

Measurement of Computational Thinking Gains

Michael Yakubovsky

Lamar University

EDLD 5315

Introduction

Computational thinking is a problem-solving methodology that is closely aligned with computer science. Companies and schools are talking about the importance and relevance of computational thinking in curricula. The discussions tend to center around computational thinking being useful in teaching problem solving to students in any content area.

Computational thinking builds problem solving and critical thinking skills in students. These skills have been shown to improve learning and retention for students.

For teachers wishing to implement computational thinking into their classroom, it is important that the learners still master the course material. At the same time, teachers should be measuring the gains in computational thinking skills as a gauge of the effectiveness of the additional components. Measuring computational thinking and problem-solving skills is not as straightforward as giving a multiple-choice test. The daily demands on teachers do not allow much opportunity to explore ways to implement computational thinking without having a negative impact on the base course curriculum. Teachers need guidance in finding a useful method of assessing growth in computational thinking skills while also assessing the course content. This review is looking at research findings related to the measurement of computational thinking skills.

Overview of studies

Lockwood and Mooney (2018) showed in their research, along with Kalelioglu, Gulbahar, and Kukul (2016) that there is not a lot of definitive research into the application of computational thinking in school curricula. There is even less in secondary schools. The low number of research studies into computational thinking in the classroom has not provided

definitive assessment methodologies yet. The lack of researched and widely used methods of assessing computational thinking has shown to contribute to a lack of significant progress in the utilization of computational thinking in core content areas. Most research found has been within computer science classes (Kaleliouglu, Gulbahar, & Kukul, 2016; Lockwood & Mooney, 2018).

Further complicating implementation, a study by Sherman and Martin (2013) found that most computational thinking assessments are not well suited to new computational thinking models and methods developed over the last few years (Lockwood & Mooney, 2018). These new methods are implementations that are specific to unique use cases, such as in mobile environments. These new developments further complicate matters when developing consistent and widespread assessment tools. Assessing gains in computational thinking skills is challenging. At the time of this study, there were no validated studies into the ability of students to transfer computational thinking skills from one subject to another (Meerbaum-Salant, Armoni, & Ben-Ari, 2010; Basawapatna, Koh, Repenning, Webb, & Marshall, 2011; Werner, Denner, Campe, & Kawamoto, 2012; Grover, Pea, & Cooper, 2015). In spite of the difficulties and challenges, there are many efforts aimed at developing comprehensive computational thinking assessments.

Research is also being conducted into the effectiveness of content delivery within the classroom. Computational thinking activities are tied to current course learning units and not delivered as a stand-alone lesson. Students need to be given opportunities to demonstrate their learning in ways that allow for the unique problem-solving skills utilized by the student. Open-ended projects provide opportunities for students to demonstrate choice in learning and therefore utilize computational thinking strategies. Conversely, these projects are difficult to reliably and objectively assess the level of computational thinking skills. Larger classrooms make this task

even more difficult and evidence suggests that scores tend to skew to one extreme or the other. One-on-one interviews with students show the ability to provide more accurate assessment of computational thinking abilities and growth but are time-consuming as well (Grover, Pea, & Cooper, 2015). The time required to assess computational thinking skills necessitates the development of more useful tools and methods for teachers. Korucu, Gencturk, and Gundogdu (2017) have found several studies that show that it is more effective to assess computational thinking skills within the context of other curricular units and activities. The blending of problem-solving skills and content provide a better context for students to demonstrate the problem solving. Blending the assessment into another pre-existing assessment may provide some time-savings for teachers.

Another widely-used method of assessing computational thinking is to use pre- and post-test assessments. Ary, Jacobs, Razavieh, and Sorensen (2009) found that the use of computational thinking pre-tests presents difficulties for assessment. A phenomenon referred to as pre-test sensitization can occur. Students can become aware of their thought processes simply by the nature of the assessment and not from the lesson that follows. The pre-test can, in some cases, cause a change in thinking among students that does not give a true representation of their learning on the post test (Yadav, Mayfield, Zhou, Hambrusch, & Korb, 2014). These findings do not negate the validity of the assessments, but server as a caution to educators and researchers that they must aware of the impact that the assessment could have and to mitigate as much of the undue influences as possible.

