Visualizing Pair Programming Dialogue: An Exploratory Study of Student Reflections on Summative and Time-Series Visualizations

Mehmet Celepkolu[1], David Austin Fussell, Aisha Chung Galdo, Kristy Elizabeth Boyer

University of Florida, Gainesville, FL 32611, USA
[1]`mckolu@ufl.edu`

Abstract. This paper is submitted as a research paper. During pair programming activities, promoting and supporting effective dialogue is an important step toward achieving learning outcomes. However, many learners do not have the teamwork skills essential for effective collaboration, which can result in conflicts between partners, inequity, frustration, and potential failure. This risk may be especially great for younger learners such as those in middle school. One way to improve teamwork is to raise students' awareness of their own collaborative dialogue. In this study, 18 middle school students pair-programmed to complete a CS/Science activity. After the activity, we mined the dialogue from the pair programming transcripts and created several different interactive visualizations reflecting the dialogue processes. Next, we conducted individual think-aloud interviews to investigate how students perceive these visualizations, elicit their misconceptions, and receive feedback for further improvement. The results show that most students were able to correctly interpret the graphs even though some students had confusions about language use. Moreover, most students gave positive feedback for the time series chart illustrating the number of words spoken over time as well as the dialogue transcript, as it presents a more detailed picture of the dialogue process than summative visualizations. The results highlight potential design improvements on current dialogue visualization tools and point to ways to improve help young learners' collaboration skills.

Keywords: Collaborative Dialogue, Group Awareness, Data Visualization.

1 Introduction

Findings from numerous studies indicate the benefits of collaborative learning such as positively impacting academic achievement [1] and developing higher levels of reasoning and critical thinking [2]. Dialogue is a "gold mine" for generating new ideas, enhancing cognitive understanding [3], provoking reasoning [4] and improving critical thinking [5]. In the computer science (CS) context, pair programming is an effective collaboration approach for improving productivity [6] and satisfaction [7], increasing student retention in CS courses [8] and promoting good programming practices [9]. In pair programming, two students work on the same code with two different roles: The *driver* is responsible for writing the code and implementing the solution, while the *navigator* is responsible for assisting with catching mistakes and providing immediate feedback. On the other hand, not all pair programming interactions are successful [10]. Some studies suggested implicitly guiding student interactions by providing knowledge-related group awareness information, which helps learners regulate their collaborative behaviors and adjust it based on the needs of the team. [11–13]. Several studies investigated the effectiveness of visual analytics systems that mine dialogue and reflect some metadata such as participation back to the group members [14, 15].

However, these feedback systems often create visualizations for the total number of contributions and disregard the temporal dynamics of the dialogue flow. Knight and Littleton [16] define temporality as a key facet of data representation for dialogue. While cumulative approaches can be useful in uncovering some patterns related to the productivity of collaborative learning, losing sequential information of interactions is inevitable [17, 18]. Several studies [19, 20] have suggested using interactive time-series visualizations for revealing temporal patterns in collaborative dialogue processes over time. Dyke et al. [21] suggest that using a sliding window approach can show the changes over time by generating a value of the indicator for a small period interaction time. Any visualizations presented should be tailored to characteristics of the learners [22] and perceived as useful [23, 24] and interpretable by the intended users [11].

These challenges lead to an open research question: How can we best support young learners' collaborative problem-solving skills through dialogue visualization approaches? To investigate this broad question, we developed an initial dialogue visualization prototype that automatically mines students' dialogues from pair programming activities and creates several interactive visualizations showing total turn taking count, total word count, total question count, and the number of words spoken over time alongside the dialogue transcript. We implemented the prototype in a classroom study and conducted think-aloud interviews to investigate students' understanding and expectation of these visualizations.

The main potential contributions of this work are threefold. First, unlike the prevalent literature in this context, this paper suggests using interactive time series-based line charts as an alternative to simple pie charts, which provide deeper information about the dialogue process. Second, previous literature often focused on visualization studies in higher education or informal learning settings; in contrast, this study reflects middle-grades learners' needs and expectations. Finally, to our knowledge, this paper is the first to investigate a visualization system for supporting students' collaborative problem-solving skills in the context of pair programming.

