DOI: 10.34641/ctestem.2022.475

Design of an Evaluative Rubric for CT Integrated Curriculum in the Elementary Grades

Florence R. Sullivan, Lian Duan, Emrah Pektas
University of Massachusetts Amherst
florence@umass.edu, lduan@umass.edu, epektas@umass.edu

ABSTRACT
Despite the recent proliferation of research concerning integrating computational thinking (CT) into K-5th grade curriculum, there is little literature concerning how to evaluate the quality of CT integrated curricula, especially curricula integrating CT into language arts and social studies content areas. In this paper, we present a theoretically derived rubric for the evaluation of CT integrated curricula for grades K-5 across the curriculum (math, science, language arts, social studies). Our rubric is divided into two sections. The first section provides guidelines for identifying the integration type (disciplinary, multidisciplinary, interdisciplinary, or transdisciplinary). The second section presents six categories of evaluation that further subsume nine sub-categories. The principal categories of evaluation include the following: conceptual coherence, role of computational technology, assessment, use of multiple representations, play, and equity. We include the play category as an aspect of developmental appropriateness. Play is an important pedagogical approach for learning in the early grades. Our work takes place in the context of the Computer Science (CS) for All initiative in the United States which emphasizes the goal of improving racial and gender diversity in CS participation. Therefore, creating integrated lessons that address equity is important. Our paper describes rubric development from the theoretical perspectives that underlie the inclusion of each type, category, and sub-category. Our evaluative rubric can guide future efforts to integrate CT/CS into the elementary curricula. Researchers can utilize our rubric to evaluate and analyze CT-integrated curricula, and educators can benefit from using this rubric as a guideline for curriculum development.

KEYWORDS
Curriculum integration, computational thinking, elementary grades, evaluation rubric,

1. ELEMENTARY CT INTEGRATION
While introducing computational thinking (CT) in the elementary grades is not a particularly new idea (Bers, Flannery, Kazakoff, & Sullivan, 2014; Papert, 1981), it is, arguably, an increasingly important one. Many professions require facility with computers (Muro, Liu, Whiton & Kulkarni, 2017), and indeed 95% of children in the United States have access to a computational device and the internet in their own homes (National Center for Educational Statistics, 2019). Teaching children CT in the early grades is warranted. While this is so, the elementary school day has a full curriculum that leaves little room for introducing a stand-alone new topic such as computer science (Sherwood, et al., 2021). To introduce CT in elementary school an integrated approach is needed. Indeed, advocates of integrated curriculum believe that utilizing naturally overlapping areas of disciplines to integrate curriculum leads to higher student engagement and consequently higher achievement (Drake & Burns, 2004; Hinde, 2005; Vars, 1991). However, curricular integration is not a simple task. It requires attention on many levels. Here we present our work on the development of a CT-integration evaluation rubric for elementary curricula. The evaluation rubric is theoretically grounded and builds on prior work. Our evaluation rubric is unique in that no other comprehensive, elementary level CT-integration evaluation rubric exists. This rubric can be used both to evaluate existing curricula, or as a guide to curriculum development.

2. CT INTEGRATION APPROACHES
In a review of the literature, we have identified three basic approaches to integrating CT in the K12 curriculum as follows: general conceptual or practice overlap (Dong, Cateté, Jocius, Lytle, Barnes, Albert, Joshi, Robinson, & Andrews, 2019; Settle, Frank, Hansen, Spaltro, Jurisson, Rennert-May, & Wildeman, 2012); specific conceptual or practice overlap (Clark & Sengupta, 2020; Israel & Lash, 2020; Sengupta, Kinnebrew, Basu, Biswas, & Clark, 2013; Weintrop, Behesti, Horn, Orton, Jona, Trouille, & Wilensky, 2016); and/or general content support (Waterman, Goldsmith & Pasquale, 2020). In terms of the first approach to integration, researchers have identified general practices and or concepts that are common to both computer science and other disciplines. For example, Dong et al., (2019) identified Pattern Recognition, Abstraction, Decomposition, and Algorithms (PRADA) as general concepts, that while foundational to computer science, are also found in many disciplines. The PRADA concepts can be used to approach problems in multiple fields. Similarly, Settle, et al. (2012), identified abstraction as a general concept, foundational to computer science work, and widespread among other disciplines.