Allen, Barr, Brylow, and Hambrush (2019) found that a blending of levels of educators in the development of computational thinking assessments and activities provides a more useful assessment tool. Involving educators across many levels has shown to have more impact in

creating more useful assessments. Post-secondary faculty have a good understanding of the future needs of high school students, while high school faculty are more aware of the needs and motivations of current high school students. The blended faculty approach has shown to provide results that are more accurate and indicative of long-term success. This approach offers opportunities to also mitigate some of the other issues that have arisen with prior computational thinking assessments.

Assessment of secondary implementations

The focus of this research is centered on secondary school classrooms. It is necessary to narrow the focus from general research about computational thinking to implementations in secondary school curricula. Lockwood and Mooney (2018) report on one study that utilized formative assessments in a robotics unit. The assessments were given to students after four and again after ten robotics lessons. The assessments showed positive growth in both computational thinking skills and in content mastery. The results of the study suggest that students need multiple lessons spanning a number of days before significant results can be seen as a result of computational thinking activities. The study suggests that students need time to process and reflect on the learning that has taken place.

Grover and Pea (2013) conducted a study into the effectiveness of an Android App Developer program in a middle school classroom. That program found evidence suggesting that communication activities play an important role in the development of problem-solving skills in core curriculum. Subsequent research conducted on middle school computer science curriculum was conducted to follow up on the previous findings. The study combined both formative and summative assessments into the final analysis. The study was looking in the feasibility of

students to gain the ability to transfer learning between programming languages of growing complexity (Grover, Pea, & Cooper, 2015).

Most research found, was conducted either in computer programming and computer science classes, or in classes that were specifically teaching a computer programming lesson. It is also important to look at the use of computational thinking in core classes as well. In one particular study, computational thinking skills were evaluated by having students design algorithms that solved several scenarios related to the science lesson. The algorithms were evaluated on completion of the task and inclusion of constructs such as conditionals, loops, and variables. The algorithms were evaluated by comparing them to an “expert” solution. The size of the variance between the algorithms the expert solution indicated the level of incorrectness (Basu, Kinnebrew, & Biswas, 2014; Lockwood & Mooney, 2018).

Assessment tools

As previously discussed, there is not a large volume of data on the use of computational thinking in the classroom. The lack of developed and researched assessments requires researchers to create custom assessments that have limitations in comparing results from multiple research studies. Lockwood and Mooney (2018) reviewed a study in which researchers gave pre- and post-tests to 6th grade students in a science class that were participating in a computational thinking-based activity. The study evaluated student’s gains in both science content and computational thinking skills. The custom-built assessments required researchers to compare student results with “expert” answers. The comparisons were then rated on a 4-point scale to allow for a qualitative analysis of the results (Basu, Kinnebrew, & Biswas, 2014). Another study utilized a similar method to compare two different methods of content delivery,

face-to-face and blended online environments. The summative and formative assessment data were compared to each other to assess the impact of the blended learning option. The researchers were using the assessments to give a common method of analysis for both delivery methods (Grover, Pea, & Cooper, 2015).

A common method of assessing computational thinking learning is to use surveys of participants. The surveys provide qualitative results for researchers. Yadav, et al. (2014) studied a particular secondary curricular unit. Following the unit, students were given a survey to assess the participant's attitude and opinions on computer science. An internal reliability assessment of the survey showed a high enough reliability to suggest that the survey was a useful tool for the study. (Yadav, Mayfield, Zhou, Hambrusch, & Korb, 2014). This method of analysis is not limited to just students in computational thinking activities. Studies into the impacts on teachers provide data as well. One such study looked at the impact that computational thinking activities had on preservice teachers with no prior computer science training. After a week-long computational thinking unit, students were given three open-ended questions designed to assess the level of computational thinking understanding (Yadav, Mayfield, Zhou, Hambrusch, & Korb, 2014).

One study of note by Grover and Pea (2013) used a combination of methods. The study used a mixture of class questioning and discussions to assess the perceived benefit of using App Inventor in relation to other programming environments (Lockwood & Mooney, 2018). The study involved a number of different data collection techniques throughout. Students completed pre- and post-training surveys. The post-training survey captured data related to the student's experience and attitudes regarding the training activities. Along with the surveys, the researchers video recorded the sessions and conducted a screen capture of the student's computer

screen while participating in the training exercises. Along with these data collection methods, the researchers also recorded audio of the students in order to capture more subtle sounds made by the students that were not captured by the video cameras. The number of data collection methods allowed for a triangulation of data to provide an analysis with a larger degree of certainty (Grover & Pea, 2013). This method of data collection provides a great deal of information but requires a more substantial investment of resources to collect and analyze the data.

