Reflection as Formative Assessment of Computational Thinking in Elementary Grades

Ha Nguyen, Leiny Garcia, Sharin Jacob, Debra Richardson, and Mark Warschauer
thicn@uci.edu, leinyg@uci.edu, sharinj@uci.edu, djr@ics.uci.edu, markw@uci.edu
University of California, Irvine

Abstract: Capturing how students develop computational thinking is critical to integrating computer science education into school settings. This paper examines the use of Flipgrid, a student-facing video platform, to invite students to reflect on their programming artifacts and document their computing language and practices. Data sources include 63 reflection videos and 41 programming projects in two fourth grade classes. We apply discourse analysis and find that student displayed enriched discourse in vocabulary and practices, but not CS concepts, in response to explicit prompts. Frequencies of vocabulary use and connection to programming narratives in reflections were strongly correlated with programming artifact scores (overall score; sub-scores in user interactivity, mechanics, and computer science constructs). We discuss implications for leveraging reflection tools in computer science instruction in K-12 classrooms.

Introduction

Methods to assess computational thinking (CT) in K-12 classrooms have received growing attention in the computing education and learning sciences community (Tissenbaum et al., 2014). To this end, researchers have leveraged artifact-based reflection interviews to capture programming process and CT development (Brennan & Resnick, 2012). However, administering interviews is time and human-resource intensive, and thus interviews usually target a focal group of students. This approach may not be optimal for teachers who want to collect whole-class information to make instructional adjustments. In this study, we invited elementary school students to create short reflections about their programming artifacts on Flipgrid, a student-facing video platform. **RQ1.** What types of CT discourse did students engage in when given certain prompts? **RQ2.** What are the associations between the frequency of discourse types and dimensions of CT proficiencies, as revealed in their programming artifacts?

Theoretical framework: Role of student reflections and scaffolds for reflections

Socio-cultural interactions play a crucial role in the development of individuals' cognitive functioning (Lave & Wenger, 1998). Rather than passive absorption, knowledge is situated in dynamic socio-cultural communities and activities (Lave & Wenger, 1998). The socio-cultural framework informs educational approaches that promote learner reflection and discourse, such as tapping into students' epistemological and conceptual beliefs in think-aloud reflection prompts to assess their understanding of science domains (Peters & Kitsantas, 2010).

Student reflection provides a venue to evaluate their CS conceptual understanding alongside other forms of assessments (Grover & Pea, 2013). Common forms of CT assessment include quizzes, scoring of programming projects, and artifact-based interviews (Brennan & Resnick, 2012). While artifacts represent evidence of students' competencies, educators and researchers may not be able to gauge the design decisions behind the artifacts by only looking at the final projects (Grover, Basu, & Schank, 2018). Additionally, scoring all student projects and coding in-depth interviews place extra burden on educators with already limited time and resources (Basu, 2019).

When using reflection as assessment, educators can use scaffolds to make inquiries and domain knowledge that are implicit to learners explicit (Lin & Puntambekar, 2019). When students are practicing argumentative discourse, explicit scaffolds (e.g., clear standards for arguments) are effective in improving students' understanding of argumentation criteria (Ryu & Sandoval, 2011). Conceptual and metacognitive scaffolds that guide students to reflect on their approaches may encourage students to develop strategies and integrate knowledge more effectively than procedural prompts that focus on specific activities (Davis, 2001).

Methods

Study setting, participants, data sources

This study is situated in a district-wide initiative in California to integrate CS education into elementary schools with a large student population of Latinx (94%) and of low socioeconomic status. A Scratch-based curriculum was adapted to align with English Language (ELD) state standards for multilingual learners of the district. This was implemented across five schools by teachers who had no formal training in CS and students who did not have formal CS experiences. The study took place in two fourth grade classes in the same school, where teachers Ellen

& Helen (pseudonyms) voluntarily used Flipgrid as a supplement to their reflection activities.