2 Methods

Participants: This study was conducted with 18 seventh grade students (ages 11-13) enrolled in a science class in a middle school in southeast United States during Spring 2019. As part of the classroom activities, students learned the *Snap!* programming language and practiced various CS concepts such as variables, conditionals, loops and object-oriented programming, and created computationally rich science activities based on the lesson topic (e.g., food web, evolution). The class met five times per week and students had been coding with a partner (either a randomly assigned or self-selected partner depending on the lesson) since the beginning of the academic year. Out of the 97 students in the class, 75 consented to data collection and nine groups (18 students) were randomly selected for this study.

Data: The data was collected as part of a four-day study. On the first day, students reviewed the CS concepts that they had learned before. On the second day, students were randomly assigned partners to use *Snap!* in a pair programming activity. The students were video/audio recorded during these activities, which were transcribed manually. Next, these transcriptions were fed into the visualization tool (Bokeh visualization library [25]) to generate interactive charts (Fig. 1 and Fig. 2). On the third day, these interactive visuals were given to the students to reflect on during a think-aloud session. On the fourth day, pairs completed another evolution activity in *Snap!*.

Think-Aloud Interviews. The main focus of this paper is to investigate young learners' perceptions, preferences and expectations of the group awareness visualizations; therefore, this paper focuses on the outcomes of the think-aloud sessions [26, 27]. We conducted the interview in three phases: First, we showed them the pie charts and asked them about their first impression and understanding of the charts. We also asked them what they were thinking as they were looking/clicking and what the numbers on the charts meant to them. Second, we showed them the interactive line chart following the same procedure. Lastly, we asked them general questions such as whether they would want to use these types of charts in future, which chart is most helpful for them and suggestions for improvements.

Charts: The following information was depicted in a pie chart form: (1) the total number of times they spoke with their partner, (2) total number of words they spoke with their partner, and (3) total number of questions they asked during the activity (Fig. 1).

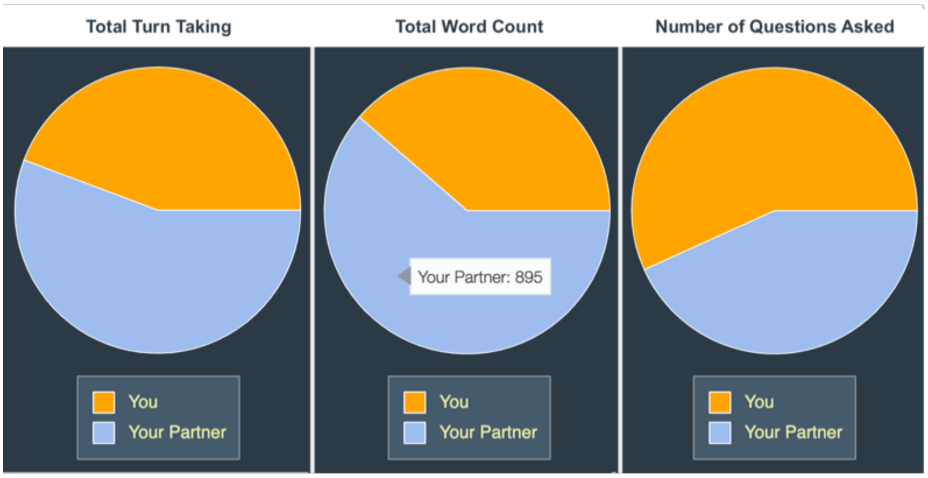


Fig. 1. Pie charts illustrating the total turn taking count, total word count and total question count.