Recently, more “standardized” assessment tools have been developed and are starting to be used to analyze the effectiveness of computational thinking implementations. Korkmaz and Bai (2019) studied and implementation where Chinese students were administered an adjusted version of the Computational Thinking Scale to assess their growth in computational thinking skills. The test was given to the test group of students twice to obtain a confidence level of the validity of the adjusted test. The test showed to have a strong positive relationship with the results of other administrations of the test. The findings suggest that the Computational Thinking Scale shows promise in being a useful tool to assess the level and gains in computational thinking skills (Korkmaz & Bai, 2019).

A recently developed assessment method called the “Fairy Assessment” has been created to assist teachers in measuring growth in computational thinking. The development team created a computational thinking assessment for middle school students that analyzed the variation in assessment results among students. The Fairy Assessment was used to compare individual versus pair programming. The research also looked at correlation of gender results. The test measured a student’s ability to think algorithmically and use abstraction and modeling. The “Fairy Assessment” seemed to show promise as a way of assessing computational thinking skills

in middle school students. The test is specific to teachers using a specific coding language, Alice and will not work for other environments or computational thinking strategies (Werner, Denner, Campe, & Kawamoto, 2012; Lockwood & Mooney, 2018).

The University of Montana used the Whimbey Analytical Skills Inventory (WASI) in a pre-, post-test configuration for analysis during a new computational thinking course. The test covers computational thinking skills such as algorithm development and usage along with use of logic. The inventory was given to a wide range of students outside of the particular subject group to account for extraneous factors such as learning from other classes (Van Dyne & Braun, 2014). In another course study, a separate test, the Real Time Evaluation and Assessment of computational Thinking was used. This tool has shown promise in evaluating the level of usage of computational thinking patterns by students when solving a given problem (Lockwood & Mooney, 2018). Both assessments showed promising results in testing but need more study to determine the effectiveness of evaluating growth in computational thinking.

SRI International has been developing a more easily administered assessment, but the current versions as of the time of this study were targeted toward elementary students (Grover, Pea, & Cooper, 2015). The assessment has a template for teachers to design an assessment that requires specific problem-solving methodologies and steps. These design patterns have the potential to afford teachers the opportunity to measure the computational thinking abilities of students and therefore, assess their growth in computational thinking ability. (Bienkowski, Snow, Rutstein, & Grover, 2015)

One of the issues that makes creating assessments difficult for teachers is determining if computational thinking is defined as a skill or as knowledge. Assessing knowledge is more effectively done differently than skills. Furthermore, there are disagreements on the way to best

assess the transfer of computational thinking learning to other activities and content areas. Again, the assessment depends on the definition of computational thinking. The way teachers assess a transfer of knowledge is very different than the way they assess the transfer of a skill. When transferring a computational thinking skill to a new subject, students will often use different techniques, which adds to the assessment complexity (Burke, Bailey, & Ruiz, 2019). It is easier for teachers to assess specific knowledge than to assess a student's use of skills and problem-solving strategies. Having a methodology for designing an assessment that affords teachers the ability to measure the thought processes used in solving the problem will be more helpful in assessing computational thinking skills (Bienkowski, Snow, Rutsein, & Grover, 2015).

Computational thinking activities that included both an unplugged and a computer activity had mixed results in the outcomes. In general, high school students tended to rush through the unplugged activities to get started on the computer activities sooner. This tended to result in lower levels of computational thinking implementation. In activities that required minimum standards of achievement before progressing to the computer, students usually had better results (Fronza, Ioini, & Corral, 2016).

Conclusion

While the body research into the implementation of confrontational thinking in classrooms is relatively small, there is much that can be learned from it. The research that has been done can give some direction to teachers in ways to assess computational thinking in their classroom. As the same time, teachers can get some information on situations to avoid. Most of the studies and research analyzed utilized either student conferences to discuss the problem-solving methods used or a written set of steps that was then compared to an "expert" solution.

Some of the studies utilized both. Regardless of the method used, it was widely reported that these methods gave the most reliable assessment of student ability and growth. The same studies also acknowledged the greater amount of time that these methods took to complete the assessments.