This study draws from two data sources to inform students' CT. Flipgrid reflection videos (n = 62) come from two reflection activities where students gave a short presentation on their end-of-unit projects (1.15 minutes on average). For Unit 1, Hellen and Ellen had students reflect in pairs to the same prompt: "Walk us through your project. What blocks did you use? What was difficult? What did you find?" For Unit 2, the teachers had students reflect individually on a digital collage they created about themselves. Ellen asked her students: "What did you learn from your classmates' feedback? What did you discover from looking at others' projects? Use programming language such as parallel." Meanwhile, Helen, whose class had more students with special needs, used a more explicit prompt for tasks: "Briefly state some details about you that are in the program. Then, pick ONE SPRITE, open the code, and explain what you did. Be sure to use vocabulary like 'algorithm,' 'parallel', ..."

Programming artifacts (n = 41) that correspond to the video responses were rated based on a validated rubric for secondary classrooms (Basu, 2019). The rubric consisted of four dimensions: Overall complexity (i.e., code readability, novelty, program completeness), user interactivity (i.e., user input, keyboard interactivity, media or special effects), mechanics (i.e., sprite motion), and coding constructs (e.g., loop, conditionals). Each dimension receives a score of 0 to 3 for increasing proficiency. We paired the artifacts and reflections from unit 2 because there is little variation in unit 1 artifacts, when students just started with Scratch and were using the same codes.

Table 1: Examples of coding scheme for CT concepts and practices: emerging and expanding discourse

	Content	**Emerging** Hint on a category	**Expanding** Link categories to desired outcomes
CONCEPT	**Loop** Repetition of a code block or script until a condition is met.	JM: And we basically made him do it. And we basically made him repeat it 10 times.	MG: I just put *switch costume* to Toucan-C and *move 10 steps* and then I put *wait 0.1 seconds* and then I repeat. And then the next time […] and I repeat all of this.
	Parallelism Having multiple scripts run at the same time.	AL: My parallel events include my sprites moving while they talk, which include my voice, and music playing in the background.	JAG: So basically parallel programming is when two things, two sprites, are basically doing the same thing at the same time. Like one cat is dancing while the other cat is moving 10 steps and then reversing 10 steps
PRACTICE	**Abstraction and Modularization**	LP: I made bubble sound because the fish goes in the water.	HC: […] I play a block that says when you click play, it says I love to sing for 2 seconds, wait 2 seconds, switch custom to Annie 3 and say my favorite color is green and switch back.
	Iteration and Experimentation	IG: What we discovered is how to change the coordinates and where to put the coordinates and size.	AL: I discovered that you can make your sprite change as it moves. One sprite. The next sprite. I also discovered that you can write words in several forms.

Analytic approaches

RQ1. We performed discourse analysis of students' Flipgrid reflections based on frameworks for CT (Brennan & Resnick, 2012). These include vocabulary specific to Scratch (e.g., stage, costume, glide); CS vocabulary (e.g., code, algorithm); CS practices (i.e., abstraction & modularization; experimentation; debug); CS concepts (i.e., loop, event, sequence, parameter, parallelism, initialization), and elaboration of a storyline tied to the projects. The storyline and Scratch vocabulary are not exact indicators of CT, yet they provide insight on student application of CT. The storyline reflects the student's narrative and desired outcome of the Scratch program, while the Scratch vocabulary reflects the extent to which students engaged with the programming activities. All categories were coded for frequency (i.e., counts of category over total CT counts for each reflection). CS practices and concepts were further coded for level of elaboration that aligned with ELD levels to demonstrate students' depth of computational knowledge (see Table 1 for examples; not all categories are included due to space limit). In this study, "Emerging" responses mention a category, whereas "expanding" elaborate on how a category connects to a desired outcome or purpose. Occurrences when students were only reading the code blocks without any insight were not counted. We performed Mann-Whitney U tests to explore whether students' discourse differed within classrooms between prompts. To establish interrater reliability, the first and second authors coded 20% of the data, resolved conflicts, and coded another 20% of the data. After Cohen's kappa of at least .82 was reached for each category, each coder coded half of the remaining data.

RQ2. We performed Spearman correlation tests to examine the association between the frequency of CT dimensions (i.e., vocabulary, concepts, practices, storylines) in reflections and their programming artifacts scores.

Hypotheses. RQ1: Reflection prompts linked to explicit practices and concepts may be related to more frequent discourse in those criteria (Ryu & Sandoval, 2011). **RQ2**: There would be moderate, positive correlations between CT dimensions and artifact scores (Basu, 2019).

