Expressing and Addressing Uncertainty: A Study of Collaborative Problem-Solving Dialogues

Fernando J. Rodríguez, Kimberly Michelle Price, and Kristy Elizabeth Boyer
University of Florida, Gainesville, Florida, USA
{fjrodriguez, kimberlymprice, keboyer}@ufl.edu

Abstract: To support learners during collaborative problem solving, developing a deeper understanding of collaborative dialogue is essential. This paper focuses on one important aspect of collaborative dialogue: expressions of uncertainty. In a study of undergraduate novice computer science students working in pairs, we observed that the students who produced the lowest quality solutions expressed uncertainty more often than those who produced middle-quality solutions. Perhaps surprisingly, pairs with the highest quality solutions also expressed more uncertainty than the middle performers. Examining the ways in which students expressed and then followed up on uncertainty revealed that higher-performing pairs utilized emerging learning opportunities when uncertainty was expressed, and remained focused on one task at a time. In contrast, the lower-performing pairs often did not resolve their uncertainty before moving on, attempting to work with multiple incomplete pieces of the solution simultaneously. These findings provide insight into how best to support collaborative learning during uncertainty.

Introduction

Collaborative dialogue is a complex process through which learners express their perspectives and catalyze learning (Gee, 2014; Howley, Mayfield, & Rosé, 2011; Rosé et al., 2008; Vygotsky, 1978). Through dialogue, *uncertainty* often arises as students self-explain (Chi, De Leeuw, Chiu, & LaVancher, 1994), implicitly inviting their collaborators to elaborate (Webb, 1982). Learners also express uncertainty as a form of politeness or hedging, allowing a less knowledgeable collaborator or oneself to avoid embarrassment (Brown & Levinson, 1978; Markkanen & Schröder, 1997). Uncertainty during collaboration can provide opportunities for learning by inciting curiosity and exploration (Berlyne, 1978). However, if collaborators repeatedly (or for prolonged periods of time) do not address their own uncertainty or that expressed by others, frustration and missed learning opportunities can ensue (D'Mello & Graesser, 2012).

There is evidence that adapting to students' uncertainty as expressed through dialogue can have significant benefit. In a study of undergraduate students learning physics through spoken dialogue with an intelligent tutoring system, the students learned significantly better when the system adapted to the presence of uncertainty (Litman & Forbes-Riley, 2014). Unlike intelligent tutoring systems, humans naturally adapt to each other's uncertainty. For collaborative problem solving in particular, in which students collaborate to produce a shared solution (Nelson, 1998), our recent work has shown that the frequency of several types of dialogue utterances, including expressions of uncertainty, are associated with quality of the shared solution (Rodríguez, Price, & Boyer, 2017). This paper takes a deeper look at expressions of uncertainty and how collaborative pairs address them during the problem solving process.

This paper examines collaborative problem solving in the domain of computer science. Specifically, the learners in this study solve programming problems in pairs within a structured collaborative paradigm known as *pair programming* (Nagappan et al., 2003). There are two collaborator roles in pair programming: the *driver* writes the program, while the *navigator* provides feedback and instructions. Together, driver and navigator produce a single shared solution (Falkner, Falkner, & Vivian, 2013; Porter & Simon, 2013). We collected dialogue and problem-solving data from pairs of students who interacted remotely through textual dialogue. We found that, perhaps not surprisingly, pairs who produced the lowest quality solutions showed more expressions of uncertainty (both in terms of absolute frequency and relative frequency) than pairs with middle-quality solutions. However, pairs who produced high-quality solutions also made significantly more uncertainty expressions than the middle performing pairs. The results show that higher-task-quality pairs were in a position to take advantage of the learning opportunities that uncertainty affords: they often addressed uncertainty by experimenting in their programming code until they resolved the uncertainty, then moved on to the next subtask. In contrast, the lower performing pairs often did not focus their efforts in the same way, leaving uncertainty unresolved and moving on to the next subtask. By understanding these processes, we hope to inform the design of adaptive systems that support student pairs during collaborative problem solving.