The second approach to integration is to identify specific conceptual or practice overlaps between computational thinking ideas and other disciplines. This approach is typically focused on integrating CT into either math or science curricula. For example, both Clark and Sengupta (2020) and Sengupta, et al., (2013) identified modeling as a specific practice in science and computer science. Moreover, computational modeling is, at this point, an indispensable aspect of most scientific research. Israel and Lash (2020) identified three specific concepts in CT and math including sequencing, looping, and conditional logic. Meanwhile, Weintrop, et al., (2016) developed a comprehensive guide to the relationship of CT to the disciplines of math and science at the secondary level, including four overarching categories and 22 practices that are specific to both CT and math and science.

Finally, the third approach focuses on general content support. Waterman, Goldsmith and Pasquale (2020) pre-identified three forms of integration of CT into the science curriculum: exist, enhance, extend. Working with the third-grade science curriculum, these researchers identified science topics where CT naturally existed as part of the inquiry activity, places where CT could enhance the learning of the topic, and places where CT could extend the learning of the science topic.

These various approaches to CT integration are valuable for curriculum developers and teachers. However, they do not, in and of themselves, speak directly to the quality of a particular CT integrated curriculum. Therefore, we have worked to develop a CT integration evaluation rubric for the elementary grades. Our rubric addresses issues of quality, developmental appropriateness, and equity. In the balance of this paper, we describe our development process, and we provide the theoretical grounding for the presence of each category of evaluation. Our goal in undertaking this work is to furnish the CS education research community with a useful tool for making important curricular decisions regarding selecting or developing a high-quality CT-integrated curriculum for the elementary grades.

3. RUBRIC DEVELOPMENT PROCESS

Our rubric development process proceeds from the literature regarding curriculum integration. Some papers have focused on identifying different types of integration– which we discuss in section 4 below. Other papers have focused on, or identified, important elements of quality that should be considered when working to integrate two or more disciplines – which we discuss in section 5 below. These quality indicators include conceptual coherence, the role of technology, assessment, and the use of multiple representations. We have included two other quality indicators that we believe are important and which contribute to the comprehensiveness of our rubric: play and equity. We include a focus on play due to the importance of play as a pedagogical approach in the early grades (National Association for the Education of Young Children, 2020). We include a focus on equity because in our United States context, there is a strong focus on improving the diversity of individuals who participate in CS including those from societally oppressed racial groups. Therefore, developing CS curricula that addresses issues of equity is important.

4. TYPES OF INTEGRATION

Various approaches to curricular integration have been posited over the years (see Davison, Miller & Metheny, 1995; Fogarty, 1991; Vars, 1991). Common to the approaches is the goal of finding overlapping connections among disciplines, such that integration is sensible. Such integration might occur through various mechanisms of overlap, for example, content integration, thematic integration, process integration, skill integration and correlational integration (Davison, et al., 1995; Fogarty, 1991; Vars, 1991). More recently, researchers have developed an integration model that includes three approaches and implies a fourth. The three approaches first discussed by Drake and Burns (2004) and later elaborated upon by Vasquez, Comer & Sneider (2013), include the following multidisciplinary, interdisciplinary, and transdisciplinary. The fourth, implied aspect is disciplinary – and this element is a part of Vasquez, Comer & Sneider's (2013) delineation of the types of integration. We have adopted their approach for our rubric. Therefore, our rubric has four types of integration: disciplinary, multidisciplinary, interdisciplinary and transdisciplinary.

