Evaluating Student Learning in a Synchronous, Collaborative Programming Environment Through Log-Based Analysis of Projects

Bernard Yett(✉), Nicole Hutchins, Caitlin Snyder, Ningyu Zhang, Shitanshu Mishra, and Gautam Biswas

Department of EECS, Institute for Software Integrated Systems, Vanderbilt University, 1025 16th Avenue South, Nashville, TN 37212, USA
{bernard.h.yett,nicole.m.hutchins,caitlin.r.snyder,ningyu.zhang,
shitanshu.mishra,gautam.biswas}@vanderbilt.edu

Abstract. In this paper we present an initial analysis of synchronous, collaborative programming in a robotics platform. Students worked in dyads and triads to complete a week-long curriculum targeting the learning of cybersecurity and computational thinking concepts, and their application using realistic robotics scenarios. We demonstrate how an analysis of individual student activity data within a group can be extrapolated to understand the group's collaborative problem-solving. We compare our findings to past literature and discuss future implications of collaborative programming research.

Keywords: Collaborative learning · Robotics · Programming action logs · K-12 education · Computational thinking · Cybersecurity

1 Introduction

Collaborative problem-solving is an essential 21st century workforce skill. Collaborative learning and problem solving have proven to be especially useful in the context of programming tasks [6]. Efforts to introduce collaborative programming in K-12 classrooms have led to tools and curricula that support co-located and remote programming tasks. However, limitations exist in the application of these tools in today's classrooms, including the inability to distinguish individual student programming actions in co-located peer-programming environments and the inability of group members to communicate and discuss verbally when they are physically separated [21].

Collaboration represents "a coordinated, synchronous activity that is a result of a continuous attempt to construct and maintain a shared conception of a problem" [14, p. 70]. Research has examined collaborative discourse for improved understanding of problem-solving [16,17] and regulatory [5,13] processes that collaborative teams implement during a programming task. However, to our

I. I. Bittencourt et al. (Eds.): AIED 2020, LNAI 12164, pp. 352–357, 2020.
https://doi.org/10.1007/978-3-030-52240-7_64

knowledge, limited research has examined individual log actions of co-located students participating in a collaborative programming environment to solve problems. In our research, we examine log data of collaborative groups working in a synchronous, block-based programming environment (BBPE), NetsBlox [3], to answer (1) *What can individual student log data tell us about the group's collaborative programming?* and (2) *How do these programming activities impact student learning?* We first provide a brief background on K-12 collaborative programming. This is followed by our log-based analysis of individual students' programming activities, and their implications on collaborative program generation. We conclude with a discussion and future implications of our research.

2 Background

Collaborative programming is an effective pedagogical approach for the learning computer science concepts and practices [6,11,18]. Research has demonstrated significant benefits (i.e., learning gains) during pair programming (two, co-located students sharing one computer) that targets inclusivity [10,18]. However, peer programming studies in K-12 have not considered designing for equality of control of the task [7] and conversational equity [15].

Recent efforts have led to the development of synchronous, collaborative programming environments [2,3,21]. These environments allow students to be co-located but working on separate machines, thus improving equality of control in the programming task while still allowing face-to-face discussions. Initial analysis of these approaches have mainly targeted discourse analysis (e.g., [21]), including comparing this approach to the more well-known pair programming. Understandings of individual student actions, captured through log data, as part of the collaborative programming task are under-researched.

3 Methods

Thirty-eight high school students participated in our intervention aimed at teaching cybersecurity and computational thinking (CT) concepts using a robotic environment as a teaching tool. Students were evaluated in cybersecurity and CT, and the results were computed as average normalized change (ANC) [12] from pre-test to post-test. An overview of the intervention and the BBPE used are presented in [9]. The computed learning gains were statistically significant in both cybersecurity and CT [20].

To analyze student work, we extracted relevant information from their activity logs and modeled the students' actions as solution construction (SC) or solution assessment (SA) actions. SC actions were subdivided into (1) **SC_ computational** actions that include adding, connecting, disconnecting, or removing a block, and (2) **SC_conceptual** actions that refer to creating, modifying, or deleting a custom block definition. SA actions were subdivided into (1) **SA_global** actions for starting a simulation simultaneously for all Sprites, (2) **SA_local** actions for starting a simulation only for the current Sprite, and (3) **SA_stop** actions for stopping all scripts for all Sprites. In addition, **change_view** actions occur when a student changes the working view from one Sprite to another. Since more complex programs tend to have multiple Sprites, this action provides important information about the context of model-building.