Related work

Prior work has considered numerous types of dialogue moves that express uncertainty. Some uncertainty utterances express confusion (Keltner & Shiota, 2003), *e.g.*, "Why did that happen?" or "I don't understand." Closely related to confusion is the notion of *cognitive dissonance* (Festinger, 1962) or *cognitive disequilibrium* (Graesser, Lu, Olde, Cooper-Pye, & Whitten, 2005), when the observed state of the world does not match what a learner expected. These cognitive states are different from the phenomena underlying a type of politeness-uncertainty (Brown & Levinson, 1978), in which a conversational partner attempts to avoid "face-threatening" moves that could lead to embarrassment on the part of the other partner. Similarly, people often *hedge* their dialogue moves, making those utterances more fuzzy and less certain, in order to save themselves from embarrassment (Markkanen & Schröder, 1997), *e.g.*, "I think…", a phrasing seen regularly in dialogue during learning (Forbes-Riley & Litman, 2011).

Berlyne (1978) emphasizes the importance of uncertainty because it leads to curiosity and exploration, with substantial potential for students to learn as the uncertainty is resolved. More specifically, if uncertainty related to confusion is not successfully addressed, it can lead to negative outcomes such as interrupted flow, frustration, and boredom (D'Mello & Graesser, 2012). For example, Litman & Forbes-Riley (2014) evaluated adaptive support to student uncertainty within an ITS for physics problems. The intelligent tutoring system provided different levels of adaptation based on student uncertainty and correctness. The system that provided adaptive support based on presence of uncertainty did improve students' learning gains. The tutor that specifically adapted to the different levels of uncertainty and correctness, however, did not provide further benefit. Our goal is to investigate uncertainty during collaborative problem solving, paying close attention to how students expressed and addressed uncertainty differently, and how those differences relate to the quality of the solution that the pair constructed.

A study by Sharma et al. (2013) investigated pair programming from the perspective of pair program comprehension. In that study, students collaboratively evaluated Java programs and researchers analyzed students' dialogue and gaze. They found that successful pairs tended to focus together on the same program elements and their dialogue was centered around describing the program, as opposed to less successful pairs whose dialogue focused on managing the collaboration. The study presented in this paper provides new findings along these lines: we investigate how expressions of uncertainty are associated with the quality of the shared solution that pairs produce.

Study description

Study participants were recruited from an introductory programming course in Java for computer science majors at a university in the southeastern United States. Out of the approximately 450 students enrolled in the course, 54 voluntarily participated in the programming study. The volunteers were 40 male and 14 female students; 25 White, 16 Latino, 11 Asian, 1 Black, and 1 Pacific Islander; with ages between 18 and 31 (M=19.6, SD=2.21). The students were assigned to pairs based on their mutual scheduling availability, for a total of 27 pairs.

Students were asked to create a program that acts as a math tutor to help young children practice addition, subtraction, and multiplication. They used the *Snap!* block-based programming language to implement this program (Figure 1). In *Snap!*, programmers create programs by dragging blocks and snapping them together to create the necessary logical structure. Block-based programming languages are increasingly common for introductory computer science both in K-12 and at the postsecondary level for programming novices. We chose this programming language because one of the broader goals of our work is to understand the affordances of block-based versus traditional textual programming languages for fostering collaboration in computer science problem solving. Although students were partway through a course in the Java programming language, the block-based programming task presented substantial challenge to them because they were addressing a new problem for which they needed to utilize an appropriate algorithm and reuse previously constructed code modules. Students had one hour to work on the activity, including implementation and software testing, with no requirement that they complete the full activity before the time ended. Only one pair completed all implementation and testing activities within one hour. The students' math tutor program needed to display an equation with the operator blanked out and prompt the user to select an operator that solves the equation. If no operator satisfied the equation, the user would select "None". The learning task was to implement the code for the tutor to 1) display the equation, 2) evaluate the user's choice, and 3) let her know if her answer was correct or incorrect. The modules for selecting the operators and "None" were implemented ahead of time and provided as scaffolding so that the students could focus on implementing the remaining functionality of the program.

