encouraging because it suggests that elaborative activities can be trained and, more importantly, that they lead to deep learning outcomes.

Elaboration may also enhance problem solving by providing high levels of evidence for accepting a contribution. During communication, partners display evidence of understanding "to a criterion sufficient for current purposes" (Clark & Brennen, 1991; p. 129). Displaying evidence of accepting a partner's contribution is a continuous function, with low levels of evidence on one end of the continuum (e.g., continued attention) and high levels on the other (e.g., collaborative completion). Finishing another person's statement, a phenomenon referred to as collaborative completion, provides strong evidence for a high level of acceptance. Consider the following exchange from a banking simulation (see Table 1, from McGregor & Chi, 2002). Cass presents part of a contribution, which her partner finishes. Dana's completion signals to Cass that she has understood and accepted Cass's contribution.

Table 1. Example of a collaborative completion

Speaker	Contribution	Proposition
Cass:	Okay, the new system would give the- give the employees...	GIVE(SYSTEM, EMPLOYEES, *x*)
Dana:	...more time to deal with the customers.	DEAL(TIME, CUSTOMERS)

Furthermore, elaborative dialogs may also increase the specification of individual contributions by assigning values to unfilled variable assignments. To illustrate how this works in an actual conversation, consider the brief exchange in the banking simulation. If we represent Cass's contribution as a proposition (see the right column in Table 1), we see that she leaves an unfilled variable (call it *x*). The verb "to give" requires three variable assignments: a) a subject doing the giving, b) an object, and c) a recipient of the object. Dana supplies the missing information for *x* in her turn. Her contribution can be thought of as an elaboration of Cass's incomplete message. The hypothesis evaluated in the current study is that elaboration serves to increase the specification of another person's message, which can then lead to efficient problem solving.

Method

Participants

Volunteers for the current study were undergraduate psychology students who received course credit for their participation. Participants were randomly assigned to one of three conditions: individuals (*n* = 20), control dyads (*n* = 39),[1] and elaborative dyads (*n* = 19). The sample size for the dyads represents the number of pairs in each condition.

Materials

Simulation. The domain chosen for the experiment was a design task. Participants were asked to design a virtual bridge using a pre-existing software package (West Point Bridge Designer 2003). The domain of bridge design was selected for several reasons. First, bridge design is an open-ended task that does not have a strictly correct answer, yet it affords several dependent measures to help quantify designs in terms of their quality. Second, the open-ended nature of the task makes it highly engaging, which ensured that the individuals would interact while solving the problem. This, in turn, allowed the experimenter to manipulate and analyze the communication patterns. Finally, working within a simulated environment allowed for the acquisition of both procedural and declarative knowledge.

Measures. To evaluate problem solving, the number of bridges tested was collected for each dyad (i.e., *iterations*). Furthermore, the amount of money each dyad saved (i.e., *savings*) was calculated by subtracting the price of their final bridge from the starting price. Finally, an optimization score was calculated by summing the stress-to-strength ratios for each member (i.e., bridge beam), which was then divided by the total number of members:

$$optimization = \frac{\sum_{i=1}^{n}\left(stress_i / strength_i\right)}{n} \quad \textbf{(1)}$$

where *n* = the number of members per bridge, *stress* is the amount of force loaded on the *i*-th member of the bridge, and *strength* is the maximum loading the *i*-th member can withstand before failure. The optimization score represents the average load per member; thus, higher values indicate better optimized designs.

Learning was assessed at two levels of depth: *shallow* and *deep*. Shallow learning was defined as the acquisition of information explicitly stated in the text. In contrast, concepts that were not explicitly stated, and thus needed to be inferred from reading the text or interacting with the simulation, comprised deep learning. Gain scores, calculated separately for both shallow and deep assessment items, were designed to measure the acquisition of knowledge from the entire task and not a direct measure of the final design *per se*.