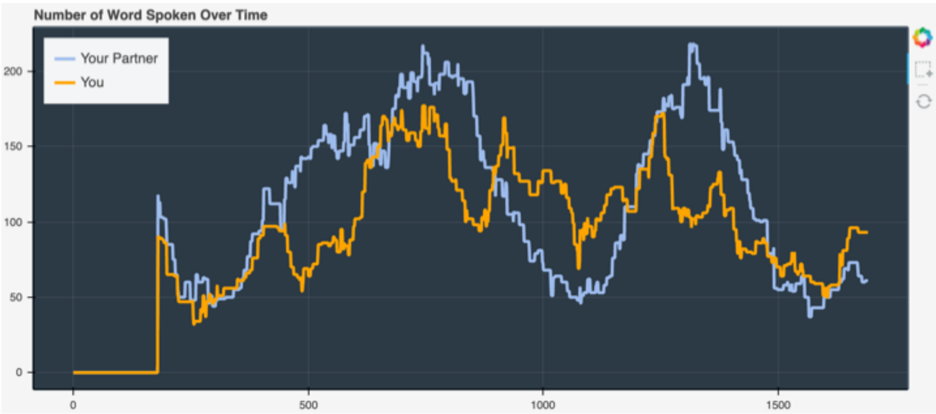
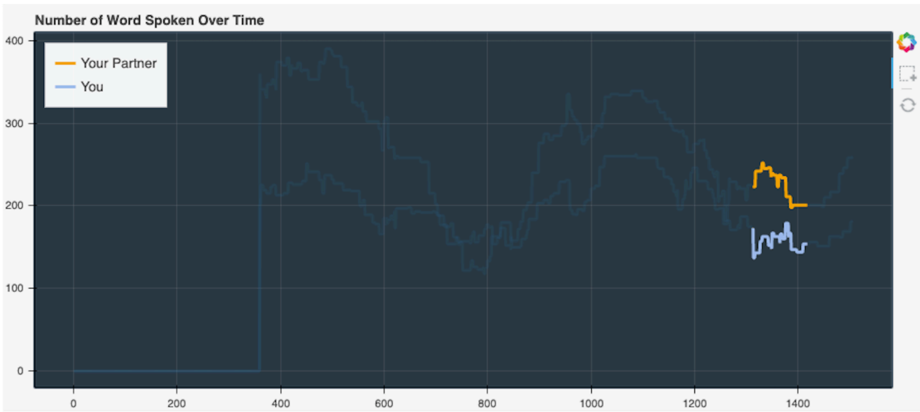


Fig. 2. Interactive line chart illustrating the number of words spoken over time.

Next, a time series-based line chart (Fig. 2) was created based on the sliding window approach [28] showing the number of words spoken over time as follows: (1) Count the total the number of words spoken by a student within a predefined time window interval, (2) Slide the window by one second (drop the earliest and add a new one) and recalculate the total number of words for the new time window, (3) Repeat the same process until the window slides over the entire the dialogue. We also integrated a dialogue exploration functionality, in which students could select certain areas of the chart and examine the dialogue (Fig. 3).



#	Time	Speaker	Sentence
0	21:56	You	And, we need a forever loop.
1	21:57	Your Partner	We need a forever, and then, then wait one second, and then, wait one second, change timer by one?
2	22:06	You	Let's look back at the ... yeah, wait one second, change time by one.
3	22:09	Your Partner	One second, change timer by one. (pause)
4	22:10	Your Partner	Let's see.

Fig. 3. Interactive line chart with a subsection of dialogue selected, along with transcript exploration feature.

3 Results

We first aimed to identify whether the charts were understandable by the students and what misconceptions may be present. Three researchers independently rated students' responses for correctness of interpretation. The rating was based on the students' responses immediately after the charts were introduced to them. Table 1 shows the number of students who interpreted the charts correctly and the interrater reliability scores (a value ranging from -1 (perfect disagreement) to 1 (perfect agreement) showing the consensus between judges) for each chart. Students were able to interpret most charts correctly except the total turn-taking pie chart. Out of 11 students who gave wrong responses, ten students thought the pie chart showed the amount of time used for controlling the computer. Given that students were taking turns depending on their roles as driver or navigator, the confusion about turn taking chart is not surprising. The remaining student mixed the turn taking chart with the word count chart. All students could successfully interpret the question number pie chart and the line charts.

Table 1. Interpretations accuracy and interrater reliability (Kappa) scores

Chart Type	Correct Interpretation (18 students)	Kappa Score
Turn Taking Count	7	.77
Word Count	17	.93
Question Count	18	.85
Line Chart	18	1

Students often made comments reflecting their thoughts and feelings on the activity while engaging with the charts. Several students reported that they talked more because the task was confusing, or they were explaining the concepts to their partner. Some students reported not being aware of talking more/less during the interaction and they would talk more/less to make it balanced. Some students reported that they asked a lot of questions because they tried to understand how to do things. Perhaps unsurprisingly, students talked about the line chart much more than the pie charts. We investigated how students used the line chart and the areas that they selected on the line chart, which can particularly be important for designing information presentation in future.