When they arrived for the study, students were seated in separate rooms. They collaborated through an interface (Figure 1) that provided a synchronized view of the problem-solving area and textual dialogue through Google Hangouts. This collaboration modality is common in students' everyday practice: they often share screens remotely and interact via text messages or instant messaging while solving problems together. Researchers randomly assigned one student to the *driver* role and the other student to the *navigator* role. Due to technical limitations of the screensharing interface, the collaborative roles remained fixed throughout the one-hour session. The driver actively engaged in programming actions, while the navigator viewed the instructions and communicated with the driver.

Students completed an assessment activity before and after the collaborative programming task. They worked individually on the pre- and post-assessments in which they were given three minutes to implement a short program to display the larger of two randomly-generated integers. We used a 10-point rubric to assign each student an assessment score. The average pre-assessment score was 4.6 (*SD*=1.9, max=10, min=1), and the average post-assessment score was 7.9 (*SD*=1.7, max=10, min=4). The average learning gain of 3.3/10 (post-assessment minus pre-assessment) is significantly nonzero (p<0.0001; paired t-test). The collaborative role is significantly associated with learning gain: drivers' average learning gain was 4.2/10, while for navigators the average was 2.3/10. This difference is statistically significant (p=0.0008; two-sample t-test).

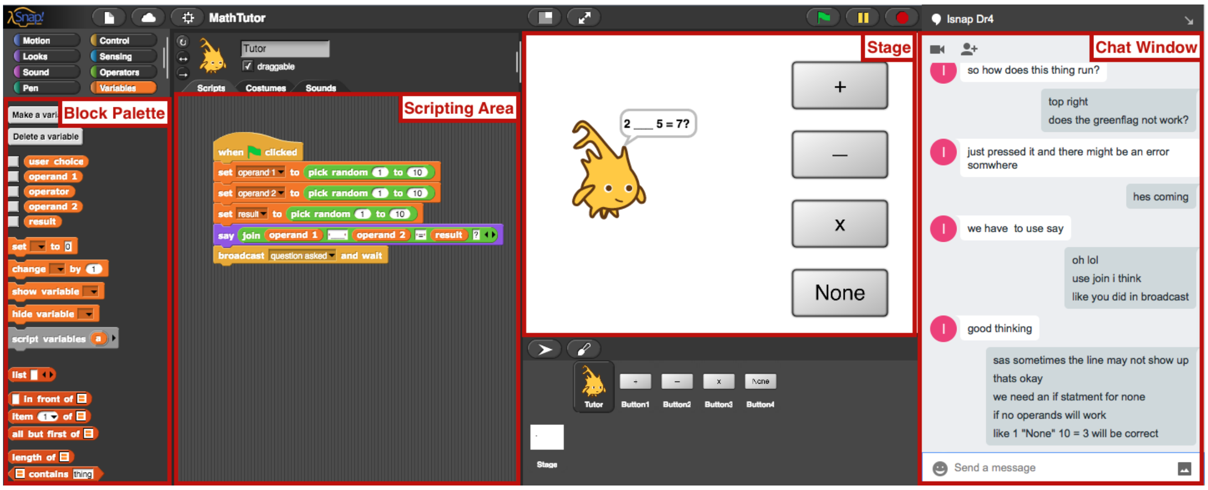

Figure 1. Problem-solving and dialogue interface.

The version of *Snap!* used in this study was instrumented with database logging of programming actions. Whenever a student performed an action in the interface (adding or removing a programming block, connecting or separating two blocks, moving blocks within the interface, editing the parameters of a block, switching between block categories, or running the program), an entry for the action was added to a database. Each event entry included the timestamp, action type, and the current state of the program. The dialogue history was extracted from Google Chat and combined with the action logs based on the relative timestamp, yielding a dataset of programming action and chat sequences. There were a total of 9935 programming actions (*M*=345 per session, *SD*=112.3, max=654, min=111) and 3438 chat messages (*M*=127 per session, *SD*=61.8, max=233, min=47). Drivers sent a total of 1089 messages (*M*=40 per session, *SD*=18.7, max=73, min=10) while navigators sent a total of 2349 messages (*M*=87 per session, *SD*=53.4, max=200, min=28).