Text. The text was written by abstracting relevant propositions from a companion text to the software (Ressler, 2002). The information selected to be in the text was easy for the participants to assimilate, yet would take time to learn from the simulation (i.e., by conducting several small experiments). Therefore, the purpose of the text was to expedite their learning declarative knowledge from the simulation.

Training instructions. The training instructions for the elaborative dyads were written to be as concise as possible to reduce the load on working memory. Acronyms were used such that the participants could easily remember to produce an **I**dea, **E**laborate upon the idea, or **R**espond to an idea (IER). The instructions were based upon the definition of elaboration in the literature (Hogan, Nastasi, & Pressley, 1999). There was an added emphasis on the idea that incomplete ideas should be made explicit, which was derived from the hypothesis that elaborations supply variable assignments (see Introduction).

Procedure

The experimental design contained one experimental condition and two control conditions. For the experimental condition, the dyads were instructed to engage in elaborative interactions using the instructions described above. The first control condition was a simple baseline condition in which individuals completed the task without a partner or any communicative manipulation. The second control condition included dyads solving the problem together, without any communicative manipulation. Except for communication training, all other aspects of the experiment were identical for each condition.

For the procedure, participants were prescreened for taking civil engineering and material science courses; after which an on-line pretest was administered (for a copy of the materials, see Hausmann, 2005). The pretest was designed to measure prior knowledge of bridge design and material science. Once the pretests were completed, the participants individually read a short text.

[1] The sample sizes were uneven because an evaluation dyad condition, which did not differ from the control condition on any dependent measure, was combined with the control dyads (Hausmann, 2005).

For the elaboration condition, instructions were given for engaging in a specific type of dialog. After reading the instructions, the experimenter answered any questions the participants had. To make the instructions concrete, the participants were given a warm-up task in which the experimenter listened to participants interact and intervened when necessary. The procedure for teaching the communication scripts was adapted from Renkl et al. (1998).

After the participants completed the warm-up task, they were then introduced to the software by watching a short movie demonstrating the tools and features of the interface. The experimenter then verbally introduced the task and answered any remaining questions. The participants were given a pre-existing bridge structure and were told that their goal was to optimize the design by making it as cheap as possible, while still being able to carry a load. The participants were given 30 minutes to complete the task. Their dialog was videotaped for later transcription.

The product generated by the participants was a single bridge design, which was assessed along two dimensions: savings and the optimization score. The total fabrication cost of the bridge was continuously displayed for the participants as an indication of their progress (see the Appendix for a screenshot of the software).

A posttest, which was identical to the pretest, was administered individually to measure how much information was learned from the text, the simulation, as well as from their interactions. Upon completion, the participants were debriefed, thanked, and excused from the experiment.

Analyses and Results

Coding scheme

A manipulation check was conducted to ensure that the communication instructions had their intended effects. A stratified sample of the control (n = 16) and elaborative (n = 8) dyads was taken such that both good and poor performers were equally represented.

Three types of elaborative statements were coded. The first type, *elaborate suggestion*, modified the speaker's initial suggestion by providing a location, an additional change, or a specific value for the proposed modification. The second type, *provide reason*, gave a justification for a particular change. The last type, *provide implication*, gave a consequence for a particular change. The average frequencies for each code, broken down by condition, are summarized in Table 2.

Although the difference between the elaborative and control condition for the mean number of total elaborative statements did not reach traditional levels of statistical significance, $F(1, 22) = 2.69$, $p = .12$, $d = .73$, the interpretation of the effect size suggests there was a medium to large effect (Cohen, 1988). Furthermore, there was a strong correlation between the total number of elaborative statements and the savings ($r = .86$, $p = .006$) and optimization score ($r = .91$, $p = .002$) for the elaborative condition; thus, replicating previous findings that elaborative dialogs are useful for problem solving.