We include the disciplinary category as some approaches to integration acknowledge the importance of developing specific disciplinary knowledge prior to engaging in multi, inter, or transdisciplinary learning (Kiray, 2012). Essentially, a unit might include some disciplinary learning prior to introducing its connection to another discipline. The second type of integration, multidisciplinary, refers to lessons or units where two disciplines are united by a common theme, but where the goals of the lesson for each discipline are not interconnected or interdependent. An example of a multidisciplinary approach to integrating CT into the curriculum would be selecting a particular theme, such as "plants" and then teaching about plants (e.g., the plant life cycle) using computational media, for example, have students create an animation of the plant life cycle from seed to flower using Scratch. The third type of integration is interdisciplinary. In this approach the two disciplines are conceptually connected, in other words a concept, common to both disciplines is at the heart of the lesson; and the learning goals for each discipline are interconnected and interdependent. An example of integrating CT with this approach is to identify a concept, such as "precision." This concept is important in both computer science and in learning how to write procedurally in English Language Arts, for example, writing precise instructions. Finally, there is the transdisciplinary type of integration. In this approach, the focus is on approaching a real-world problem from multiple disciplinary lenses. An example of integrating CT using this approach would be to identify a community problem, for example the presence of large potholes in the streets, and then develop a plan for solving the problem from various lenses, including sociological (survey the community to discover thoughts about the problem), English Language Arts (write up the results of the survey) and computer science (create an application using GPS technology that allow people to automatically flag the location of a pothole). It is important to recognize that each of these types of integration (multidisciplinary, interdisciplinary, and transdisciplinary) are equally valid and equally useful. The approach selected should be driven by the overall goals of a given lesson or unit (Kiray, 2012).

5. QUALITY INDICATORS

Here in section 5, we will describe the quality indicators, including discussing their roots in the literature. We have developed a four-point qualitative evaluation system including the following assessments: poor, fair, good, excellent. These assessments use a graduated presence-absence evaluative approach. For example, if a quality indicator is judged as poor, it is judged so due to the wholesale absence of the indicator. The rest of the assessment levels move in a graduated way towards the full presence of the quality indicator. Due to space limitations, we are not able to provide a detailed description of each element of the four-point evaluative rubric for each quality indicator. However, we plan to publish the full rubric at a later date. Here we provide the theoretical underpinnings and rationale for the rubric.

5.1 Conceptual Coherence

The first quality indicator is conceptual coherence (Roehrig, Dare, Ring-Whalen & Wieselmann, 2021). Coherence can be achieved through scanning curricular standards to find synergistic integration points, assembling these points to ensure horizontal and vertical progression throughout the school year and across the grades, designing learning activities to achieve the learning objectives in the integrated subject areas, and aligning standards and learning goals and activities with assessments (Drake & Burns, 2004; Case, 1994). While conceptual coherence may be evaluated across varying timescales, for the purpose of this evaluative rubric, coherence concerns the relationship among concepts introduced in a lesson. For example, how are the concepts sequenced and linked to one another? How do the concepts work together to build a picture of the topic of interest? How interrelated are the concepts?

We evaluate this indicator on two levels: (1) the coherence of CT concepts throughout the lesson; and (2) the coherence of the CT concepts with the target domain concepts in the lesson. In terms of our rubric, CT concept coherence refers to CT concepts being introduced in a clear, meaningful order. For example, in a lesson that introduces the CT concepts of algorithms and debugging, we would expect to see the concept of algorithm introduced first, then the concept of debugging. In terms of coherence across two disciplines, we would be looking for the overlap and connection among concepts. For example, in a third-grade lesson that is introducing the concept of algorithms within the context of an English Language Arts lesson, specifically a lesson on story structure (e.g., first, then, next, last) we would look for how the lesson connects the idea of an algorithm as a specific sequence of steps to the idea of story structure also as a specific sequence. A good example of how to connect these ideas is to have children use the Scratch technology to create a short, animated story in Scratch that uses the simple story structure, first, then, next, last. Indeed, Burke & Kafai (2010) have demonstrated that a similar technique has been successful with teaching older children about both coding and writing stories using Scratch.