Data Analysis

We were interested in examining how dialogue unfolded within each pair and comparing these dialogues based on the pairs' performance on the given task. To evaluate the quality of the pairs' solutions, we collected the final versions and graded them with a purpose-built 11-point rubric, which accounted for presence of all necessary code blocks and test results for functionality of the program. The average solution score was 7.1 (*SD*=2.1, max=11, min=3). For further analysis, student pairs were split into three groups based on their solution quality: pairs with a score of 9, 10, or 11 were classified as *High* (*N*=8); pairs with a score of 6, 7, or 8 were classified as *Medium* (*N*=11); and the remaining pairs, who scored 3, 4, or 5, were classified as *Low* (*N*=8). Given these sample sizes, we used the nonparametric Wilcoxon rank-sum test to evaluate significant differences between the groups.

We extracted the textual dialogue messages and split them into utterances based on punctuation marks (periods, question marks, exclamation marks). Uncertainty utterances were tagged as part of a broader dialogue labeling study that included thirteen distinct tags such as direct and indirect instructions, questions and answers, and partner feedback. Table 1 describes the dialogue act tagging scheme applied to the chat messages. More details about the tagging scheme can be found in our previous work (Rodríguez, Price, & Boyer, 2017). The dialogue act labeling was reliable, with Cohen's Kappa of 0.73. For the work presented in this paper, we take a closer look at messages tagged as expressions of uncertainty (*total*=133, *M*=4.9 per session, *SD*=4.0, max=14, min=1), which include explicit statements of confusion (*e.g.*, "Huh?") as well as hedged suggestions (*e.g.*, "Maybe we should…") and uncertain explanations (*e.g.,* "I *think* it's because…"). With regards to driver uncertainty specifically, we found that these events were more frequent in both high-performing (*total*=16, *M*=2 per session, *SD*=0.76, max=3, min=1) and low-performing pairs (*total*=22, *M*=2.75 per session, *SD*=1.49, max=5, min=1) when compared to medium-performing pairs (*total*=10, *M*=0.91 per session, *SD*=0.94, max=3, min=0) with Wilcoxon rank-sum test *p-values* of 0.0180 and 0.0082, respectively.

Table 1: Dialogue act tagging scheme. This paper focuses on the Uncertainty tags

Tag	Name	Description	Example
S	Statement	Statement of information or an explanation	*We need to create a program for kids to learn math*
U	Uncertainty	Statement of uncertainty, suggestion, or indication of confusion	*unsure how to add strings together*
D	Directive	Explicit instruction to the partner (includes references to specific interface elements)	*wait put the if back*
SU	Suggestion	Polite or indirect instruction to the partner	*maybe we can do if user choice = +*
ACK	Acknowledgement	Acknowledging a partner's previous message	*oh ok gotcha*
M	Meta-comment	Reflection on the problem-solving process (what the student is thinking or doing)	*hmmm*
QYN	Yes/No Question	Task-related question requesting a yes/no response	*can the answer be negative?*
QWH	Wh- Question	Task-related question requesting information (who, what, where, when, why, and how)	*how do I take in their input?*
AYN	Yes/No Answer	Response to a task-related yes/no question	*yea*
AWH	Wh- Answer	Response to a task-related information question	*The program should be able to generate erroneous questions*
FP	Positive Feedback	Response to the partner's actions that is distinctly positive	*oh nice*
FNON	Nonpositive Feedback	Response to the partner's actions that is not distinctly positive	*thats weird*
O	Off-task	Unrelated to the task	*wow its sweet in this room*

Examining uncertainty in collaborative dialogue

When examining driver uncertainty, our hypothesis was that high- and low-performing pairs dealt with moments of uncertainty differently; in particular, we hypothesized that high-performing pairs addressed and resolved uncertainty while low-performing pairs were unable to or took longer to do so. In our dataset, driver uncertainty manifested itself one of two primary ways: suggestions/hedges, and explicit confusion. In this section, we describe examples of each kind of uncertainty and compare how it was managed by high- and low-performing pairs.