Table 2. Manipulation check: Elaborative dialogs

	Condition			
	Control Dyads		*Elaborative Dyads*	
Elaborate Suggestion	8.12	(5.91)	12.12	(4.97)
Provide Reason	0.12	(0.34)	0.25	(0.71)
Provide Implication	0.69	(0.87)	1.00	(1.41)
Total Elaborative	**8.94**	**(6.46)**	**13.38**	**(5.76)**

The effect of elaborative interactions

Problem solving. There was an effect of condition on the number of iterations, $F(2, 75) = 3.45$, $p = .04$. Post-hoc analyses revealed a reliable difference between the individuals ($M = 49.55$, $SD = 23.36$) and the control dyads ($M = 36.79$, $SD = 13.35$), but no difference between the elaborative dyads ($M = 41.00$, $SD = 18.62$).

There was a marginal effect of condition on optimization score, $F(2, 75) = 2.60$, $p = .08$. Post-hoc analyses revealed a reliable difference between the elaborative dyads and control dyads, $d = .61$, but no difference between the individuals (see Figure 1). Elaborating a partner's ideas and suggestions increased the dyads' ability to optimize their designs. On the other hand, there was no effect of condition on savings, suggesting all conditions constructed equally priced bridges, $F(2, 75) < 1$.

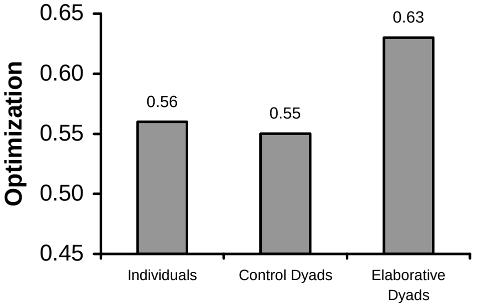


Figure 1: Average optimization scores for the control and elaborative dyads.

Learning. As expected, there was no effect of condition on text-explicit knowledge, suggesting that participants in all conditions learned roughly the same amount of shallow knowledge, $F(2, 133)^2 < 1$.

In contrast, there was a main effect of condition on inferential learning gains (see Figure 2), reflecting a higher score for the elaborative dyads ($M = 11.18$, $SD = 18.81$)

[2] The degrees of freedom differ from the prior analysis because problem solving was evaluated at the group level, while learning was assessed individually.

than the control dyads (*M* = 3.31, *SD* = 15.33), *F*(2, 133) = 3.08, *p* < .05.

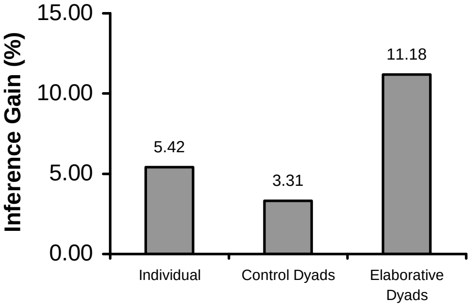


Figure 2: Average inference gains for the control and elaborative dyads.

The communication efficiency hypothesis

Why did the elaboration condition perform better and learn more deep knowledge than the control condition? There are three interrelated hypothesized explanations: 1.) communication efficiency, 2.) effective use of simulation feedback, and 3.) problem-solving strategy differences.

Communication efficiency. The first hypothesis is related to the way in which the elaborative partners interacted. Communication efficiency is an obvious choice because it is the only dimension in which the two conditions differed. Elaborating upon another person's idea may increase the specificity of each suggested modification. Dyads devoted several turns to negotiate which action to take. In the present task, there are at least three variable assignments that need to be made. First, a change needs to be specified (e.g., change the type of steel), then a specific value needs to be specified (e.g., from carbon steel to high strength low alloy steel), and finally the location needs to be identified (e.g., member #12). Values for each of these variables can take multiple turns to establish. An elaborative sequence may come to establish the variable assignments more quickly than one in which the partner asks the other to specify the variable assignments.

To test the efficiency hypothesis, clarification questions were coded in the elaborative and control conditions. Clarification questions were selected because the goal of the question is to establish the variable assignments, which is illustrated in the following exchange (see Table 3). Beth makes the suggestion that they switch some members to hollow beams (turn 82). Abby asks which solid members should be changed to hollow (83), to which Beth replies that the top member should be changed (84).

