Introducing the Computer Science Concept of Variables in Middle School Science Classrooms

Philip Sheridan Buffum
psbuffum@ncsu.edu[1]

Kimberly Michelle Ying
kimberlymprice@ufl.edu[2]

Xiaoxi Zheng
xzheng1995@gmail.com[2]

Kristy Elizabeth Boyer
keboyer@ufl.edu[2]

Eric N. Wiebe
wiebe@ncsu.edu[3]

Bradford W. Mott
bwmott@ncsu.edu[1]

David C. Blackburn
dblackburn@flmnh.ufl.org[4]

James C. Lester
lester@ncsu.edu[1]

[1]Department of Computer Science, North Carolina State University, Raleigh, North Carolina
[2]Department of Computer & Information Science & Engineering, University of Florida, Gainesville, Florida
[3]Department of STEM Education, North Carolina State University, Raleigh, North Carolina
[4]Department of Biology, University of Florida, Gainesville, Florida

ABSTRACT

The K-12 Computer Science Framework has established that students should be learning about the computer science concept of variables as early as middle school, although the field has not yet determined how this and other related concepts should be introduced. Secondary school computer science curricula such as Exploring CS and AP CS Principles often teach the concept of variables in the context of algebra, which most students have already encountered in their mathematics courses. However, when strategizing how to introduce the concept at the middle school level, we confront the reality that many middle schoolers have not yet learned algebra. With that challenge in mind, this position paper makes a case for introducing the concept of variables in the context of middle school *science*. In addition to an analysis of existing curricula, the paper includes discussion of a day-long pilot study and the consequent teacher feedback that further supports the approach. The CS For All initiative has increased interest in bringing computer science to middle school classrooms; this paper makes an argument for doing so in a way that can benefit students' learning of both computer science and core science content.

CCS CONCEPTS

• **Social and professional topics → Computing education**; *Computational thinking*; K-12 education

KEYWORDS

Middle school; Computational Thinking; Science classrooms

SIGCSE '18, February 21-24, 2018, Baltimore, MD, USA

ACM ISBN 978-1-4503-5103-4/18/02…$15.00
https://doi.org/10.1145/3159450.3159545

1 INTRODUCTION

Over the past decade, the computer science education community has placed an increasing emphasis on computer science in K-12, culminating in the White House's announcement of the Computer Science for All initiative in January 2016. Curriculum frameworks and interventions have reached millions of students, contributing to significant momentum. We are beginning to see longitudinal studies of K-12 students' computing attitudes and self-efficacy [1], design-based research examining how well a block-based programming curriculum prepares students for later text-based programming [15], and empirical studies to investigate the factors that affect middle school students' learning of algorithmic thinking [16]. Most notably, the CS For All initiative has energized efforts to prepare in-service teachers to teach computer science in their classes [23] as well as efforts to introduce middle school computer science curricula, such as by Code.org [9].

Yet the present urgency to deliver computer science to K-12 classrooms—for all its benefits—threatens to brush aside deep reflection on individual concepts, analogous to the kind of long-term reflection that has benefited other fields such as mathematics education. Because the field of computer science is relatively young, a single computer science concept might provide researchers with years of fruitful work: examining how to teach it, understanding the optimal stage in students' education when they should learn it, and identifying common misconceptions that students might acquire in the process of learning it. Computer science education researchers must take dispassionate and measured approaches to examining how we might best teach core computer science concepts at the middle grade level, even as we maintain the passion and urgency to the broader goal of bringing computer science to all children.

Emerging research has begun to pursue this goal. Researchers at the University of Chicago have identified concepts that theoretical work has considered important but that classroom research has not yet investigated [24]. They have furthermore begun developing learning trajectories for individual concepts in elementary school [11]. They have already presented trajectories for three concepts (Sequence, Repetition, and Conditionals) and have highlighted the importance of considering the trajectories of other disciplines (specifically mathematics) as we make curricular decisions for computer science education [25].