We aggregated these results across the four days of group work during the intervention, using the Gini coefficient [4] as a means of comparing the distribution of actions by different students within a particular group. Spearman's ρ [19] was then used to compare results. As a smaller Gini coefficient result indicates more equality in action distribution, a descending approach was used for ranking results. All other categories were treated in the usual ascending manner. The Benjamini-Hochberg (B-H) procedure [1] ($Q = 0.25$) was used for group and individual results separately to control for false positives.

To be considered a group, each member had to contribute at least one action to at least one group project, completed the pre-post-test, and worked together for at least three of the four collaborative days. This process resulted in twelve groups ($n = 12$) with sufficient data to analyze—six dyads and six triads.

For computing the number of actions by each group member, we first excluded any projects that at least one group member did not contribute to. Then, groups were evaluated based on their Gini coefficient, the average number of group actions taken per group member per day (Group Actions), the average ANC of all group members (Average ANC), and the average number of each category of actions taken per group member per day (for example, Group SA_local Actions). In total, eleven tests of significance were conducted.

We also analyzed students at the individual level, to observe if holding particular self-appointed responsibilities within a group improved their own conceptual knowledge as a result. We started with the thirty students making up the groups from the previous analysis. One was disqualified due to perfect scores on the pre-post-tests (resulting in no observable ANC), leaving twenty-nine ($n = 29$) for final analysis. The students were evaluated on pre-post growth in terms of ANC, actions they took as individuals while working on group projects (Individual actions), and the percentage of actions taken by an individual relative to their group (Individual Share of Actions). These measures were further divided into the six categories of actions provided above, resulting in fourteen tests of significance.

4 Results

We begin with a breakdown of the actions performed by students that fell under the previously detailed criteria for inclusion: (1) SA_local = 16,185 actions; (2) SC_computational = 27,377 actions; (3) change_view = 2,490 actions; (4) SA_global = 2,221 actions; (5) SC_conceptual = 782 actions; and (6) SA_stop = 918 actions. The majority of group actions taken were a combination of SA_local and SC_computational (75+% for every group), as well as 75+% for all but one individual student (60+% for that student).

The results for groups as a whole appear in Table 1. The lone significant result ($p < 0.05$) was the relationship between the Gini coefficient and the number of actions taken by a group divided by the number of group members and the number of days that group worked together ($\rho = 0.61, p = 0.04$). However, B-H analysis indicated that this was a false positive. All other results had weak correlations, indicating that no group categories had a significant impact on the ANC of students in those groups.

Table 1. Most significant correlation coefficients of group-based results

Variable 1	Variable 2	Spearman's ρ	p-value
Gini coefficient	Group actions	0.61	0.04
SA_global	Average ANC	−0.34	0.29
Gini coefficient	Average ANC	0.21	0.50
Group actions	Average ANC	0.19	0.56

Finally, we analyze results for individual students (though still within the context of their group work), seen in Table 2. The primary result of significance ($p < 0.01$) compared the average normalized change for each student to the percentage of actions that fall within the SC_computational category ($\rho = 0.47, p = 0.009$). Post hoc B-H procedure confirmed the validity of this result, though it rejected the apparently significant ($p < 0.05$) result corresponding to a student's quantity of SC_computational actions per day ($\rho = 0.38, p = 0.04$). Other results presented here were only weakly positively correlated and were not statistically significant.

Table 2. Most significant correlation coefficients of individual-based results

Variable 1	Variable 2	Spearman's ρ	p-value
ANC	**Share of SC_computational actions**	**0.47**	**0.009**
ANC	Individual SC_computational actions	0.38	0.04
ANC	Overall share of actions	0.31	0.10
ANC	Individual SA_global actions	0.30	0.11
ANC	Overall actions	0.28	0.14

5 Conclusions and Future Work

Our findings indicate that students who heavily participated in model building (SC_computational actions) experienced some pre-post-test gains. This will inform future work as we seek to more systematically evaluate log actions while incorporating both more advanced techniques such as differential sequence mining [8,22] and additional data sources for more accurate action counts. We also seek to combine this log-based approach with discourse analysis [16,21] to create a comprehensive framework for analyzing students during synchronous, collaborative programming tasks - particularly during solution construction.

Acknowledgements. This material is based in part upon work supported by National Security Agency Science of Security Lablet H98230-18-D-0010 and National Science Foundation grants CNS-1644848, CNS-1521617, and DRL-1640199. Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the US Government.