Table 3. Clarification question example

Turn	Speaker	Contribution	Code
82	Beth:	So these are ah solid tube. Alright, so we can make some of these hollow.	
83	Abby:	The top ones or the…	clar_q
84	Beth:	The top.	
85	Abby:	Like every other one?	clar_q
86	Beth:	Try it…	

In contrast, consider an elaborative episode (see Table 4). In this brief exchange, Mike proposes that they make the diagonal members a smaller diameter (turn 82). Dan accepts Mike's proposal, and elaborates it by suggesting a location (i.e., the middle, 83). They are able to make a suggestion and implement it relatively quickly because they can avoid the need to ask for clarification. In addition, they are building off one-another's ideas, instead of merely fleshing out a single partner's idea.

Table 4. Elaboration example

Turn	Speaker	Contribution	Code
82	Mike:	Cause usually, I don't know, do you want to try making the cross members smaller?	
83	Dan:	Um, we could,- just the ones in the middle not the ones on the end.	elab
84	Mike:	Yeah, right.	

The elaboration condition (*M* = 18.00, *SD* = 9.71) generated marginally fewer clarification questions than the control condition (*M* = 26.37, *SD* = 11.32), *F*(1, 22) = 3.19, *p* = .09, *d* = .79. Because the elaboration condition asked marginally fewer clarification questions, this suggests that the communication instructions had a direct effect on their dialog, which, in turn, had an indirect effect on their problem-solving performance.

Simulation feedback. The second hypothesized explanation involves the way in which dyads used the feedback from the simulation. The most useful feedback for the present task was found in two different sources. The first is the color-coded feedback, which was superimposed over the individual members, when the user tested a bridge. The magnitude of the tensile and compressive forces is encoded in the simulation as a continuous change in color intensity in the drawing. (For an example, refer to the screenshot in the Appendix. The members in the center of the bottom cord are darker blue than the members at the ends, indicating higher levels of tension.) The second source is the Member List in which the information is presented as a numerical ratio of the member's strength to impressed force. An example of using simulation feedback in a dialog can be found in the following exchange (see Table 5).

Table 5. Simulation feedback example

Turn	Speaker	Contribution
168	Ben:	Let's see if we can break it. Then we can change them all, some of these.
169	Nathan:	Hm-mmm
		[Test design: iteration 13]
170	Ben:	Find one that's getting really **red**.
171	Nathan:	Still not that **red**, right?
172	Nathan:	This vertical one too, and this vertical one. They're still **pinkish**.
173	Nathan:	We could make them thinner.

In this example, Ben sets the goal to find a member that is not experiencing much compression (turn 170). They use this information to select a member to change. Once the change was made, they observed the effect by looking at the intensity change (171). This pair used this information to suggest a location (i.e., a particular member), as well as to make specific changes (i.e., cross-sectional diameter, 173).

To test the hypothesis that the elaborative condition was more effective in exploiting the feedback provided by the simulation, the explicit mention and use of the feedback was coded. The elaborative condition made explicit mention and use of the simulation feedback more times than the control condition (see Table 6), $F(1, 22) = 4.69$, $p = .04$, $d = .80$. Because the feedback from the simulation was an effective cue for redesigning a cheaper and more optimized bridge design, feedback utilization may help explain why the elaboration condition performed better and learned more than the control condition. For instance, the correlation between color-coded feedback and optimization score for the elaboration condition ($r = .64$, $p = .09$) was stronger than the control condition ($r = .30$, $p = .26$).

Table 6. Average frequency of explicit mentioning of simulation feedback

	Condition			
	Control Dyads		*Elaborative Dyads*	
Color-coded FB	3.50	(4.10)	8.75	(11.12)
Member-list FB	0.00	(0.00)	1.12	(2.47)
Total Feedback	**3.50**	**(4.10)**	**9.87**	**(10.45)**

Note. Standard deviations are in parentheses.