This position paper discusses the context and issues surrounding the ways in which students are introduced to the concept of *variables* in middle school classrooms. Throughout this paper, the reader should note that each of the three disciplines discussed (computer science, mathematics, and science) has a distinct conceptual understanding of the term 'variable', yet these different conceptualizations share a certain degree of compatibility. We will illustrate through a review of existing curricula (Section 3) that many middle school computer science curricula currently introduce the computer science concept of variables in the context of students' mathematics education. We make no negative claim about contextualizing computer science concepts with mathematics; yet we consider it healthy to question that particular approach's ubiquity. Moreover, we speculate that an interdisciplinary approach—involving both mathematics and science education—might hold greater potential for pushing forward the vision of CS For All. Indeed, this position paper will make the argument for introducing the computer science concept of variables to middle school students within the context of science classes.

2 RELATED WORK

Over the past decade, the computer science education community has shown increasing interest in researching how to develop computer science curricula at the middle grades level. Researchers have developed stand-alone computer science courses at the middle school level using a "braided teaching" approach [21] or by leveraging existing tools such as Scratch [28] and Alice [26]. More work needs to be done to investigate whether relying on these popular tools privilege students who have already used them in prior computer science activities, and whether teachers who are unfamiliar with computer science will feel empowered to fully adopt the curriculum [20]. In addition, while block-based programming can make computer science more accessible to novices by removing some syntactic challenges, students will still struggle to learn important but abstract computer science concepts, such as variables, unless they also have a strong curriculum supporting such concepts to go along with the block-based programming environment [14].

As an alternative to developing stand-alone computer science courses at the middle grades level, other research has looked at integrating computational thinking into other subjects. Science classrooms have seen the introduction of computational thinking in the form of modeling and simulation activities [29]. More recent research, informed by interviews with mathematicians and scientists, has proposed three additional categories of computational thinking practices for science and mathematics classrooms (along with modeling and simulation): data, computational problem solving, and systems thinking [31]. In contrast to existing research, this paper takes particular interest in how students might learn about the computer science concept of variables. Although some of the aforementioned modeling and simulation activities may have provided opportunities for students to use variables in their block-based programs [4], prior research has not explored how to teach the specific concept of variables in the context of middle school science classrooms. The next section will examine how current curricula address the concept of variables.

3 REVIEW OF EXISTING CURRICULA

3.1 K-12 Computer Science Curricula

Historically, K-12 computer science education stemmed out of undergraduate computer science programs, motivated by the need to prepare pre-college students for the rigor of an undergraduate computer science curriculum [3]. Early K-12 computer science curricula thus emerged from curriculum developers adapting undergraduate introductory courses to the secondary school level [2]. Given the strong ties between computer science and mathematics at the undergraduate level, it is unsurprising that many secondary school computer science curricula also tend to include connections to mathematics. The application of certain research-based methods, such as a spiral curriculum [5] and learning trajectories [8], may have led to an inevitability that many of these K-8 computer science curricula should also make similar references to students' mathematics knowledge.

In the United States, secondary school students might encounter the computer science concept of variables in one of several different computer science courses. In some cases, such as Exploring Computer Science, the connection to math is in the forefront of established lesson plans [13]. In other cases, such as the CollegeBoard's two Advanced Placement (AP) courses for computer science, individual teachers might offer varied implementations of the given course but the existing teacher resources tend to encourage connecting the concept of variables to mathematics. CollegeBoard's official Teacher's Guide includes strategies for teaching conceptual understanding of variables using arithmetic examples for the AP Computer Science A course [7], as does Code.org for its online implementation of the AP Computer Science Principles course [10].

As with secondary school computer science curricula, emerging middle school computer science curricula in the United States often emphasize an interdisciplinary connection to mathematics. One large-scale effort, Bootstrap, integrates computer science concepts with algebra, reinforcing the content of eighth-grade math curricula for middle school students who have reached that level in their math education [27]. Code.org has released a middle school course that shares similarities with its AP CS Principles course; instead of contextualizing variables within algebra content that many middle school students do not yet have, though, it does so with students' prior geometry knowledge (x- and y-coordinates) [9]. Another project has integrated computer science with the math education of younger students, including the computer science concepts of sequencing, repetition and conditionals [25].