5.2 Role of Computational Technology

The second quality indicator concerns the role of computational technology in the CT-integrated lesson. By the role of technology, we mean the way the technology is used to support student learning, with a special focus on the extent to which the technology supports learning in all of the disciplinary topics included in the lesson. For example, as noted above, Burke & Kafai (2010), utilized the Scratch technology to examine student learning of coding, as well as their learning related to creative writing. We (Authors, 2021) found similar support for student learning of coding and elements of narrative when Scratch was integrated into a fourth-grade classroom. A primary reason why Scratch appears to be a suitable technology for teaching Language Arts (for example, narrative elements in storytelling), is the design of the technology itself. Scratch is developed using a theatrical metaphor of “the stage” for which one creates or selects a background, selects or creates characters (termed “sprites” in the Scratch software), and then develops either some sort of animation, an interactive story or an interactive game. Aris is another computational media that uses a narrative metaphor to engage students in game design and storytelling across any number of disciplines (Litts, Lewis & Mortensen, 2020).

Other computational technologies that support CT-integrated learning include those that support both programming and modeling activities like NetLogo (Wilensky, 1999), AgentSheets (Repenning, 1993) and CTsim (Sengupta et al., 2013). In these technologies, students use code to set the parameters to run simulations and create models. These technologies can also be used to create games that integrate learning in CT, science and math (Clark & Sengupta, 2020), and potentially appeal to youth interests in doing so. The appropriateness of a computational technology for supporting learning across the integrated disciplines is key here.

5.3 Assessment

Our third quality indicator is assessment. Assessment plays a key part in any type of learning for students and educators alike. Through well-designed assessments, aligned with learning goals and with clear criteria, students not only gain information on what they know and where they need improvement, but also establish trust in teachers (Guskey, 2003). Teachers utilize assessments to identify troubled spots, understand the nature of students' struggles, and examine and adjust their teaching methods (Guskey, 2003). Assessments of CT-integrated lessons must measure both CT and domain knowledge of the integrated academic subjects. For assessment to reflect both CT and domain knowledge in subject matter, Drake and Burns (2004) suggest pulling overlapping standards apart to record separately students' progress in each subject of the integrated curriculum.

Moreover, the use of multimodal assessments is key, especially where equity is concerned (Burke & Kafai, 2010). Multimodal assessments include but are not limited to software metrics, audio and video recordings, and observation notes. These assessments allow students to demonstrate their understanding and competency applying concepts and skills and express their dispositions and attitudes towards computational thinking (Burke & Kafai 2010; Tang, Yin, Lin, Hadad, & Zhai 2020). Based on these understandings, we include in our criteria for assessment the following subcategories: alignment with integrated learning objectives and multimodality.

5.4 Multiple Representations

The fourth quality indicator concerns the use of multiple representations in the lesson, and specifically representations that are relevant to the disciplines being integrated. Others have argued for the importance of developing representational competence (Ainsworth, 2006; Kozma & Russell, 2005) as an important aspect of learning in a specific discipline, like science. Representational competence refers to the ability to be able to read and understand specific modal representations. This is important because discipline specific representations encode the social and cognitive affordances in the material features of the representation (Kozma, 2003). Therefore, understanding disciplinary representations is an important element of understanding in the discipline.

Meanwhile, researchers have begun to identify representations that bridge disciplines, and would, therefore, be very useful in helping students learn in an interdisciplinary fashion. Sengupta et al., (2013) identified representations that result from computational modeling activities as specific to the fields of computer science, math, and science. These models are typically abstract representations of a phenomena. In this case, the concept that can be taught in an interdisciplinary fashion via the development of the representation is abstraction. This work was followed up by Clark and Sengupta (2020) who pioneered the use of modeling in disciplinary integrated gaming (DIGs) environments. They argue that "…the design of DIGs focuses on engaging students more deeply in specific representational practices of developing, interpreting, manipulating, and translating across specific disciplinary model types" (p.330). In our evaluative rubric, we analyze the existence of disciplinary representations in the lesson, and specifically, the degree to which specific representations are utilized and how they align with the CT-integrated learning goals of the lesson.

