Culturally Responsive Computing: Extending Theory to Early Childhood Educational Contexts

James A. Larsen – Department of Theory & Practice in Teacher Education, University of Tennessee, Knoxville, United States, JLarse13@utk.edu

Bethany Parker – Department of Theory & Practice in Teacher Education, University of Tennessee, Knoxville, United States, Betdpark@vols.utk.edu

Frances K. Harper – Department of Theory & Practice in Teacher Education, University of Tennessee, Knoxville, United States, FHarper1@utk.edu

Amir Sadovnik – Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, United States, ASadovnik@utk.edu

Lori Caudle – Department of Child and Family Studies, University of Tennessee, Knoxville, United States, LCaudle@utk.edu

Margaret Quinn – Department of Child and Family Studies, University of Tennessee, Knoxville, United States, MQuinn10@utk.edu

The CRRAFT Partnership

Abstract: A growing movement towards expanding computer science education in K-12 has broadened gaps in computing opportunities along lines of race, ethnicity, class, and gender. Emergent theories and practices related to culturally responsive computing show promise in addressing this gap; however, little is known about engaging culturally and linguistically diverse preschoolers in computer science. The current study utilized qualitative content analysis to explore how an extant theory of Culturally Responsive Computing aligns with an early childhood culturally relevant robotics curriculum. Findings suggest that while the assumptions of culturally responsive computing were evident throughout the curriculum, there are several key considerations when extending the theory to early childhood contexts. Overarching themes included (1) emphasizing the value of non-digital tools and activities and (2) aligning the goals of culturally responsive computing with children's current level of social development.

Keywords: Early Childhood Education, Robotics, Computer Science, Culturally Responsive Teaching

Introduction

Computational thinking (CT), which involves using computer science concepts and techniques (e.g., algorithms, debugging) to solve problems, is widely considered an essential skill for promoting academic and career success in the 21st century (e.g., Wing, 2008). Unsurprisingly, scholars, practitioners, and government officials have called for an expansion of computer science education in the United States and abroad. However, opportunities to engage in quality computer science education are largely divided along lines of race, ethnicity, gender, and class (State of Computer Science Education Report, 2019). Furthermore, despite a growing movement to increase access and participation, there are few examples of programs designed specifically to promote sustained engagement among students who are underrepresented in computer science.

To address this gap in research and practice, several scholars have attempted to adapt and apply culturally relevant and culturally responsive teaching frameworks to computer science education, and emerging research evidence demonstrates the positive effects of such approaches (e.g., Scott & White, 2013), predominantly focused on older student populations (e.g., middle school). Given the increased popularity of introducing educational robotics and coding activities in pre-k classrooms, it is important to extend this body of work to early childhood contexts. In this study, we sought to fill this gap in the literature by critically examining how an extant theory of culturally responsive computer science education aligns with an early childhood culturally relevant robotics program designed to support culturally and linguistically diverse preschoolers' CT skills and sense of belonging in computer science. Specifically, we asked: 1) *in what ways does the curriculum engage students in culturally responsive computing?* and 2) *what are the ways, if any, that culturally responsive computing theory should be modified for early childhood contexts?*

Culturally Responsive Computing Theory

Drawing from theories of culturally responsive teaching and professional experiences applying culturally responsive theory in computer science programs, Scott et al., (2015) proposed five tenets of Culturally Responsive Computing (CRC) to inform the design of meaningful computer science education experiences for students from historically underrepresented groups. First, CRC theory maintains that *all students are capable of digital innovation*, and computer science education must extend beyond learning foundational knowledge and skills to mastery experiences and, ultimately, opportunities to contribute to innovation and creation. The second tenet expands on students' capacity for digital innovation and suggests that educators must create a *learning context that supports the transformational use* of *technology*. The authors indicate that these learning contexts should be characterized by high expectations, autonomy support, and opportunities for students to understand and push the rules and limitations of computer science technologies and techniques.

Tenets three and four, respectively, suggest that *learning about one's self along various intersecting sociocultural lines allows for technical innovation* and that *technology should be a vehicle by which students reflect and demonstrate understanding of their intersectional identities*. Taken together, these tenets make the case that for computer science to be culturally responsive, it must (1) offer students the opportunity to learn about and understand the multiple layers of their identities, (2) help students navigate issues of equity, equality, and social justice, and (3) challenge students to manipulate digital media to create counter-normative images of themselves. The fifth and final tenet of CRC maintains that *barometers for technological success should consider who creates, for whom, and to what ends rather than who endures socially and culturally irrelevant curriculum.* This tenet ensures that metrics for evaluating computer science education for diverse student populations sets a high standard for learning outcomes to consider student empowerment and participation in social justice. It should be noted, however, that this theory was derived from researchers' experiences with late elementary and middle-school students. As a result, little is known about the relevance of the CRC theory tenets in early childhood contexts.