As Table 2 indicates, the majority of students selected the interval parts that they were either talking more or less than their partner. One student says, "*Well, that, I obviously, could've talked more, because the graphs were so different and he obviously, clearly, talked way more and I need to ask more questions, and maybe contribute more.*" Another interesting point is that while some students highlighted the interval parts related to their own dialogue, some students focused on the times in which the group talked a lot/equal/a little compared to other parts of the chart. While reading their dialogue between their partner, some students provided explanations that they talked less when they were the driver because they were busy with implementing the solution. After examining the dialogue, one student said, "*I think we got off topic a little bit, but we did help each other out,*" and another student mentioned similar ideas, "*Sometimes we get off topic, but ... then once we do find out what the next step is, we kind of go off of that.*"

Table 2. Number of students (out of 18) selecting each type of area on the line graph

Behavior	Student talked more	Student talked less	Group talked a lot	Group talked equal	Group talked a little	Starting	Middle	Ending
Number of Students	10	6	5	1	4	3	1	5

Out of 18 students, only two students said they would not be interested in using these visualizations in the future. Out of 16 students who gave positive responses for using the charts in future, nine students preferred the line chart, four students preferred the pie chart and two students preferred both types of charts. The most common reasons to use the line chart were that it presents more details, shows when they talked the most/least, shows the dialogue content, provides information about the strength and weaknesses, and is useful for record keeping for future use (remembering how they solved the problem). For example, one student said, "*It doesn't just tell you how many words you spoke and things. It tells you exactly when you spoke and how much you spoke at that time, so it's more detailed*.". One student also emphasized the importance seeing the dialogue over time: "*…because you can see like, everything… Like, it's easier to see over time.*". Another student emphasized the importance of seeing the transcript: "*I personally like the line graph more because you can drag and it says exactly what we said, whereas here it just says how many like words we spoke and stuff.".* Some students also mentioned about the potential impact of the line chart on their behavior. One student said, "*I think this is good, 'cause it shows who talked when and what they said; so it helps them, like, -Oh, I need to talk more, I need to talk less, compared to what my partner did.*".

4 Conclusion

The overarching goal for this study was to investigate middle school learners' perceptions, preferences and expectations of dialogue visualizations. The findings from 18 students' think-aloud interviews indicated that interactive time-based series chart is promising for helping students explore the evolving nature of the dialogue during pair programming activities, and may support productive reflections on an individual student's dialogue behaviors as well as the group dynamic. In future work, it is important to develop dialogue visualizations that adapt to students' needs such as understanding their own contributions, on- and off-topic shifts in dialogue, and building a deeper understanding of productive dialogue patterns.

Acknowledgments. This research was supported by the National Science Foundation through grant DRL-1640141. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the authors, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