With the CS for All initiative energizing efforts to develop rigorous computer science learning at the middle school level, many more curricula may emerge in the coming years. *The K-12 Computer Science Framework* may guide much of that work. The framework includes *Variables* as a sub-concept under the broader concept of *Algorithms and Programming* [17]. Following a spiral curriculum model [5], it lays out a vision in which children first encounter variables as early as kindergarten, and then revisit the concept with increasing formality and abstraction as their education progresses. Before they enter sixth grade, they should (ideally) know that programming languages use variables to work with data, and that variables can have different types (e.g., numeric versus character). As illustrated in Table 1, they can then develop a more sophisticated understanding of the concept as they go through the middle grades.

Table 1: K-12 Computer Science Framework statement on Variables sub-concept for the 6-8 grade band

6-8	Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.

In its description of *Variables* for the 6-8 grade band, the framework notes that students should deepen their understanding of the concept at this stage. Students should learn that "the term variable is used differently in programming than the way it is commonly used in mathematics" [17]. [1] Piaget believed children enter the stage of formal operations in their cognitive development at this age (i.e., around age 12) [22], so there is a sound basis for prompting interdisciplinary connections in the 6-8 grade band. Yet students' success at making interdisciplinary connections will depend on their progress in those other disciplines. We must therefore examine the standards of other disciplines.

3.2 Middle School Math and Science Curricula

In the United States, forty-two states and the District of Columbia have (at least partly) adopted the Common Core State Standards for Mathematical Practice [30]. The standards expect students to encounter the mathematical concept of variables in each of the three middle school grades (6th, 7th and 8th). In sixth grade, for example, standard 6.EE.C.9 expects students to be able to "Recognize and evaluate quantitative relationships between independent and dependent variables." A seventh grade standard, 7.EE.B.4, expects students to be able to "Use variables to represent quantities in a real-world or mathematical problem, and construct simple equations and inequalities to solve problems by reasoning about the quantities." An example of an eighth grade standard involving variables is 8.SPA.2: "Know that straight lines are widely used to model relationships between two quantitative variables." We can thus find ample opportunity for teaching the computer science concept of variables in any of these middle grades while also addressing at least one math standard.

Yet mathematics is not the lone subject in which students might encounter the term "variable" in middle school. The Next Generation Science Standards (NGSS) framework considers both computational and mathematical thinking as central to K-12 science education [18]. The framework envisions three dimensions of learning: practices, cross-cutting concepts, and discipline core ideas [18]. In NGSS, each standard emerges as a combination of all three dimensions.

Table 2: The Concept of Variables in the Next Generation Science Standards for Grades 6-8

Practice 3. Planning and Carrying Out Investigations
"…identifying the relevant variables and considering how they might be observed, measured, and controlled…"

Practice 4. Analyzing and Interpreting Data
"…expressing relationships between different variables in the data set…"

Practice 5. Using Mathematics and Computational Thinking
"…to test relationships between variables, and to explore the interplay of diverse external conditions…"

Practice 6. Constructing Explanations and Designing Solutions
"…explain observed relationships between variables and describe the mechanisms that support cause and effect inferences about them…"

Cross-cutting concept: Scale, Proportion, and Quantity
"…students should be able to apply [algebraic thinking] to examine their scientific data to predict the effect of a change in one variable on another…"

Cross-cutting concept: Systems and System Models
"…students' models should incorporate a range of mathematical relationships among variables…and some analysis of the patterns of those relationships."

As Table 3 illustrates, the science concept of variables (which is aligned with Common Core mathematics) appears in multiple practices and cross-cutting concepts. More concretely, several middle school standards for life science involve the concept of variables [19]. MS-LS1-1, for example, expects students to be able to conduct an investigation into cell structures. The performance expectation for this standard incorporates Practice 3 (Planning and Carrying Out Investigations) and the cross-cutting concept of Scale, Proportion, and Quantity. It also references a connection to a Common Core math standard (6.EE.C.9) on independent and dependent variables. MS-LS2-3, another middle school life science standard that makes a connection to the 6.EE.C.9 math standard, expects students to be able to model the interactions, energy and dynamics of ecosystems. The next section describes a prototype activity that we designed around this standard and piloted in middle school science classrooms.