Problem-solving strategies. The third explanation is derived from the types of strategies each condition employed. Dyads were categorized as employing either an effective or ineffective strategy. Strategy effectiveness was determined through a task analysis. The companion text to the software made a few recommendations for optimizing a truss, which were then supplemented with optimization strategies developed through protracted interactions with the simulation by the experimenter. The following strategies were coded (see Table 7). Positive strategies included: (1) identifying individual members with either a high or low amount of stress, which is helpful because it can guide dyads toward (or away from) modifying certain members; (2) only manipulating one feature or property during a single iteration; (3) finding the point at which a member fails, which helps locate each member's optimized load, and (4) making a single change and observing the resulting change in price, which also helps with the task of making a bridge as cheap as possible. Negative strategies included: (5) removing members or joints; (6) adding superfluous members or joints to the existing structure (this is only effective in very few circumstances); and (7) using longer members (because strength decreases as members increase in length, especially for members under compression).

The control dyads were more likely to be classified using a negative strategy than the elaborative dyads, $\chi^2(1) = 4.11$, $p = .04$. Additionally, the elaborative dyads were marginally more likely to be classified as using a positive strategy, $\chi^2(1) = 3.00$, $p = .08$. Therefore, it seems that, at minimum, the elaborative dyads were most likely to avoid the use of negative strategies and potentially engage in positive strategies.

Table 7. Average frequency of explicit mention of positive and negative strategies

	Condition			
	Control Dyads		*Elaborative Dyads*	
1. Identify high/low stress	0.12	(0.50)	0.25	(0.46)
2. V.O.T.A.T.	0.31	(0.60)	0.25	(0.46)
3. Identify failure points	0.37	(0.72)	0.75	(1.75)
4. Conduct experiments	0.44	(0.73)	1.00	(2.07)
Positive strategies	**1.25**	**(1.69)**	**2.25**	**(4.37)**
5. Sub. members & joints	5.87	(5.12)	2.62	(3.50)
6. Add members & joints	1.75	(1.84)	1.87	(1.64)
7. Use longer members	0.69	(1.35)	0.75	(1.75)
Negative strategies	**8.31**	**(5.61)**	**5.25**	**(5.73)**

Note. Standard deviations are in parentheses.

Discussion

As stated in the introduction, elaborative activities enhance learning by increasing an individual's ability to recall information, comprehend a text, and learn conceptual material. An analogous finding was observed in the present experiment because the elaborative condition produced better designs and answered more deep-level questions correctly than the control condition. How did instructions to elaborate lead to better problem-solving performance and, therefore, learning?

An analysis of the interactions may offer a potential explanation. Specifically, elaboration may facilitate the quick assignment of variables to unfilled slots. The results suggested that elaboration may have been effective in filling unassigned variables because the elaborative dyads asked fewer clarification questions than the control condition. Additionally, the results indicate that the elaborative dyads were better able to use the feedback from the simulation in deciding *where* to implement the changes.

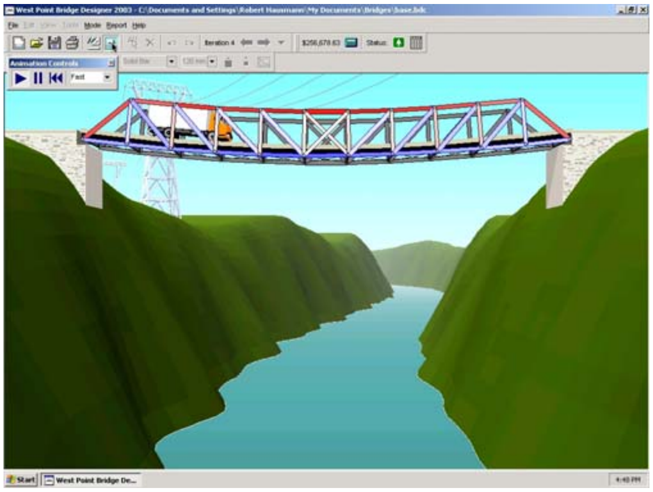
Combining these two results, making faster variable assignments and better use of the simulation's feedback, an exchange between two dyads in the elaboration condition might be interpreted in the following way. One person suggests that they modify the member properties by switching solid members to hollow. The second person may elaborate the suggestion by looking at the feedback and making a recommendation. If the second person makes explicit how she made her recommendation, then the use of the feedback is now available to the dyad for future use. The finding that the elaborative dyads were more likely to be classified as using positive strategies supports this interpretation.

