International Society of the Learning Sciences

Shared struggle: Supporting intercultural collaborative problem solving in massive online computer science courses

Maxwell Bigman, Stanford University, mbigman@stanford.edu
Roy Pea, Stanford University,roypea@stanford.edu

Abstract: Synchronous collaborative problem solving is underutilized in MOOCs. The *PearProgram* tool for collaboratively writing computer programs helped struggling students in an introductory computer science (CS) MOOC achieve statistically significant improved academic and social-emotional outcomes compared to struggling students not using PearProgram. In our structuring collaborative work for the "dual-problem space" (Barron, 2003), PearProgram dyads navigated both relational and problem-solving aspects of collaboration. The paper highlights value in framing a "shared struggle" for learners to re-orient a traditional problem-solving focus.

PearProgram: A Theory Driven Tool

PearProgram (PP) is a virtual learning tool for introductory CS students learning to pair program (Bigman et al., 2021). Pair programming is a well-documented K-16 CS education best practice (Denner et al., 2014; Umapathy & Ritzhaupt, 2017; Williams et al., 2000), yet we lack virtual learning studies of pair programming. If implemented properly, it positively affects student program assignment grades, exam scores and persistence, especially in intro CS classes (Salge & Berente, 2016; Umapathy & Ritzhaupt, 2017). Pair programming fosters greater confidence (Hanks et al., 2011), deeper understanding (McDowell et al., 2006) and further interest (Werner et al., 2004). The major *PP* design contribution is to structure the collaboration around Barron's (2003) "dual space model" of collaborative problem solving: "Collaboration might productively be thought of as involving a dual-problem space that participants must simultaneously attend to and develop a content space (consisting of the problem to be solved) and a relational space (consisting of the interactional challenges and opportunities)." (p. 310). Though many CS education and CSCL tools help learners with content, *PP* is one of the first CS education tools highlighting the social nature of learning, explicitly supporting relational aspects of collaborative problem solving. *PP* supports students navigating the dual-problem space by foregrounding mutual learning and accountability (Cohen & Lotan, 2014), roles (Langer-Osuna, 2017), and features to support joint attention (Schneider & Pea, 2013) through embedded teaching tips (Walker et al., 2014). PP supports the seven CSCL affordances (Hmelo-Silver et al., 2017), with special emphasis on structuring the collaborative learning process and supporting knowledge co-construction through embedded communication tools (chat and video).

Study and Results

After *PP* pilot success in online CS courses (Bigman et al., 2021), it was offered as an intervention for struggling students in *Code in Place*—an introductory online programming class based on the Stanford CS1 course in Spring 2020 and 2021. The course enrolled 12,000+ students with 900+ volunteer global teachers each offering. It ran five weeks as synchronous weekly small group meetings supported asynchronous content assignments, "creating a human-centered learning model built around a community that stresses learning for all, the importance of kindness, and peer support" (Piech et al., 2021). *PP* was offered as opt-in intervention during week 3 of *Code in Place*. Of ~2800 "struggling" students who'd started but didn't complete the 1st assignment, 100 students entered a *PP* session. On study completion, 12 Zoom user interviews were recorded with consent and auto-transcribed, checked with raw audio files using grounded theory.

Academic benefits. The pass rate of non-PP "struggling" students was 21.5% vs 50% ($p < .00001$) for *PP* students. PP students did better on assignments, completing a 2nd assignment with 79% average pass rate vs 47% for non-PPs ($p < .00001$). 57% of PP Students completed the 3rd assignment vs 24% for non-PPs ($p < .00001$). *PP* uses are thus associated with better retention and academic outcomes for struggling students.

Social benefits. Interviews indicated *PP* students had social-emotional learning benefits: increased motivation (12 instances), confidence (10), psychological safety (15). Students emphasized enjoying *PP* sessions (8 instances), and valuing meeting people from other cultures (12). Students found value in the dual-problem space framing as it led to partners that showed respect, understanding, and patience (16, e.g. "What I noticed, like, even when you're working you can't really separate a person from their work."