References

1. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. Roy. Stat. Soc.: Ser. B (Methodol.) **57**(1), 289–300 (1995). https://doi.org/10.2307/2346101
2. Boyer, K.E., Dwight, A.A., Fondren, R.T., Vouk, M.A., Lester, J.C.: A development environment for distributed synchronous collaborative programming. In: ITiCSE (2008). https://doi.org/10.1145/1384271.1384315
3. Broll, B., et al.: A visual programming environment for learning distributed programming. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, pp. 81–86 (2017). https://doi.org/10.1145/3017680.3017741
4. Dorfman, R.: A formula for the GINI coefficient. Rev. Econ. Stat., 146–149 (1979). https://doi.org/10.2307/1924845
5. Emara, M., Tscholl, M., Dong, Y., Biswas, G.: Analyzing Students' Collaborative Regulation Behaviors in a Classroom-integrated Open Ended Learning Environment. International Society of the Learning Sciences, Philadelphia (2017)
6. Hanks, B., Fitzgerald, S., McCauley, R.A., Murphy, L., Zander, C.: Pair programming in education: a literature review. Comput. Sci. Educ. **21**, 135–173 (2011). https://doi.org/10.1080/08993408.2011.579808
7. Infante, C., Hidalgo, P., Nussbaum, M., Alarcón, R., Gottlieb, A.: Multiple mice based collaborative one-to-one learning. Comput. Educ. **53**(2), 393–401 (2009). https://doi.org/10.1016/j.compedu.2009.02.015
8. Kinnebrew, J.S., Segedy, J.R., Biswas, G.: Integrating model-driven and data-driven techniques for analyzing learning behaviors in open-ended learning environments. IEEE Trans. Learn. Technol. **10**(2), 140–153 (2015). https://doi.org/10.1109/TLT.2015.2513387
9. Lédeczi, Á., et al.: Teaching cybersecurity with networked robots. In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, pp. 885–891 (2019). https://doi.org/10.1145/3287324.3287450
10. Liebenberg, J., Mentz, E., Breed, B.: Pair programming and secondary school girls' enjoyment of programming and the subject information technology (IT). Comput. Sci. Educ. **22**(3), 219–236 (2012). https://doi.org/10.1080/08993408.2012.713180
11. Lim, S.: Implementing social learning for more equitable collaboration in introductory computer science education (2019). https://doi.org/10.7298/hjfz-t152
12. Marx, J.D., Cummings, K.: Normalized change. Am. J. Phys. **75**(1), 87–91 (2007). https://doi.org/10.1119/1.2372468

13. Müller, M.M.: A preliminary study on the impact of a pair design phase on pair programming and solo programming. Inf. Softw. Technol. **48**(5), 335–344 (2006). https://doi.org/10.1016/j.infsof.2005.09.008. http://www.sciencedirect.com/science/article/pii/S0950584905001412. eASE 2005
14. Roschelle, J., Teasley, S.D.: The construction of shared knowledge in collaborative problem solving. In: O'Malley, C. (ed.) Computer Supported Collaborative Learning. NATO ASI Series, vol. 128, pp. 69–97. Springer, Berlin (1995). https://doi.org/10.1007/978-3-642-85098-1_5
15. Shah, N., Lewis, C., Caires, R.: Analyzing equity in collaborative learning situations: a comparative case study in elementary computer science. International Society of the Learning Sciences, Boulder (2014). https://doi.org/10.22318/icls2014.495
16. Snyder, C., Hutchins, N., Biswas, G., Emara, M., Grover, S., Conlin, L.: Analyzing students' synergistic learning processes in physics and CT by collaborative discourse analysis (2019). https://doi.org/10.22318/cscl2019.360
17. Tsan, J., Lynch, C.F., Boyer, K.E.: "Alright, what do we need?": A study of young coders' collaborative dialogue. Int. J. Child-Comput. Interact. **17**, 61–71 (2018). https://doi.org/10.1016/j.ijcci.2018.03.001. http://www.sciencedirect.com/science/article/pii/S2212868917300387
18. Werner, L., Denner, J., Campe, S., Torres, D.M.: Computational sophistication of games programmed by children: a model for its measurement. ACM Trans. Comput. Educ. **20**(2) (2020). https://doi.org/10.1145/3379351
19. Wissler, C.: The spearman correlation formula. Science **22**(558), 309–311 (1905)
20. Yett, B., et al.: A hands-on cybersecurity curriculum using a robotics platform. In: Proceedings of the 51st ACM Technical Symposium on Computer Science Education, pp. 1040–1046 (2020). https://doi.org/10.1145/3328778.3366878
21. Zakaria, Z., et al.: Collaborative talk across two pair-programming configurations. In: International Society of the Learning Sciences (ISLS), June 2019. https://doi.org/10.22318/cscl2019.224
22. Zhang, N., Biswas, G.: Understanding students' problem-solving strategies in a synergistic learning-by-modeling environment. In: Penstein Rosé, C., et al. (eds.) AIED 2018. LNCS (LNAI), vol. 10948, pp. 405–410. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93846-2_76