Current Study Background

This study represents one dimension of a larger project – the Culturally Relevant Robotics: A Family-Teacher Partnership (CRRAFT) – with the primary objective of creating home- and school-based programs to promote BIPOC and linguistically diverse preschooler's computational thinking and a sense of belonging in computer science. Design-based research (DBR) methodology provided the overarching framing for the program co-development by a team of university researchers and teacher educators, an instructional coach, teachers, and parent representatives from two public, Title I preschools in the southeastern region of the United States. The iterative DBR approach included the design, development, formative evaluation, testing, revision, and retesting of CT learning activities in classrooms and homes. The project included three DBR stages, each with iterative cycles of design, testing, and revision. This paper reports on an analysis following the first DBR stage, which involved the iterative co-development and pilot testing of the CRRAFT school-based program.

Study Rationale

The current study expands on the literature exploring CRC (Scott et al., 2015) as a theory of addressing the computing education opportunity gap by exploring how its tenets map onto the CRRAFT school-based program. This research has the potential to advance CRC theory and inform practice in early childhood computer science education. More specifically, this study will contribute to researchers' and practitioners' understanding of how to best support diverse learners' CT and sense of belonging when students are just beginning their formal education. As a result, educators will be more equipped with the knowledge and tools necessary to support long-term participation and, therefore, increased representation in computer science-related academic programs and careers.

Method

Materials

Written CRRAFT program documents from the conclusion of the first DBR stage served as the primary data sources for this study. The program consists of four distinct curricular "phases," each with a classroom-based plan and a corresponding home-based plan. Each phase explicitly highlights one of the powerful ideas for CT in early childhood (Bers, 2019): (1) algorithms/sequencing; (2) the design process; (3) modularity; and (4) control structures. Given the teacher education focus of the SITE Conference, this paper focuses exclusively on the school-based program.

The written program documents include four sections for each phase: (1) an introductory page about CRRAFT; (2) two pages with relevant information about the emphasized powerful CT idea, a summary of key concepts, and information about materials; (3) a brief, one-page lesson plan for a focal experience, which all teachers agreed to teach during the unit; and (4) two or three pages with brief descriptions of activities, to support students' CT learning, from which teachers could use as appropriate. The current analysis focused on sections 2 and 3 to ensure the findings reflect what *all* CRRAFT teacher partners utilized and implemented. A full copy of the curriculum is available at www.crraft.org.

Analytical Approach

This study adopted a directed approach to qualitative content analysis, which is appropriate when seeking to validate or extend theory (Hsieh & Shannon, 2005). Specifically, we utilized conceptual and relational analytic approaches. Conceptual analysis is a qualitative content analysis approach that allows researchers to identify the ideas present in a text, either explicitly or implicitly, while relational analysis is used to identify relationships between the concepts present in a text (Carley, 1990). In this particular study, we were interested in identifying the presence of, and relationships between, specific concepts related to CT and CRC because the existence and interactions of those concepts in the curricular materials characterize the opportunities to learn CT and engage in CRC.

The coding process began by independently creating and applying descriptive codes to Phase 1 documents. We started with a priori codes for CT (e.g., modularity, debugging, sequencing/algorithms) based on the Bers (2019) framework and relied on our expertise in computer science and CT in early childhood to identify implicit occurrences of those concepts. Given the lack of examples of CRC in early childhood, we decided to create codes for CRC inductively, relying on our expertise in early childhood and culturally responsive education. The authors then discussed codes and points of disagreement until consensus was reached on descriptive code definitions and applications. Final descriptive codes (Fig. 1) were then applied to all documents. Next, a review of excerpts by descriptive code was conducted alongside theory of CRC such that descriptive codes were mapped to the five tenets.

This process of mapping the CRRAFT curriculum onto the original tenets of CRC theory involved several steps. First, we identified broad areas of alignment between the curriculum and each CRC tenet. Next, we established agreement on the primary codes most strongly related to areas of alignment identified in step one and reviewed all excerpts in the curriculum that included those primary codes. Finally, we established agreement on patterns of descriptive codes that co-occurred with primary codes established in step two. Altogether, this analysis provided insights into opportunities for students to engage in CRC across the program, which provided information about how to extend CRC theory to early childhood contexts.