In conclusion, the results suggest that leaving undergraduates to their default interaction style may be usefully enhanced if they are given short instructions on how to elaborate upon their partner's ideas and suggestions.

Acknowledgements

Funding for this research is provided by the National Science Foundation, Grant Number SBE-0354420 to the Pittsburgh Science of Learning Center (PSLC, http://www.learnlab.org). The author is indebted to Leslie R.M. Hausmann, Michelene T.H. Chi, Christian D. Schunn, Kevin Crowley, and Jonathan W. Schooler for their critical comments on previous drafts.

Appendix



References

Anderson, J. R., & Reder, L. M. (1979). An elaborative processing explanation of depth of processing. In L. S. Cermak & F. I. Craik (Eds.), *Levels of processing in human memory*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Brown, A. L., & Palincsar, A. S. (1989). Guided, cooperative learning and individual knowledge acquisition. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 395-451). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

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.

Clark, H. H., & Brennen, S. E. (1991). Grounding in communication. In L. B. Resnick, J. M. Levine & S. D. Teasley (Eds.), *Perspectives on socially shared cognition* (pp. 127-119). Washington, DC: American Psychological Association.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (2 ed.). Hillsdale, NJ: Lawrence Earlbaum Associates, Inc.

Dillenbourg, P., Baker, M., Blaye, A., & O'Malley, C. J. (1995). The evolution of research on collaborative learning. In P. Reinman & H. Spada (Eds.), *Learning in humans and machine: Towards an interdisciplinary learning science* (pp. 189-211). New York: Elsevier Science Inc.

Hausmann, R. G. M. (2005). *Elaborative and critical dialog: Two potentially effective problem-solving and learning interactions.* Unpublished doctoral dissertation, University of Pittsburgh, Pittsburgh.

Hausmann, R. G. M., & Chi, M. T. H. (2002). Can a computer interface support self-explaining? *Cognitive Technology, 7*(1), 4-14.

Hogan, K., Nastasi, B. K., & Pressley, M. (1999). Discourse patterns and collaborative scientific reasoning in peer and teacher-guided discussions. *Cognition and Instruction, 17*(4), 379-432.

McGregor, M., & Chi, M. T. H. (2002). Collaborative interactions: The process of joint production and individual reuse of novel ideas. In W. D. Gray & C. D. Schunn (Eds.), *24nd Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Lawerence Erlbaum.

Reder, L. M., Charney, D. H., & Morgan, K. I. (1986). The role of elaborations in learning a skill from an instructional text. *Memory and Cognition, 14*(1), 64-78.

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*(1), 90-108.

Ressler, S. J. (2002). *Designing and building file-folder bridges: A problem-based Introduction to engineering*. Defense Dept., Army,: United States Military Academy.

Stark, R., Mandl, H., Gruber, H., & Renkl, A. (2002). Conditions and effects of example elaboration. *Learning and Instruction, 12*(1), 39-60.

Stein, B. S., & Bransford, J. D. (1979). Constraints on effective elaboration: Effects of precision and subject generation. *Journal of Verbal Learning and Verbal Behavior, 18*, 769-777.

van Boxtel, C., van der Linden, J., & Kanselaar, G. (2000). Collaborative learning tasks and the elaboration of conceptual knowledge. *Learning and Instruction, 10*, 311-330.