Students' incoming knowledge level is likely an important influencing factor in how students expressed, and addressed, uncertainty during collaboration. Indeed, students in the *High* group for task solution quality were more knowledgeable at the outset: they scored significantly higher on the pre-assessment than students in the *Low* solution quality group. The average pre-assessment score for students in the *High* collaborative solution quality group was 5.2/10, while the average pre-score for students who generated a *Low* quality collaborative solution was 3.8/10 (p=0.0078; Wilcoxon rank-sum test). This higher knowledge at the outset likely enabled collaborators to more effectively identify their own confusion or uncertainty, and helped them more successfully address it. The following section examines driver uncertainty events and how they appear to relate to the group's success in the collaborative problem-solving activity. Each excerpt presented in the following subsections contains the original student messages, some of which contain typos. The gender of each student is indicated at the start of the excerpt (*M* for male, *F* for female). Driver uncertainty messages appear in bold text for emphasis.

Case 1: Expressions of uncertainty as suggestions

Students often communicated with their partners in a hedged manner that can indicate politeness or face-saving, but which manifests as uncertainty. For example, when the pair identified an error, the driver often suggested a reason for the error or proposed a solution. These kinds of utterances typically began with "I think" or "I could."

Excerpt 1 parts *a* and *b* show examples of uncertainty events for a high-performing pair and a low-performing pair. For the high-performing pair, the driver and navigator were having trouble completing a subtask. The navigator proposed a solution while the driver hypothesized about the reason for the program error.

The driver then made changes to the program, tested it, and proved his intuition was right. In the low-performing pair, students attempted a different task. Both the driver and the navigator were unsure of how to solve the task. The driver suggested an approach, and the navigator approved by giving positive feedback but then expressed uncertainty by following his feedback with "I think." The pair did not return to discuss the driver's uncertainty further. It was left unresolved as the pair moved on to another subtask.

Excerpt 1: Suggestion dialogue excerpts.

a) High-Performing Pair (driver: M, nav.: F)	**b) Low-Performing Pair** (driver: M, nav.: M)
Navigator (15:31:12): hmmm **Navigator (15:31:33):** maybe they all need to be in one say block **Driver (15:41:34): I think it just says result** *(Driver edits program parameters and tests it)* **Driver (15:42:06):** yup, it skips the first two "say"s	*(Pair discuss how to assign values to equation parameters for 2 minutes)* **Driver (11:49:31): I think we can with "se"t** *(Driver experiments with and figures out how to set the value of a variable)* **Navigator (11:51:21):** yes its gonna be something like what you're doing **Navigator (11:51:31):** i think

This kind of uncertainty is related to the concept of *subjective uncertainty* in that it leads to specific exploration, behavior prompted by events of uncertainty that focuses on eliminating it (Berlyne, 1978). Both of the instances of uncertainty shown in the excerpts represent an opportunity for learning. In the high-performing pair's session (excerpt 1a), both the driver and navigator were attempting to solve the same problem, providing suggestions to each other for consideration. The driver then tested and confirmed his thinking, resolving the uncertainty. He even explicitly stated his findings in the dialogue, letting the navigator know that they could move on to the next subtask. They were able to take advantage of the learning opportunity by exploring potential solutions and resolving the uncertainty. The low-performing pair found themselves in a similar situation, but with a different outcome. Both members were trying to solve the same problem, and the driver was able to figure out the solution on his own. However, the driver did not explicitly let the navigator know that he had found a solution. Instead, the navigator acknowledged the driver's attempt at a solution in the dialogue but expressed his own uncertainty about the approach. They did not leverage the opportunity in part because the driver did not explicitly state that he had arrived at a solution, and the navigator appears to have missed the opportunity to add an understanding of the approach to the pair's common ground.

Case 2: Expressions of uncertainty as confusion

Many uncertainty events are utterances indicating that the speaker is confused or does not understand something. These utterances often begin with "I don't know" or "I'm not sure".

Excerpt 2 parts *a* and *b* show two examples of confusion-related uncertainty, one from a high-performing pair and one from a low-performing pair. In Excerpt 2a, the driver in a high-performing pair expressed to the navigator that he did not know how to implement a component of the task. The navigator pointed out to the driver where he could find the programming blocks he needed, and the driver added the block to the program. In the low-performing pair (Excerpt 2b), the driver also told the navigator that she did not know how to implement a component of the task. The navigator provided quick guidance, but then the conversation shifted to another part of the task for a few minutes. Afterward, the navigator revisited the previous task component with the driver, and they both mentioned that they did not know how to complete it. At this point, the conversation shifted once more to a separate task component for a longer period of time. After testing the program, the driver brought back the unresolved issue, but the session ended without resolving this issue.