Implications and future directions

Importance of shared struggle. As students worked together collaboratively problem solving, they seemed to benefit from PP's framing of the programming problem as something to conquer together—not an individual achievement task. Students saw the problem to be overcome as a team: "We're all kind of struggling together, we're all clinging to the same kinda lifeboat, and that makes it easier for people to work through the problems that they're having." His notion of "shared struggle" suggests an important reframing of collaborative problem solving in CSCL tasks, perhaps especially useful in online courses with intercultural learners.
Future research: PP dyads navigated the dual problem space diversely—some focused on relational issues before substantive pair programming work, others dived into programming work and worked out emerging relational issues. A fertile CSCL research direction is to identify the most effective of many possible paths pairs can negotiate in the dual-problem space, depending on factors like prior knowledge, dispositions and social expectations. We aver scaffolding technology will play important roles supporting the research and learning of pairs while navigating the dual-problem space in virtual collaborative learning.
Future technology**.** *PP* currently supports students as participants in a system of distributed intelligence (Pea, 1993) as their scaffolds never fade (Pea, 2004): "Scaffolds are not found in software but are functions of processes that relate people to performances in activity systems over time" (p. 446). We should learn "what processes of fading need to be employed" (op. cit) to sustain learners' autonomous performance of the capabilities in question in transfer situations mattering for both content and relations. We foresee employing adaptive collaborative learning supports (Rummel et al. 2008) for PP content *and* relational issues.

References

Barron, B. (2003). When smart groups fail. *Journal of the Learning Sciences*, *12*(3), 307–359.

Bigman, M., Roy, E., Garcia, J., Suzara, M., Wang, K., & Piech, C. (2021). PearProgram: A more fruitful approach to pair programming. *Proc. 52nd ACM Technical Symposium on Computer Science Education*, 900–906.

Cohen, E. G., & Lotan, R. (2014). *Designing groupwork: Strategies for the heterogeneous classroom*.

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education*, 277–296.

Hanks, B., Fitzgerald, S., & Murphy, L. (2011). *Pair programming in education: A literature review*. 40.

Hmelo-Silver, C., Jeong, H., Faulkner, R., & Hartley, K. (2017, January). Computer-supported collaborative learning in STEM domains: Towards a meta-synthesis. In *Proceedings of the 50th HICSS,* 2066-2075.

Langer-Osuna, J. M. (2017). Authority, identity, and collaborative mathematics. *Journal for Research in Mathematics Education*, 48(3), 237-247.

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2003, May). The impact of pair programming on student performance, perception and persistence. *Proceedings of 25th International Conference on Software Engineering* (pp. 602-607). IEEE.

Pea, R. D. (1993). Practices of distributed intelligence and designs for education. In G. Salomon (Ed.). *Distributed cognitions (pp. 47-87)*. New York: Cambridge University Press.

Pea, R. D. (2004). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. *Journal of the Learning Sciences*, *13*(3), 423–451.

Piech, C., Malik, A., Jue, K., & Sahami, M. (2021). Code in Place: Online Section Leading for Scalable Human-Centered Learning. *Proceedings 52nd ACM Technical Symposium-Computer Science Education*, 973–979.

Rummel, N., Weinberger, A., Meier, A., Voyiatzaki, E., Spada, H., Avouris, N., Walker, E., Rose, C., Kumar, R., Gweon, G., Wang, Y.-C., & Joshi, M. (2008). *New challenges in CSCL: Towards adaptive script support*.

Salge, C. A. D. L., & Berente, N. (2016, January). Pair programming vs. solo programming: What do we know after 15 years of research? *49th Hawaii International Conference on System Sciences,* 5398-5406.

Schneider, B., & Pea, R. (2013). Real-time mutual gaze perception enhances collaborative learning and collaboration quality. *International Journal of Computer-Supported Collaborative Learning*, 375–397.

Umapathy, K., & Ritzhaupt, A. D. (2017). A meta-analysis of pair-programming in computer programming courses: implications for educational practice. *ACM Transactions on Computing Education*, *17*(4),1-13.

Walker, E., Rummel, N., & Koedinger, K. (2013). Adaptive intelligent support to improve peer tutoring in algebra. *International Journal of Artificial Intelligence in Education*, *24*, 33-61.

Werner, L. L., Hanks, B., & McDowell, C. (2004). Pair-programming helps female computer science students. *Journal on Educational Resources in Computing*, *4*(1), 4.

Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *IEEE Software*, *17*(4), 19–25.