5.5 Play

Our fifth quality indicator is the role of play in the lesson. Play is an important component of learning for young children as the National Association for the Education of Young Children (NAEYC) (2020) argues "Play promotes joyful learning that fosters self-regulation, language, cognitive and social competencies as well as content knowledge across disciplines. Play is essential for all children, birth through age 8" (p. 9). They further argue that play is a major developmentally appropriate approach for preschool and early elementary and a universal phenomenon across all cultures (NAEYC, 2020). Since our evaluative rubric is designed for evaluating elementary level (Kindergarten – Grade Five) CT-integrated curriculum, developmental appropriateness is one of the theoretical lenses through which we developed our rubric.

What makes an activity play? How is play defined? Vygotsky (1978) argues that there are four criteria that make up play including: (1) play is fun, (2) play has rules (explicit or implicit) (3) play includes imaginary situations (explicit or implicit), and (4) play has a purpose. We included the latter three elements in our evaluative rubric. We excluded the criteria of fun due to the thoroughly subjective nature of the concept. Further, we divide our analysis of play as an aspect of the CT-integrated curriculum into two sub-categories: playful activities and games. We distinguish among these elements as Vygotsky (1978) did to provide a level of precision in analysis. For example, in playful activities, the rules are implied and the imaginary situation is explicit. Whereas in a game, the rules are explicit and the imaginary situation is implied. Therefore, our evaluative rubric first distinguishes between playful activities versus games that might be used to present the curriculum. Then, we evaluate the degree to which the elements of play (has rules, has an imaginary situation, has a purpose) are discernible for students and support learning.

5.6 Equity

The sixth and final quality indicator is equity. Due to the long history of the oppression of people of color in the United States (Kendi, 2016), we are specifically interested in addressing racial equity in our evaluative rubric. This is not meant to underplay the importance, and indeed, necessity of addressing gender equity, but due to the context in which we have developed our evaluative rubric, our current focus is on racial equity. Here, we draw most notably upon the work of Muhammad (2020), who developed a four-layered equity framework named the Historically Responsive Literacy (HRL) Framework. This framework includes the following elements: (1) identity development, (2) skill development, (3) intellectual development, and (4) Criticality. While Muhammad's (2020) framework focuses on the teaching of literacy, we adopt it here due to the relevance of the elements to supporting students in developing computational thinking. Muhammad (2020) argues that students have the potential for success when their identity such as culture, gender, and race is incorporated in the curriculum and affirmed. This notion is affirmed by the work of Cheryan, Plaut, Davies & Steele (2009) who demonstrated how cultural elements of computer science learning environments left women feeling excluded. Without seeing themselves and their interests reflected in computer science learning environments, women were less interested in pursuing the field.

Muhammad (2020) defines skills as "competence, ability, and expertise based on what educators deem to be important for student learning in each content area" (p. 85). She argues that skills should be taught in a context that provides social, emotional, or intellectual relevance to students, and they should be given opportunities to put the skills learned into practice. Muhammad (2020) defines intellect as "what we learn or understand about various topics, concepts, and paradigms" (p. 104). In other words, as learning takes place, one asks, "What am I becoming smarter about?" According to Muhammad, intellect also holds the meaning of applying the knowledge learned into action. Finally, Muhammad (2020) differentiates between "c" critical and "C" Critical. For her, while critical means to think deeply about something, Criticality is related to power, power dynamics, entitlement, oppression, and equity. She defines Criticality as the "…capacity to read, write, and think in ways of understanding power, privilege, social justice, and oppression, particularly for populations who have been historically marginalized in the world" (p. 120). We adopt Muhammad's four-layer approach in evaluating CT-integrated lessons for their attention to equity. In other words, we evaluate the degree to which children can see themselves in the lesson (identity), learn and practice skills in the lesson (skills), become knowledgeable about computer science, what it is and how it fits, broadly, into our world (intellect) and engage with aspects of societal oppression (Critically). This last element may be easier to accomplish through CT-integrated lessons in Language Arts and Social Studies.