[1] For specific differences and how students' familiarity with the mathematical concept leads to misunderstandings about the computer science concept, see [14].

4 PROTOTYPE ACTIVITY WITH VARIABLES: POPULATION CALCULATOR

4.1 Design of Activity

Seeking to integrate computer science concepts into middle school science classrooms, we noted the specific opportunity to address the concept of variables within the context of the existing science curriculum. We thus designed an activity that called for students to use block-based programming to create a "population calculator" based on initial populations and growth rates of bats and mosquitoes. The task reinforces science learning objectives regarding ecosystems. Although we designed the activity for a state that has not yet adopted the Next Generation Science Standards (and consequently used that state's own science standards as the primary curricular scaffold for the activity), the activity supports NGSS standard MS-LS2-3.

The task required students to use at least one variable and challenged them to use as few variables as possible in completing the program. It is possible to create the program using only a single variable for population (rather than a separate population variable for each of the two species). Discovering how this is possible addresses a central learning objective from the K-12 Computer Science Framework for grades 6-8: "Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs" [17].

We customized Blockly to develop a programming interface for this activity that included only the blocks relevant to the problem solution. Figure 1 shows a screenshot of a solution using a single variable for population, which is generalizable to any species. To be clear, the program in the screenshot is not intended as an exemplar (the reader may note that the program doubles the population after 20 years for any species input, ignoring the fact that growth rates differ by species).

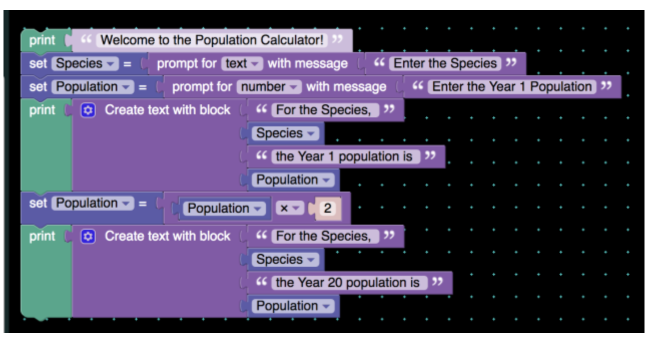

Figure 1: Screenshot of the block-based program for the Population Calculator

4.2 Piloting of Activity

During Spring 2017, we piloted the population calculator learning activity in a middle school in the southeastern United States. The demographics of the school are as follows: of 700 students, there were 53% female and 47% male, 68% White/Caucasian, 18% Black/African-American, 8% Hispanic/Latino, 1% Asian/Pacific Islander and 7% Other (including multiracial and some races not listed in the above categories). Our primary research question for the pilot study was, "How can students come to understand the computer science concept of variables in the context of middle school science classrooms?"

This pilot study was conducted on a single day with the classes of a seventh grade science teacher. This teacher taught five class periods, each 50 minutes long and with approximately 20 students. In each period, we began with a full class discussion about what students have heard about variables. While the rest of the researchers took detailed notes, one of the researchers led the discussion with prompts such as "What have you all heard about variables?" and "Where have you heard the term 'variable'?" Following the class discussion and a short demo of the programming interface, students had approximately 30 minutes to develop their programs. They each had individual laptops, but were allowed to collaborate as much as they liked. The researchers took notes, answered student questions and asked students to explain their thought processes at key moments. The classroom teacher monitored students' progress, intervening at her discretion. Following the last class period, the research team interviewed the teacher to gain her rich feedback on the activity.

5 CLASSROOM OBSERVATIONS

During the full-class discussions in each of the five periods, certain students responded immediately by noting that they were familiar with the concept of a variable from their math class. However, with these seventh graders enrolled in varying levels of math, other students (who remained quiet) may not have encountered variables before. When pressed further about what a variable is in mathematics, students responded with definitions such as, "they're the letters in math problems", "like a number you don't know" and "a letter that can stand in the place of a number." Only later in each discussion did students note that they have heard of variables in their science class, too (and sometimes this came after the classroom teacher prompted them to remember that fact). In the science context, students noted that "there are dependent and independent variables" and that a variable can be "an unknown number or factor in a science experiment."