Research on cognitive load theory suggests that having too many simultaneous workflows puts a strain on working memory, limiting the amount of information that can be processed (Renkl & Atkinson, 2010). In the high-performing pair session, the driver expressed uncertainty and the pair worked together to resolve it before moving on to the next steps. Since they focused on a single task at a time, they were able to utilize the full potential of their working memory to quickly overcome the uncertainty. Conversely, the low-performing pair switched between several subtasks, leaving any established uncertainty unresolved. By attempting to complete multiple subtasks simultaneously, the low-performing pair may have taxed their working memory and inhibited their ability to address the uncertainty.

Excerpt 2: Confusion dialogue excerpts.

a) High-Performing Pair (driver: M, nav.: F)	**b) Low-Performing Pair** (driver: F, nav.: M)
Driver (13:55:43): unsure how to add strings together **Navigator (13:55:47):** can you use an operator? **Navigator (13:55:31):** go to the ops *(Driver switches to the Operator block category)* **Driver (13:56:07):** oh shoot nice *(Driver adds "join" block to the program)* **Driver (13:56:21):** this one?	**Driver (14:11:23): im not sure how to display the operands** **Navigator (14:12:17):** try locating the "say" button *(Pair discuss a different task)* **Navigator (14:21:13):** Is there a way we can use the "say" button and then put the operators in it **Navigator (14:21:26):** then use user choice button to store the users result **Driver (14:22:30):** Yeah I havent figured out yet how to display more than one variable at a time *(Pair discuss a different task)* **Driver (14:45:25): im not sure how to make it display the operands and the result together**

Case 3: Addressing uncertainty around a similar task

In our third example (Excerpt 3 *a* and *b*) the high-performing and low-performing pairs expressed uncertainty toward the same task: selecting a random operator on which to base their equation. Both excerpts occur near the end of the one-hour collaborative session. In the high-performing pair, the driver expressed uncertainty at how to complete the task, and then the navigator asked a clarification question. This prompted the driver to explain his point of view on the task at hand. The navigator understood the question and turned to the task instructions to find an answer, warning the driver of the time remaining a few minutes later. During this time, the driver edited the program for a few minutes and found the solution ten minutes before the end of the session. For the low-performing pair, the driver similarly expressed uncertainty regarding the given task. The navigator provided feedback, but the driver did not explicitly acknowledge this feedback. The driver experimented with the program for a few minutes, and the navigator asked a question about a separate task. The driver answered the navigator's question and proceeded to implement his program, running out of time for the session in the process.

Excerpt 3: "Selecting a random operator" excerpts

a) High-Performing Pair (driver: M, nav.: F)	**b) Low-Performing Pair** (driver: F, nav.: M)
Driver (16:11:12): i dont know how to let it pick a random one to show now **Navigator (16:12:10):** You mean a random operation symbol? **Driver (16:12:21):** no, **Driver (16:12:26):** like you see how i have 4 blocks? **Driver (16:12:32):** one for each option? **Driver (16:12:48):** you know how to let the program randomly picks one? **Navigator (16:12:52):** oh i see… **Navigator (16:13:08):** i don't know let me check the instructions again *(Driver experiments with program)* **Navigator (16:15:11):** *<copy/pasted instructions and answer>* *(Driver experiments with program)* **Navigator (16:18:34):** btw the guy said we have 10 min left *(Driver experiments with program)* **Driver (16:23:38):** DONE	**Driver (14:51:00): I still dont understand how to choose a random operator though** **Navigator (14:51:02):** yeah, we just need to get the question to display and then say correct or incorrect when given the use input *(Driver experiments with program)* **Navigator (14:54:47):** Is there like one big "say" button to display the equation all at once? **Driver (14:55:27):** Hmm.. **Driver (14:55:27):** not that I see *(Driver experiments with program)* *(Session runs out of time)* **Navigator (14:59:11):** Nice job we were close!

