Peer Conversations about Refactoring Computer Code: Negotiating Reflective Abstraction through Narrative, Affect, and Play

Oia Walker-van Aalst, David DeLiema, Virginia J. Flood, & Dor Abrahamson
UC Berkeley Graduate School of Education

Abstract
In computer programming, reflective abstraction (Piaget, 2001) is a near constant while writing computer code. Programmers must perpetually refactor existing code, dramatically restructuring it without changing its outcome. This paper investigates how middle school students new to programming frame occasions for refactoring code during peer-to-peer interactions. This research took place in free-of-charge, two-week summer coding workshops (M-F 9am - 4pm) for 5th - 10th grade students from high-poverty urban communities. Using laptop screen recordings and GoPro recordings of gesture and body movement, we marked, transcribed, and interpreted moments of refactoring using conventions of interaction analysis (e.g., Goodwin, 2013). Our fine-grained analysis of a refactoring episode, representative of refactoring conversations in our larger data set, demonstrates how processes of reflective abstraction can be socially organized. We find that learners playfully and affectively frame the risks and benefits of refactoring, using narrative to negotiate the value of dramatically restructuring working code. Our analysis illustrates how the microgenetic process of organizing thought into higher planes of professional competence is an irreducibly social and affective process, with implications for how students manage risk in their developing coding practice.

Introduction
In reflective abstraction (Piaget, 2001), existing knowledge structures are dramatically restructured through explicit reflection to cope with radically new situations where accommodation and assimilation are insufficient. In computer programming, reflective abstraction is a ubiquitous necessity at all levels of expertise. Programmers must perpetually refactor existing code—dramatically restructuring it without changing its outcome—to reduce unnecessary complexities and improve readability.

Refactoring code is challenging. It requires inventing new abstractions, a central component of developing computational thinking skills (K–12 Computer Science Framework, 2016; Shute, Sun, & Clarke, 2017; Wing, 2008). Like all processes of reflective abstraction, refactoring is also risky (Murphy-Hill, Zimmermann, Bird, & Nagappan, 2014). Altering a working program can introduce bugs and cause it to break. Refactoring places coders in a state of "cognitive limbo" (Abrahamson, 2012) and both students and professional programmers must always weigh its costs and benefits.

In this paper, we investigate how *middle school students new to programming* frame occasions for refactoring code during peer-to-peer interactions in a summer coding camp. Our fine-grained analysis of a refactoring episode demonstrates how processes of reflective abstraction can be socially organized. We find that learners playfully and affectively frame the risks and benefits of refactoring, using narrative to negotiate the value of dramatically restructuring working code. Our analysis illustrates how the microgenetic process of organizing thought into higher planes of professional competence is an irreducibly social and affective process.

Research design and analysis
This research took place in free-of-charge, two-week summer coding workshops (M-F 9am - 4pm) for 5th - 10th grade students (n=120) from high-poverty urban communities. Classes were led by undergraduate computer-science majors new to teaching. We report on students' work in PixelBots, a programming environment in which students wrote JavaScript to control the movement and painting actions of an animal avatar around a two-dimensional grid, relying on coding concepts such as sequences, function calls, arguments, parameters, loops, and function definitions. Using laptop screen recordings and GoPro recordings of gesture and body movement, we marked, transcribed, and interpreted moments of refactoring using conventions of interaction analysis (e.g., Goodwin, 2013). We conducted five case studies of students' coding practices and zeroed in on moments of refactoring. Our analysis aimed to

understand the social context within which reflective abstraction can take place. We report here on one episode to provide the generative contours of our argument.

Findings

Two middle-school girls are working on writing loops in JavaScript to code how a PixelBot paints repeating sequences of color along each side of a square. Aya has written a program that paints 3/4 of the target image when she decides to refactor her code down from multiple loops into one loop. Aya announces to Van that she is currently deleting all of her code. Van exclaims, "Nooo!" and asks: "Do you regret your work?" and "Did you make a big mistake?" Aya responds with an emphatic "No" to each question, brushing Van's accounts aside. However, when Aya runs her newly simplified, refactored code, the PixelBot makes an errant right turn and begins painting color in the wrong grid squares. Watching, Van immediately reacts, "Whaat!? Why did you turn right—", and Aya exclaims, "Oops!" Both girls begin laughing. Van continues to ask Aya why she made her PixelBot turn right, and Aya explains in an exaggerated voice of resignation, "I messed up...I'm not supposed to do that. Now I regret it."