The researchers observed that students in all classes were highly engaged with the block-based programming activity and nearly all completed it. Students made connections to other types of growth rates, such as human populations in their hometown. Many students, however, were reluctant to consolidate the number of different variables in their programs. This led to productive conversations about the nature and purpose of variables. Some students also struggled with the understanding of how and why to initialize a given variable. Another common misconception was students conflating the variable itself with the variable's identifier (e.g., expecting that printing a variable named "population" would output the string "population" rather than a numeric value). These misconceptions are consistent with other research studies that have investigated how students learn the computer science concept of variables [14]. As students worked through their misconceptions in the context of this science activity, they had the opportunity to make a distinction between the computer science concept of variables and the scientific concept: namely that a program's variable can represent a scientific variable while presenting additional considerations (*e.g.*, initialization).

6 SCIENCE TEACHER FEEDBACK

Following the last period, the research team interviewed the classroom teacher to get her thoughts on the activity overall and how it fits with her science curriculum. She noted that all the students should know about dependent and independent variables from their science class, but some students seemed to have forgotten it. She covered the concept in a Practice of Science unit in the early Fall. Considering that our classroom activity occurred in late Spring (at least seven months later) and also that students struggle with the Practice of Science unit due to it being the most abstract, it did not surprise the teacher that many students struggled to recall discussing variables in their science class. The teacher consequently liked the potential of inserting this computer science activity in that Practice of Science unit to make it both more engaging and less abstract, thus increasing the likelihood that students will achieve lasting mastery of the science concept.

Going further, the teacher explained that the whole nature of variables and science is especially difficult for middle school students to understand. Although they conduct many different experiments throughout the year, and often hear "how does [blank] affect [blank]", they have a hard time concretely understanding that this is referring to variables: "the vocabulary is the ongoing battle." That challenge arguably becomes exacerbated by students hearing the same term in other contexts (e.g., math class) without learning how it is connected to its use in science.

Finally, the teacher provided the research team with a better understanding of the student population and specifically their math backgrounds. In this particular school, some of the students are in a magnet program and tend to come from families with more resources than the mainstream students. There exists a stark contrast in the math levels between magnet students and their peers. In her seventh-grade science classes, she estimated that two-thirds of the magnet students are currently taking algebra, with the other third in pre-algebra. Meanwhile, she could think of only one of her mainstream students who was in algebra, a handful in pre-algebra, and most in the "on level" math course. Given the significant differences in math experience, teaching the computer science concept of variables in the context of science might achieve more equitable outcomes than teaching the concept in math classes alone.

7 DISCUSSION

The question of equity drives our inquiry in this matter, as it does the larger CS For All movement. In the past, few students in the United States were graduating high school with any background in computer science because so much of pre-college computer science education occurred in out-of-school activities, experienced by a privileged and self-selecting minority. By bringing computer science to middle school classrooms, we can reach a broader population of students [6], yet we must still examine whether our in-school curricular choices are sufficiently addressing the entrenched inequities that linger.

This paper argues that contextualizing computer science concepts (in this case, variables) within a rich life science activity may engage a broader population of students in thinking about the computer science. While some middle school students might embrace the more abstract consideration of variables that can come from learning the concept in a math class, our classroom observations and teacher feedback reinforce our sense that a more concrete, "real-world" connection can greatly benefit most students' learning. Indeed, this sort of integration on the concept of variables can benefit students' learning of both computer science and life science. We furthermore argue for developing lessons that specifically teach the concept of variables, rather than relying only on modeling and simulation activities in which students might encounter block-based programs with variables.[2] As recent research has clarified, students do not gain a true conceptual understanding of variables merely through exposure to variables in block-based programming activities [14].

We mentioned in Section 1 that it is healthy to question what is the root cause for the near consensus that computer science concepts such as variables should be contextualized in mathematics as early even as middle school. There are many possible explanations, but the answer might partly be a trickle-down effect, in which undergraduate computer science has certain curricula, then high school curricula seek to prepare students for the undergraduate experience, and then middle school curricula aim to prepare students for the high school experience. It might also partly be due to mathematics just seeming like the most natural fit (perhaps because researchers who have gone through undergraduate computer science programs strongly associate computer science with mathematics). In either case, it is important to consider the varying math levels of students. Coupling computer science exclusively with mathematics at such an early stage as middle school might increase the risk of tracking students in computer science before they even reach high school.