Excerpts 1 and 2 provided evidence that student collaboration in high-performing pairs encouraged specific exploration and properly managed the pair's cognitive load. Excerpt 3 is consistent with the previous two. In Excerpt 3 part *a*, the driver made sure that the navigator understood his source of uncertainty, and both partners were able to engage in specific exploration with respect to their given roles: the driver experimented with the program, and the navigator reviewed the task instructions. Additionally, by only focusing on one task, the pair had enough processing power in their working memory to address the current task. In contrast, Excerpt 3 part *b* suggests that the students were unable to surpass their confusion, and they may have suffered the effects in terms of cognitive load from not addressing one source of uncertainty before moving on to the next task. In the excerpt, the driver expressed uncertainty, the navigator gave feedback, but the driver did not acknowledge this feedback and continued to work on the task. The navigator also did not provide more feedback on the driver's actions. In contrast, the driver from the high-performing pair let the navigator know that he was able to solve the task and that they could move on; by not explicitly stating her process, the driver from the low-performing pair may have left them unable to address their uncertainty.

Recommendations and limitations

The results described above suggest some recommendations for supporting students during uncertainty in collaborative problem solving. In Excerpt 1a we saw that the driver from the high-performing pair notified the navigator that he had solved the current subtask, while in Excerpt 1b the driver from the low-performing pair did not mention this in his dialogue. Adaptive scaffolding such as that provided by real-time intelligent learning environments could detect the expression of uncertainty during a subtask and prompt the collaborator to tell his partner when he believes the subtask is solved, addressing both students' uncertainty and providing an opportunity for learning. In Excerpt 2, the high-performing pair focused on one subtask while the low-performing pair switched between several subtasks and did not resolve their uncertainty, possibly due to an increased cognitive load. A real-time collaboration scaffolding system could encourage students to resolve uncertainty in one subtask before moving on to the next. Finally, Excerpt 3 shows how communication between the driver and navigator differed in high- and low-performing groups. The navigator in the high-performing pair asked the driver a follow-up question and maintained open communication regarding what she was doing; the navigator from the low-performing pair steered the focus away from the main task and towards another task. A potential suggestion to assist this pair would have been to encourage the navigator to converse with the driver more by asking her about her thought process and providing feedback. Through this interaction, the partners can achieve a common understanding of each other's process and have a clearer picture of what remains to be completed.

The results discussed in this paper must be interpreted in light of its limitations. First, the sample of student participants was based on volunteers who received a small amount of course credit in exchange for completing an alternate assignment. The nature of this homework credit may have introduced bias in the sample. Another limitation involves the implementation of the pair programming roles. Usually, students within a pair alternate between the driver and navigator roles. Due to the technical limitations of the screensharing software, students in our study were not able to switch roles during the activity. Finally, whether these results will generalize to other populations of students or other collaborative paradigms remains to be seen.

Conclusion

The relationship between uncertainty and task performance during collaboration is complex. We have observed that a larger number of uncertainty utterances were expressed in dialogues of high-performing pairs and low-performing pairs when compared to medium-performing pairs. These expressions of uncertainty are clearly important, and the ways they are dealt with is different between pairs that do well and those who do not. We found that low-performing pairs missed some opportunities for learning and may have pushed their collective working memory to the limit when attempting to multitask. The results suggest that the ways in which collaborators express and address uncertainty could be highly influential in their success in a learning activity, and highlight the importance of supporting this aspect of collaboration.

There are several important directions for future work. Continuing to investigate practices for resolving uncertainty in collaborative problem solving is an important step toward more effectively supporting learners. If uncertainty during collaborative problem-solving activities can be identified, adaptive scaffolds may be able to promote and support specific exploration, reducing the negative effects of unresolved uncertainty and improving learning. Future work should investigate how this adaptive support can manifest itself effectively within collaborative problem solving. Designing and evaluating different forms of support for collaborative problem solving can lead to the next generation of adaptive scaffolding that holds the potential to significantly improve the learning experience.

References

Berlyne, D. E. (1978). Curiosity and learning. *Motivation and Emotion*, *2*(2), 97–175.

Brown, P., & Levinson, S. C. (1978). Universals in language usage: Politeness phenomena. *E. N. Goody (Ed.), Questions and politeness: strategies in social interaction*, 56–311.