6 RUBRIC SCALE AND APPLICATION

The evaluation rubric has 14 items, including the quality indicators described above and their subcategories. Seven items might not apply to some lessons due to their absence, including 1) role of computational technology, 2) playful activity rules, 3) playful activity purpose, 4) playful activity imaginary situation, 5) game purpose, 6) game rules, and 7) game imaginary situation. The 14 items in the rubric have an equal weight of 4 points each. Four rating categories are utilized as has been recommended in the literature (Stone, 2003), each of which corresponds to a score ranging from 1 to 4 points.

Our application of the rubric started with a testing evaluation of 11 lessons. Two research assistants rated the lessons independently and compared the results afterward. Disagreements were resolved through discussions to clarify the criteria for the four grading levels and the presence and absence of certain curricular elements, such as play or technological tools. For example, to differentiate "good" (3 points) from "excellent" (4 points) for the category "skill," the defining element was decided to be "explicit teaching or discussion of how the skills reflect the professional practices of computer scientists or professionals of other disciplines." The evaluators then followed the agreed-upon research notes detailing these clarifications to grade the lessons. We make judgments based on our experience as researchers and veteran K-12 educators. We utilized simple scoring and converted the score into a percentage as the overall rating. Since the rubric is over ten pages in length, we are unable to provide an evaluative example here.

7 CONCLUSION

Here we have presented the design of our evaluative rubric for CT-integrated lessons in the elementary grades. This evaluative rubric is an important adjunct to the tools CS educators and researchers have available to them for selecting and/or creating quality, CT-integrated curriculum for elementary schools. As noted earlier, due to time constraints, it is most likely that the discipline of computer science will need to be integrated across the curriculum at the elementary level, if it is to be taught at all. We have endeavored to design a comprehensive evaluative rubric. This rubric not only attends to types of integration (multidisciplinary, interdisciplinary, and transdisciplinary approaches), but also to what to look for in terms of quality (conceptual coherence, the role of technology, assessment and the use of multiple representations). Finally, we have attended to both the developmental appropriateness of pedagogical approaches, as well as issues of equity in curriculum development. Because of the integrated nature of including CS in the elementary curriculum, a rubric such as ours is an important and much needed tool.

Our future work includes applying this rubric to a set of 115 lessons created for kindergarten through 5th grade, both to validate the use of the rubric, as well as evaluate the quality of the curriculum.

8 REFERENCES

Ainsworth, S. (2006). Deft: A conceptual framework for considering learning with multiple representations, *Learning and Instruction, 16,* 183-198.

Authors, (2021).

Bers, M.U., Flannery, L. Kazakoff, E.R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education, 72*, 145-157.

Burke, Q., & Kafai, Y. B. (2010, June). Programming & storytelling: opportunities for learning about coding & composition. *In Proceedings of the 9th international conference on interaction design and children* (pp. 348-351).

Case, R. (1994). Our crude handling of educational reforms: The case of curricular integration. *Canadian Journal of Education, 19*(1), 80-93.

Clark, D.B., & Sengupta, P. (2020) Reconceptualizing games for integrating computational thinking and science as practice: collaborative agent-based disciplinarily-integrated games, *Interactive Learning Environments, 28*(3), 328-346, DOI: 10.1080/10494820.2019.1636071

Cheryan, S., Plaut, V.C., Davies, P.G., & Steele, C.M. (2009). Ambient belonging: How stereotypical cues impact gender participation in computer science. *Journal of Personality and Social Psychology, 97*(6), 1045-1060.

Davison, D.M., Miller, K.W., & Metheny, D.L. (1995). What does integration of science and mathematics really mean? *School Science and Mathematics, 95*(5), 226-230.

Dong, Y., Cateté, V., Jocius, R., Lytle, N.,Barnes, T. Albert, J., Joshi, D., Robinson, R., & Andrews, A., (2019). PRADA: A practical model for integrating computational thinking in K-12 education. *In proceedings of the annual meeting of the ACM SIGCSE '19, February 27–March 2, 2019, Minneapolis, MN, USA.*

Drake, S.M. & Burns, R.C. (2004). Meeting standards through integrated curriculum. *Association for Supervision and Curriculum Development.* Alexandria, VA

Fogarty, R. (1991). Ten ways to integrate curriculum. *Educational leadership, 49*(2), 61-65.