In short, we pose the question: Might an over-emphasis on the math connection widen the gap between high-achieving students (who might generally be more predisposed to study computer science) and their peers? Someone could counter by noting that undergraduate computer science programs are mathematically rigorous and therefore we must start preparing students early on for learning computer science with a similarly mathematics-centered approach. Yet, we must remember that CS For All does not imply that all K-12 students will go on to major in a traditional computer science program at a traditional 4-year college. Imagining what certain aspects of middle school computer science might look like thus might call for broadening our preconceived expectations.

[2] This is not to discount the importance of those sorts of activities in general.

8 CONCLUSION

We envision middle school students learning core computer science concepts, like variables, in such a way that they feel their futures are open. It is true that we must hold high expectations and prepare students for potential success in more advanced computer science courses. Students should indeed draw connections to computer science concepts in their math classes. Yet that should not be the only place we teach core computer science concepts. This paper has endeavored to point out an opportunity to teach one specific computer science concept, variables, in the context of middle school science where students are likely to encounter the term 'variable'. Helping students connect that science term to the computer science and mathematical concepts of the same name represents not only an opportunity, but perhaps a necessity.

This position paper sees a research area rich with future work. We are at only the beginning of our understanding of how we can integrate the computer science concept of variables with science curricula. We still need greater understanding of how middle school students understand the scientific concept of variables in relation to the computer science concept, and how we can build interdisciplinary lessons that lead to rich science and computer science learning. We furthermore note that several other computer science concepts might also integrate well with middle school science curricula. Finally, while this paper has focused on the potential of integrating this computer science concept with middle school science, we should not limit discussion on interdisciplinary integration to only STEM subjects. For example, another opportunity to introduce the computer science concept of variables might exist in social studies curricula, which often expect students to analyze cause and effect in various sociohistorical contexts [12].

With a comprehensive research approach that incorporates both a narrowing focus (i.e., in-depth investigations into how to teach an individual computer science concept) and a broadening focus (i.e., open consideration of all opportunities for where to introduce that computer science concept), we can find greater success at bringing CS For All to the middle grade level.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation through Grants CNS-1138497 and DRL-1640141. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

REFERENCES

[1] Aritajati, C., Rosson, M.B., Pena, J., Cinque, D. and Segura, A. 2015. A Socio-Cognitive Analysis of Summer Camp Outcomes and Experiences. *Proceedings of SIGCSE '15*, 581–586.

[2] Association for Computing Machinery 2004. *A Model Curriculum for K-12 Computer Science : Final Report of the ACM K-12 Task Force Curriculum Committee*. ACM.

[3] Association for Computing Machinery 1993. ACM model high school computer science curriculum. *Association for Computing Machinery. Communications of the ACM*. 36, 5 (1993), 87.

[4] Basu, S., Kinnebrew, J.S. and Biswas, G. 2014. Assessing student performance in a computational-thinking based science learning environment. *Intelligent Tutoring Systems. ITS 2014. Lecture Notes in Computer Science* (2014), 476–481.

[5] Bruner, J. 1960. *The Process of Education*. Harvard University Press.

[6] Buffum, P.S., Frankosky, M.H., Boyer, K.E., Wiebe, E.N., Mott, B.W. and Lester, J.C. 2016. Empowering All Students: Closing the CS Confidence Gap with an In-School Initiative for Middle School Students. *Proceedings of SIGCSE '16*, 382–387.

[7] Carter, D.P. Teacher's Guide: AP Computer Science. *CollegeBoard Advanced Placement Programming*.

[8] Clements, D.H. and Sarama, J. 2004. Learning Trajectories in Mathematics Education. *Mathematical Thinking and Learning*. 6, 2 (2004), 81–89.

[9] Code.org | Computer Science Discoveries: *https://studio.code.org/courses/csd*. Accessed: 2017-08-28.