References

1. McDowell, C., Werner, L., Bullock, H., Fernald, J.: The effects of pair-programming on performance in an introductory programming course. In: Proceedings of the 33rd SIGCSE technical symposium on Computer science education - SIGCSE '02. p. 38. ACM Press, New York, New York, USA (2002)
2. Gokhale, A.A.: Collaborative Learning Enhances Critical Thinking. J. Technol. Educ. 7, 1995–2017 (1995)
3. Kuhn, D., Crowell, A.: Dialogic argumentation as a vehicle for developing young adolescents' thinking. Psychol. Sci. 22, 545–552 (2011).
4. Mercer, N.: Talk and the Development of Reasoning and Understanding. Hum. Dev. 51, 90–100 (2008).
5. Kuhn, D.: A Role for Reasoning in a Dialogic Approach to Critical Thinking. Topoi. 37, 121–128 (2018).
6. Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C., Balik, S.: Improving the CS1 experience with pair programming. ACM SIGCSE Bull. 35, 359 (2003).
7. Celepkolu, M., Boyer, K.E.: The Importance of Producing Shared Code Through Pair Programming. In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education - SIGCSE '18. pp. 765–770. ACM Press, New York, New York, USA (2018)
8. McDowell, C., Werner, L., Bullock, H.E., Fernald, J.: Pair programming improves student retention, confidence, and program quality. Commun. ACM. 49, 90–95 (2006).
9. Cockburn, A., Williams, L.: The Costs and Benefits of Pair Programming. Extrem. Program. Flex. Process. Softw. Eng. XP2000. 223--247 (2000)
10. Celepkolu, M., Boyer, K.E.: Thematic Analysis of Students' Reflections on Pair Programming in CS1. In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education - SIGCSE '18. pp. 771–776. ACM Press (2018)
11. Bodemer, D., Dehler, J.: Group awareness in CSCL environments. Comput. Human Behav. 27, 1043–1045 (2011).
12. Fransen, J., Kirschner, P.A., Erkens, G.: Mediating team effectiveness in the context of collaborative learning: The importance of team and task awareness. Comput. Human Behav. 27, 1103–1113 (2011).
13. Soller, A., Martinez, A., Jermann, P., Muehlenbrock, M.: From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning. Int. J. Artif. Intell. Educ. 15, 261–290 (2005)
14. DiMicco, J.M., Pandolfo, A., Bender, W.: Influencing group participation with a shared display. In: Proceedings of the 2004 ACM conference on Computer supported cooperative work - CSCW '04. p. 614. ACM Press, New York, New York, USA (2004)
15. Charleer, S., Klerkx, J., Duval, E., Laet, T. De, Verbert, K.: Towards balanced discussions in the classroom using ambient information visualisations. Int. J. Technol. Enhanc. Learn. 9, 227 (2017).
16. Knight, S., Littleton, K.: Dialogue as Data in Learning Analytics for Productive Educational Dialogue. J. Learn. Anal. 2, 111–143 (2015)

17. Chen, B., Resendes, M.: Uncovering what matters: analyzing transitional relations among contribution types in knowledge-building discourse. In: Proceedins of the Fourth International Conference on Learning Analytics And Knowledge - LAK ’14. pp. 226–230. ACM Press, New York, New York, USA (2014)
18. Kapur, M.: Temporality matters: Advancing a method for analyzing problem-solving processes in a computer-supported collaborative environment. Int. J. Comput. Collab. Learn. 6, 39–56 (2011).
19. Muukkonen, H., Hakkarainen, K., Kosonen, K., Heikkilä, A.: Process- and context-sensitive research on academic knowledge practices: Developing CASS-tools and methods. In: Proceedings of the Seventh International Computer Supported Collaborative Learning Conference. pp. 541–543 (2007)
20. Reimann, P.: Time is precious: Variable- and event-centred approaches to process analysis in CSCL research. Int. J. Comput. Collab. Learn. 4, 239–257 (2009).
21. Dyke, G., Kumar, R., Rosé, C.P.: Challenging Assumptions: using sliding window visualizations to reveal time-based irregularities in CSCL processes. In: Proceedings of the 10th international conference of the learning sciences. pp. 363–370 (2012)
22. Schnaubert, L., Bodemer, D.: Providing different types of group awareness information to guide collaborative learning. Int. J. Comput. Collab. Learn. 14, 7–51 (2019).
23. Janssen, J., Erkens, G., Kirschner, P.A.: Group awareness tools: It’s what you do with it that matters. Comput. Human Behav. 27, 1046–1058 (2011).
24. Nova, N., Wehrle, T., Goslin, J., Bourquin, Y., Dillenbourg, P.: Collaboration in a multi-user game: impacts of an awareness tool on mutual modeling. Multimed. Tools Appl. 32, 161–183 (2007).
25. Bokeh 1.1.0 documentation, https://bokeh.pydata.org/en/latest/
26. Ericsson, A., Simon, H.A. (Herbert A.: Protocol analysis : verbal reports as data. MIT Press (1993)
27. Ericsson, K.A., Simon, H.A.: Verbal reports as data. Psychol. Rev. 87, 215–251 (1980).
28. Datar, M., Gionis, A., Indyk, P., Motwani, R.: Maintaining Stream Statistics over Sliding Windows. SIAM J. Comput. 31, 1794–1813 (2002).