Findings

Reflections revealed varied CT patterns in relation to prompts

Table 2: Student CT discourse frequencies (over total lines of talk) in video reflections

	Overall (n = 63)		**Unit 1 (n = 22)**		**Unit 2 (n = 41)**		**W *(p) Ellen***	**W *(p) Helen***
	Unit 1	Unit 2	Ellen	Hellen	Ellen	Hellen		
CS vocabulary	.09 (.13)	.10 (.13)	.08 (.10)	.14 (.17)	.11 (.13)	.08 (.13)	111 *(.68)*	89.5 *(.21)*
Scratch vocabulary	.33 (.19)	.41 (.25)	.29 (.19)	.37 (.20)	.36 (.31)	.44 (.20)	111 *(.71)*	148.5 *(.29)*
CS Concept	.18 (.15)	.16 (.19)	.16 (.13)	.20 (.18)	.19 (.23)	.14 (.15)	106.5 *(.86)*	98 *(.40)*
CS Practice	.12 (.10)	.08 (.12)	.13 (.11)	.10 (.09)	.06 (.08)	.10 (.14)	62.5 *(.07)*	106 *(.59)*
Storyline	.13 (.10)	.10 (.08)	.16 (.13)	.09 (.06)	.09 (.10)	.10 (.07)	67 *(.12)*	133 *(.63)*
Story elaboration	.14 (.13)	.14 (.13)	.17 (.16)	.10 (.06)	.13 (.14)	.14 (.12)	89.5 *(.59)*	139 *(.48)*

Note. Standard deviations in parentheses. Mann-Whitney tests indicated differences within class between units.

Overall, the use of Scratch vocabulary was the most frequent type of CT discourse across units and classrooms (M = .33, SD = .19 in unit 1; M = .41, SD = .25 in unit 2; Table 2). Students often began their reflection by explaining their use of code blocks and demonstrating their programs. The frequencies for the other types of discourse were roughly balanced. Although to a lesser extent than Scratch vocabulary, students conveyed application of CS concepts (mostly loops, sequence, events, and parallelism), practices (particularly abstraction and experimentation), and connections to narratives (storyline and story elaboration) in their reflections.

We examined whether students' CT discourse differed between the two units/prompts (Figure 1). In the second unit, students were introduced to additional CS concepts, namely event, parallelism, and sequence. Mann-Whitney U tests indicated no difference in the frequency and elaboration of CS concepts within classes. There was an overall increase in Ellen's class for emerging talks in experimentation and iteration ($p < .05$).

More vocabulary and storyline use were generally related to higher CT proficiencies

Spearman's correlation tests indicate a strong association between the frequency of Scratch vocabulary in reflections and overall artifact scores ($r_s = .40$, $p = .02$), as well as the subdomains—user experience ($r_s = .59$, $p < .001$) and concepts ($r_s = .41$, $p = .02$). Interestingly, the presence of a storyline and the extent to which students elaborated on their narratives show a strong, positive correlation with students' overall scores and several subdomains (user experience, concepts, and mechanics). An interpretation of these findings is that students who experimented with ways to engage users and navigate objects (i.e., sprite) seemed to be able to elaborate on these aspects in their reflections, using both the Scratch interface vocabulary and their everyday language.

Discussion: Implications for curriculum design

Employ explicit prompts. Students' answers appeared to align with certain aspects of the prompt content and indicate emerging levels of elaboration. For example, in unit 2, Ellen's class more frequently used the prompted vocabulary (e.g., "parallel") and described iteration with different codes when discussing what students discovered. However, the results did not conclusively indicate an increase in students' emerging and advanced discourse across all concepts and practices that the prompts included. In place of explicit, static prompts, dynamic scaffolds that adapt to students' understanding, or a combination of both, may support enriched discourse within contexts where students are learning to present their experiences (Lim & Puntambekar, 2019).