Chi, M. T. H., De Leeuw, N., Chiu, M.-H., & LaVancher, C. (1994). Eliciting Self-Explanations Improves Understanding. *Cognitive Science*, *18*(3), 439–477.

D'Mello, S., & Graesser, A. (2012). Dynamics of affective states during complex learning. *Learning and Instruction*, *22*(2), 145–157.

Falkner, K., Falkner, N. J. G., & Vivian, R. (2013). Collaborative Learning and Anxiety: A phenomenographic study of collaborative learning activities. *Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE)*, 227–232.

Festinger, L. (1962). *A Theory of Cognitive Dissonance*. Stanford University Press.

Forbes-Riley, K., & Litman, D. (2011). Benefits and Challenges of Real-Time Uncertainty Detection and Adaptation in a Spoken Dialogue Computer Tutor. *Speech Communication*, *53*(9–10), 1115–1136.

Gee, J. P. (2014). *An Introduction to Discourse Analysis: Theory and Method*. Routledge.

Graesser, A. C., Lu, S., Olde, B. A., Cooper-Pye, E., & Whitten, S. (2005). Question Asking and Eye Tracking during Cognitive Disequilibrium: Comprehending Illustrated Texts on Devices When the Devices Break Down. *Memory & Cognition*, *33*(7), 1235–1247.

Howley, I., Mayfield, E., & Rosé, C. P. (2011). Missing Something? Authority in Collaborative Learning. *Proceedings of the 9th International Conference on Computer-Supported Collaborative Learning (CSCL)*, 366–373.

Keltner, D., & Shiota, M. N. (2003). New Displays and New Emotions: A Commentary on Rozin and Cohen (2003). *Emotion*, *3*(1), 86–91.

Litman, D., & Forbes-Riley, K. (2014). Evaluating a Spoken Dialogue System that Detects and Adapts to User Affective States. *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 181–185.

Markkanen, R., & Schröder, H. (1997). Hedging: A Challenge for Pragmatics and Discourse Analysis. *Hedging and Discourse: Approaches to the Analysis of a Pragmatic Phenomenon in Academic Texts*, 3–20.

Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Miller, C., Balik, S., & Yang, K. (2003). Improving the CS1 Experience with Pair Programming. *Proceedings of the 34th ACM Technical Symposium on Computer Science Education (SIGCSE)*, 359–362.

Nelson, L. M. (1998). *Collaborative Problem Solving: An Instructional Theory for Learning through Small Group Interaction*. Indiana University.

Porter, L., & Simon, B. (2013). Retaining Nearly One-Third more Majors with a Trio of Instructional Best Practices in CS1. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE)*, 165–170.

Renkl, A., & Atkinson, R. K. (2010). Structuring the Transition From Example Study to Problem Solving in Cognitive Skill Acquisition: A Cognitive Load Perspective. *Educational Psychologist*, *38*(1), 15–22.

Rodríguez, F. J., Price, K. M., & Boyer, K. E. (2017). Exploring the Pair Programming Process: Characteristics of Effective Collaboration. *Proceedings of the 48th ACM Technical Symposium on Computer Science Education (SIGCSE)*, To appear.

Rosé, C. P., Wang, Y.-C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., & Fischer, F. (2008). Analyzing Collaborative Learning Processes Automatically: Exploiting the Advances of Computational Linguistics in Computer-Supported Collaborative Learning. *International Journal of Computer- Supported Collaborative Learning*, *3*, 237–271.

Sharma, K., Jermann, P., Nüssli, M.-A., & Dillenbourg, P. (2013). Understanding Collaborative Program Comprehension: Interlacing Gaze and Dialogues. *Proceedings of the 10th International Conference on Computer-Supported Collaborative Learning (CSCL)*, 430–437.

Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press.

Webb, N. M. (1982). Student Interaction and Learning in Small Groups. *Review of Educational Research*, *52*(3), 421–445.

Acknowledgments

We would like to thank the members of the LearnDialogue Group for their helpful input. This material is based upon work supported by the National Science Foundation under grants CNS-1622438, DUE-1625908, and a Graduate Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.