Guskey, T. R. (2003). How classroom assessments improve learning. *Educational Leadership, 60*(5), 6-11.

Hinde, E.T. (2005). Revisiting curriculum integration: A fresh look at an old idea. *The Social Studies, 96*(3), 105-111. DOI 10.3200/TSSS.96.3.105-111

Israel, M., & Lash, T. (2020). From classroom lessons to exploratory learning progressions: mathematics+ computational thinking. *Interactive Learning Environments 28*(3), 362–382.

Israel, M., Pearson, J.N., Tapia, T., Wherfel, Q.M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education, 82*, 263-279.

Kendi, I.X. (2016). *Stamped from the beginning: The definitive history of racist ideas in America.* Nation Books.

Kiray, S. A. (2012). A new model for the integration of science and mathematics: The balance model. *Energy Education Science and Technology Part B: Social and Educational Studies, 4*(3), 1181–1196.

Kozma, R. (2003). The material features of multiple representations and their cognitive and social affordances for science understanding. *Learning and Instruction, 13,* 205-226.

Kozma, R., & Russell, J. (2005). Students becoming chemists: developing representational competence. In J. K. Gilbert (Ed.), *Visualizations in science education* (pp. 121-146). Dordrecht, The Netherlands: Springer.

Muro, M., Liu, S., Whiton, J. & Kulkari, S. (2017). Digitalization and the American workforce. *The Brookings Institute.* Online at https://www.brookings.edu/research/digitalization-and-the-american-workforce/

National Association for the Education of Young Children, (2020). Developmentally appropriate practice: *A position statement of the National Association for the Education of Young Children.* Online at https://www.naeyc.org/sites/default/files/globally-shared/downloads/PDFs/resources/position-statements/dap-statement_0.pdf

Repenning, A. (1993). Agentsheets: A tool for building domain-oriented visual programming. *Conference on Human Factors in Computing Systems.* 142–143.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: a theoretical framework. *Education and Information Technologies, 18,* 351e380. http://dx.doi.org/10.1007/s10639-012-9240-.

Settle, A., Franke, B., Hansen, R., Spaltro, F., Jurisson, C., Rennert-May, C., & Wildeman, B., (2012). Infusing computational thinking into the middle- and high-school curriculum. *In proceedings of the annual meeting of the ACM ITiCSE'12,* July 3–5, 2012, Haifa, Israel.

Sherwood, H., Martin, W., Rivera-Cash, E. Yan, W., Adair, A., Pierce, M. Liu, R., Fancsali, C., Israel, M. (2021). Diverse approaches to school-wide computational thinking integration at the elementary grades: A cross-case analysis. *In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages.

https://doi.org/10.1145/3408877.3432379

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, *148*, 103798.

U.S. Department of Education, National Center for Educational Statistics (2021). The condition of education 2021 *(NCES 2021-144).* Children's internet access at home. Online at https://nces.ed.gov/fastfacts/display.asp?id=46

Vars, G.F. (1991). Integrated curriculum in historical perspective. *Educational Leadership, 10*, 14-15.

Vasquez, J. A., Comer, M., & Sneider, C. (2013). *STEM lesson essentials, grades 3–8: Integrating science, technology, engineering, and mathematics.* Portsmouth, NH: Heineman.

Waterman, K.P., Goldsmith, L., & Pasquale, M. (2020). Integrating computational thinking into elementary science curriculum: An examination of activities that support students' computational thinking in the service of disciplinary learning. *Journal of Science Education and Technology 29* (1), 53–64.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U., (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*, 127–147. DOI 10.1007/s10956-015-9581-5.

Wilensky, U. (1999). NetLogo. *Center for Connected Learning and Computer-Based Modelling (http://ccl.northwestern.edu/netlogo).* Northwestern University, Evanston, IL.