Pair reflection with assessments. We found a strong correlation between several aspects of students' CT proficiencies and their use of vocabulary and storyline elaboration. This suggests external validity of the reflection rubric in capturing CT in vocabulary, concepts, and practice, as also present in student programming artifacts. More importantly, findings indicate the affordances of employing reflection as a form of student discourse to represent a more comprehensive view of their CT thinking. When paired with student programming artifacts, we come to understand not only what students designed, but also why and how they created the programs. The link between the presence of a storyline and students' CT proficiencies illustrate the affordance for students to develop language proficiency alongside CT, and apply their everyday and CS language repertoires in the learning of CS.

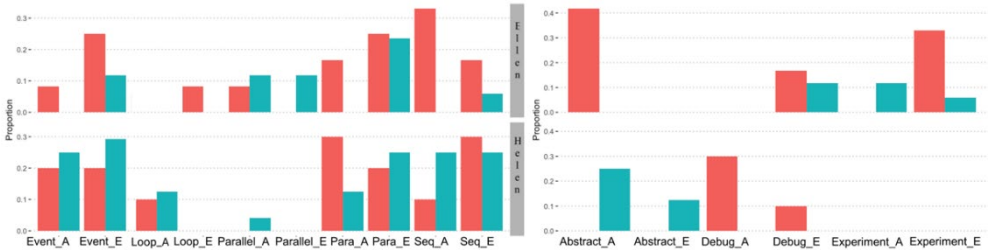

Figure 1. Elaboration of concepts (left) & practices (right). Iterations concept and Reuse/Remix practice omitted from figure due to low presence in student talks. "_E" = Emerging; "_A" = Expanding. Unit 1 = red; 2 = blue

Conclusion

This study examines the use of a reflection tool in elementary CS classrooms. Students who applied both CS vocabulary and everyday narratives in reflections appeared to produce higher-scoring programming artifacts for user interactivity, mechanics, and CS constructs. The study contributes to the emergent work on examining opportunities to promote CS discourse in two ways. We illustrate the use of a coding scheme for CT in connection to language development for upper elementary students. The rubric not only relies on frequency counts, but also the levels of elaboration across several domains of CS and language. We also explore the utility of a video reflection tool in capturing student CT vocabulary, concepts, and practices. Findings about the relation between programming artifacts and student language have implications for designing CS activities for younger learners, who are beginning to acquire both academic language and CS-specific knowledge.

Limitations affect the generalizability of these findings. First, other curricular aspects may have influenced student discourse in the later unit. Second, as the classrooms had unobservable contexts, the analysis was careful to not make any between-class comparisons. Future work will examine differences in reflection among student subgroups, teachers' perceptions of student reflections, and other forms of scaffolds for CT.

References

Basu, S. (2019, February). Using Rubrics Integrating Design and Coding to Assess Middle School Students' Open-ended Block-based Programming Projects. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1211-1217). ACM.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual meeting of the American Educational Research Association, Vancouver, Canada* (Vol. 1, p. 25).

Davis, E. A. (2000). Scaffolding students' knowledge integration: Prompts for reflection in KIE. *International Journal of Science Education*, *22*(8), 819-837.

Grover, S., Basu, S., & Schank, P. (2018, February). What we can learn about student learning from open-ended programming projects in middle school computer science. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 999-1004). ACM.

Grover, S., & Pea, R. (2013, March). Using a discourse-intensive pedagogy and android's app inventor for introducing computational concepts to middle school students. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 723-728). ACM.

Lave, J., & Wenger, E. (1998). Communities of practice. New York: Cambridge University Press.

Lin, F., & Puntambekar, S. (2019). Designing Epistemic Scaffolding in CSCL.

Peters, E. E., & Kitsantas, A. (2010). Self-regulation of student epistemic thinking in science: The role of metacognitive prompts. *Educational Psychology*, *30*(1), 27-52.

Ryu, S., & Sandoval, W. A. (2012). Improvements to elementary children's epistemic understanding from sustained argumentation. *Science Education*, *96*(3), 488-526.

Tissenbaum, M., Sheldon, J., Sherman, M. A., Abelson, H., Weintrop, D., Jona, K., ... & Snow, E. (2018). The state of the field in computational thinking assessment. International Society of the Learning Sciences, Inc. [ISLS].

Acknowledgments

We would like to acknowledge the teachers and students who participated in this study. This work is supported by the National Science Foundation, award 1738825.