This brief interaction demonstrates how reflective abstraction is shaped within a rich socio-material ecology (Erickson, 1996) and framed by peers through narrative processes of storytelling in the moment. We note Van's multiple moves to narratively situate Aya's choice to refactor, which foreground issues of regret, past mistakes, and intention. These moves develop a picture of refactoring as a risky and possibly regrettable course of action. But Aya rebuffs Van's first two causal propositions, jointly negotiating the narrative space as the interaction unfolds (DeLiema, 2017; Ochs, Taylor, Rudolph, & Smith, 1992). These narrative moves co-occur with and make relevant public affective stances— "No!" "What!?" and "Oops!"—which show brief forms of protest and surprise. These exclamations serve as unfolding meta-epistemic affective stances (Jaber & Hammer, 2016)—feelings about the process of developing knowledge—on the risk of refactoring code and the materialization of an unforeseen logic error. The laughter, exaggerated reactions, and funny voices from the students set a playful tone/key in this exchange (Goffman, 1974) creating a dramatic but ultimately playful and safe space for cultivating an orientation to refactoring (Steen & Owens, 2001).

In the next moment of this exchange, when Aya attempts once again to delete her code, Van leans over and blocks Aya from doing so, saying, "Don't erase everything!" The two playfully tussle away from their laptops while laughing, leaning into one another and gently clashing with their hands and arms. In these moments, Aya and Van are jostling over whether to apply the moral of the story they just constructed to the next coding action. The girls are playfully asking: Should the regret Aya just experienced carry forward into her next move, constraining her choice to take another bold action?

Discussion

The episode discussed above, and others in our data set, show how the practice of refactoring code, representative of the more general process of reflective abstraction, occurs in peer-to-peer interaction within the context of collaboratively constructed narratives, affective stances, and playful keys. Our other examples of refactoring mix these three elements in unique and context-specific ways. These socially constructed threads that interweave the process of reflective abstraction are important to understand because they shape how students make meaning of their attempts to push their knowledge forward under risky conditions and serve the students as possible lessons learned for their next coding opportunities. Moving forward as a field, we need to better understand whether these micro narratives persist over time within dyads, become part of the broader classroom culture, evolve over time, change within the key of play, and shape students' ongoing decisions about pursuing reflective abstraction.

References

Abrahamson, D. (2012). Rethinking intensive quantities via guided mediated abduction. *Journal of the Learning Sciences, 21*(4), 626-649.

DeLiema, D. (2017). Co-constructed failure narratives in mathematics tutoring. *Instructional Science, 45*(6), 709-735.

Erickson, F (1996). Going for the zone: The social and cognitive ecology of teacher-student interactions in classroom settings. In D. Hicks (Ed.), *Discourse, learning, and schooling* (pp. 29-62). Cambridge, England: Cambridge University Press.

Goffman, E. (1974). *Frame analysis: An essay on the organization of experience*. Cambridge, MA: Harvard University Press.

Goodwin, C. (2013). The co-operative, transformative organization of human action and knowledge. *Journal of Pragmatics, 46*, 8–23.

Jaber, L. Z., & Hammer, D. (2016). Engaging in science: A feeling for the discipline. *Journal of the Learning Sciences, 25*(2), 156-202.

K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org.

Murphy-Hill, E., Zimmermann, T., Bird, C., & Nagappan, N. (2015). The design space of bug fixes and how developers navigate it. *IEEE Transactions on Software Engineering, 41*(1), 65-81.

Ochs, E., Taylor, C., Rudolph, D., & Smith, R. (1992). Storytelling as a theory-building activity. *Discourse Processes, 15*, 37–72.

Piaget, J. (2001). *Studies in reflecting abstraction*. Sussex, England: Psychology Press.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142-158.

Steen, F. F., & Owens, S. A. (2001). Evolution's pedagogy: An adaptationist model of pretense and entertainment. *Journal of Cognition and Culture, 1*(4), 289-321.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, 366(1881), 3717-3725.