[10] Code.org | Computer Science Principles: *https://code.org/educate/csp*. Accessed: 2017-08-28.

[11] Dwyer, H.A., Hill, C., Carpenter, S., Harlow, D. and Franklin, D. 2014. Identifying Elementary Students' Pre-Instructional Ability to Develop Algorithms and Step-by-Step Instructions. *Proceedings of SIGCSE '14*, 511–516.

[12] Farris, P.J. 2007. *Elementary and middle school social studies : an interdisciplinary, multicultural approach*. Waveland Press.

[13] Goode, J. and Margolis, J. 2011. Exploring Computer Science. *ACM Transactions on Computing Education*. 11, 2 (Jul. 2011), 1–16.

[14] Grover, S. and Basu, S. 2017. Measuring Student Learning in Introductory Block-Based Programming. *Proceedings of SIGCSE '17*, 267–272.

[15] Grover, S., Pea, R.D. and Cooper, S. 2014. Expansive Framing and Preparation for Future Learning in Middle-School Computer Science. *International Conference of the Learning Sciences (ICLS) Conference* (2014), 992–996.

[16] Grover, S., Pea, R.D. and Cooper, S. 2016. Factors Influencing Computer Science Learning in Middle School. *Proceedings of SIGCSE '16*, 552–557.

[17] k12cs.org – K–12 Computer Science Framework: *https://k12cs.org/*. Accessed: 2017-08-28.

[18] National Research Council Press 2011. *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. The National Academies.

[19] Next Generation Science Standards: *https://www.nextgenscience.org/*. Accessed: 2017-08-28.

[20] Ni, L. 2009. What Makes CS Teachers Change?: Factors Influencing CS Teachers' Adoption of Curriculum Innovations. *ACM SIGCSE Bulletin*. 41, 1 (Mar. 2009), 544–549.

[21] Pasternak, A. and Vahrenhold, J. 2012. Design and Evaluation of a Braided Teaching Course in Sixth Grade Computer Science Education. *Proceedings of SIGCSE '12*, 45–50.

[22] Piaget, J. 1972. Intellectual Evolution from Adolescence to Adulthood. *Human Development*. 15, 1 (1972), 1–12.

[23] Pollock, L., Mouza, C., Czik, A., Little, A., Coffey, D. and Buttram, J. 2017. From Professional Development to the Classroom: Findings from CS K-12 Teachers. *Proceedings of SIGCSE '17*, 477–482.

[24] Rich, K., Strickland, C. and Franklin, D. 2017. A Literature Review through the Lens of Computer Science Learning Goals Theorized and Explored in Research. *Proceedings of SIGCSE '17*, 495–500.

[25] Rich, K.M., Strickland, C., Binkowski, T.A., Moran, C. and Franklin, D. 2017. K-8 Learning Trajectories Derived from Research Literature. *Proceedings of the 2017 ACM Conference on International Computing Education Research - ICER '17*, 182–190.

[26] Rodger, S.H., Dalis, M., Gadwal, C., Hayes, J., Li, P., Wolfe, F., Zhang, W. and Liang, L. 2012. Integrating Computing into Middle School Disciplines through Projects. *Proceedings of SIGCSE '12*, 421–426.

[27] Schanzer, E., Fisler, K., Krishnamurthi, S. and Felleisen, M. 2015. Transferring Skills at Solving Word Problems from Computing to Algebra Through Bootstrap. *Proceedings of SIGCSE '15*, 616–621.

[28] Schofield, E., Erlinger, M. and Dodds, Z. 2014. MyCS: CS for Middle-Years Students and their Teachers. *Proceedings of SIGCSE '14*, 337–342.

[29] Sengupta, P., Kinnebrew, J.S., Basu, S., Biswas, G. and Clark, D. 2013. Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*. 18, 2 (Jun. 2013), 351–380.

[30] Standards for Mathematical Practice | Common Core State Standards Initiative: *http://www.corestandards.org/Math/Practice/*. Accessed: 2017-08-28.

[31] Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. and Wilensky, U. 2016. Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*. 25, 1 (Feb. 2016), 127–147.