Several groups and organizations are working to come up with an assessment method for teachers to use in the classroom that reduce the amount of time required to assess yet at the same time maintain reliability. In the meantime, it seems that teachers looking for ways to incorporate computational thinking into their classroom would be best suited incorporating it into another existing unit that spans several weeks or more. In addition, the teachers are more likely to get better results by utilizing one of the two assessment methods mentioned previously. This process has shown to give good assessment reliability to teachers and to minimize additional time needed for assessment as it blends two assessments into one. The research to date is not definitive, but it does help teachers in moving toward implementing computational thinking into their classroom and being able to measure the gains students have in problem-solving skills.

References

- Allen, V., Barr, V., Brylow, D., & Hambruch, S. (2010, January). Computational thinking in high school courses. *Proceedings of the 41st ACM technical symposium on Computer science education*, 390-391. DOI: 10.1145/1734263.1734395
- Ary, D., Jacobs, L., Razavieh, A., & Sorensen, C. (2009). *Introduction to research in education*. Wadsworth.
- Basawapatna, A., Koh, K., Repenning, A., Webb, D., & Marshall, K. (2011). Recognizing computational thinking patterns. *Proceedings of the 42nd ACM technical symposium on computer science education* (pp. 245-250). New York: ACM.
- Basu, S., Kinnebrew, J., & Biswas, G. (2014). Assessing student performance in a computational-thinking based science learning environment. *International Conference on Intelligent Tutoring Systems* (pp. 476-481). Springer. https://doi.org/10.1007/978-3-319-07221-0_59
- Bienkowski, M., Snow, E., Rutsein, D., & Grover, S. (2015). *Assessment design patterns for computational thinking practices in secondary computer science: A first look*. Menlo Park, CA: SRI International. Retrieved from <http://pact.sri.com/resources.html>
- Burke, Q., Bailey, C., & Ruiz, P. (2019). *Assessing computational thinking*. Retrieved from Center for Innovative Research in Cyber Learning: Retrieved from <https://circlcenter.org/assessing-computational-thinking/>
- Fronza, I., Ioini, N., & Corral, L. (2016). Teaching software design engineering across the k-12 curriculum: Using visual thinking and computational thinking. *Proceedings of the 17th Annual Conference on Information Technology Education*. Boston: ACM. DOI: 10.1145/2978192.2978220
- Grover, S., & Pea, R. (2013, March). Using a discourse-intensive pedagogy and android's app inventor for introducing computational concepts to middle school students. *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 723-728). Denver: Association for Computing Machinery. DOI: 10.1145/2445196.2445404
- Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2), 199-237. <http://dx.doi.org/10.1080/08993408.2015.1033142>
- Kalelioglu, F., Gulbahar, Y., & Kukul, V. (2016, May). A framework for computational thinking based on a systematic research review. *Modern Computing*, 4(3), 583-596. Retrieved from https://www.researchgate.net/publication/303943002_A_Framework_for_Computational_Thinking_Based_on_a_Systematic_Research_Review

- Korkmaz, O., & Bai, X. (2019, June). Adapting computational thinking scale (cts) for Chinese high school students and their thinking scale skills level. *Participatory Educational Research*, 6(1), 10-26. <http://dx.doi.org/10.17275/per.19.2.6.1>
- Korucu, A., Gencturk, A., & Gundogdu, M. (2017). Examination of the computational thinking skills of students. *Journal of Learning and Teaching in Digital Age*, 2(1), 11-19. Retrieved from <http://joltida.org/index.php/joltida/article/view/25>
- Lockwood, J., & Mooney, A. (2018, Jan). Computational thinking in secondary education: Where does it fit? A systematic literary review. *International Journal of Computer Science Education in Schools*, 2(1). DOI: 10.21585/ijcses.v2i1.26
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2010). Learning computer science concepts with scratch. *Proceedings of the sixth international workshop on computing education research* (pp. 69-76). New York: ACM.
- Sherman, M., & Martin, F. (2015, June). Assessment of mobile computational thinking. *Journal of Computing Sciences in Colleges*, 30(6), 53-59.
- Van Dyne, M., & Braun, J. (2014). Effectiveness of a computational thinking (CS0) course on student analytical skill. *Proceedings of the 45th ACM technical symposium on Computer science education*. Atlanta: ACM. DOI: 10.1145/2538862.2538956
- Werner, L., Denner, J., Campe, S., & Kawamoto, D. (2012). The fairy performance assessment: Measuring computational thinking in middle school. *Proceedings of the 43rd ACM technical symposium on computer science education* (pp. 215-220). ACM. DOI: 10.1145/2157136.2157200
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. (2014, March). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, 14(1). DOI:<http://dx.doi.org/10.1145/2576872>