Results

Analyses revealed that while evidence of all five tenets of CRC theory were present throughout the curriculum, additional considerations and adaptations are warranted when extending CRC to early childhood contexts. Key findings are organized based on curricular themes and patterns corresponding to CRC tenets.

Tenets 1 & 2: Unplugged "Innovation" and "Transformational Use"

The primary finding of our analyses regarding the first two tenets of CRC was a unifying theme: extending the original tenets beyond their original framing was necessary to encompass a broader range of CT experiences in early childhood. Specifically, the curriculum promoted *innovative and transformational use of both digital and non-digital technologies and skills*. The CRRAFT curriculum and materials offer challenging opportunities for culturally and linguistically diverse learners to advance from basic foundational knowledge and skill acquisition to mastery, innovation, and creation (tenet 1). To this end, the use of *non-digital technology and skills as tools for creative expression* emerged as an overarching theme involving several frequently co-occurring codes: *unplugged*, and *student-directed* activities, *create*, *imagine*, *share*, *improve (as components of the design process)*, and *project ownership*. For example, in Phase 1 children are introduced to the computer science concept "sequencing" by learning to recognize how the order of events in a story shapes its meaning. They are then encouraged to contemplate what would happen if the order of events were changed and to "create their own stories and sequences based on the changes" [Student-Directed, Create, Unplugged]. In this way, the curriculum helps children recognize how computer science concepts and skills can be used as vehicles for innovation and creative expression.

In regards to the second tenet, the CRRAFT curriculum highlights several ways educators can foster a learning context that encourages students to "make technologies do what they want them to do" (p. 422). It became evident that in the context of early childhood, expanding this tenet to explicitly emphasize l*earning contexts supporting transformational use of digital and non-digital technologies* was a central theme. We identified several codes related to the learning context, such as *play-based*, *student-directed*, *project ownership*, and *collaboration* that co-occurred with codes *purpose*, *imagine*, *test*, and/or *improve*, to elicit the transformational use of technologies and techniques during *plugged* and/or *unplugged* activities. For example, in preparation for the Phase 2 focal experience (i.e., building a community robot), children first work alone or in small groups to build a prototype using crafts and recyclables and "imagine different ways to use the materials" [Student-Directed, Unplugged, Create, Imagine]. Next, they "show and describe their prototypes" to help the class "imagine even better solutions or ways of doing things" [Unplugged, Project Ownership, Share, Imagine, Improve]. Throughout this activity, teachers create a learning context that supports the joint exploration of the design process for transformational purposes (i.e., creating a robot that can solve problems in the community).

Tenets 3 & 4 "Intersectionality & Identity Exploration" Through Collaboration and Funds of Knowledge

In much the same way as tenets one and two, tenets three and four of Scott et al.'s (2015) expanded CRC framework do not map neatly onto the CRRAFT curriculum, but can be expanded upon within the early childhood context. The curriculum makes two significant contributions to the third and fourth tenets of the CRC framework. First, it provides children with *opportunities to learn about and from one anothe*r; and second, it provides young children with *diverse representation in computer science that leverages children's funds of knowledge*. Together, these contributions extend what it means for young children to "learn about oneself along various intersecting sociocultural lines…" (Scott et al., 2015, p. 12) and "...demonstrate understanding of their intersectional identities" (Scott et al., 2015, p. 15).

This extension of CRC theory is evident in the curriculum through co-occurring codes related to a) general messages about computer science and specific concepts such as *coding* and *hardware/software*, b) opportunities to leverage *funds of knowledge* such as *culture as assets* and *cultural practices*, and c) activities that promote social interactions such as *collaboration* and *share*. Because the CRRAFT curriculum recognizes that "young children benefit from learning computer science collaboratively," [Collaboration, Computer Science] lessons and activities encourage educators to "prioritize group work and cooperative [work] over individual work" [Collaboration, Computer Science].

Beginning in Phase 1 of the CRRAFT curriculum, educators are encouraged to recognize the strengths that cultural diversity brings to the digital world through their use of collaborative learning experiences and the use of materials that represent the social identities of the students in their classrooms. Specifically, educators are encouraged to "be sure to celebrate all the rich traditions…that children have experienced in their homes and communities" [Culture as Assets, Cultural Practices]. The practice of highlighting children's cultures and communities as positive contributors to computer science continues throughout the curriculum and evolves into rich, hands-on experiences. For example, in the Phase 4 Focal Experience, educators not only discuss the significance of weaving in Latin American cultures and how "historically, weavers are some of the first computer programmers, [Culture as Assets, Cultural Practices, Computer Science] they also give children the opportunity to produce their own patterned weaving crafts and "connect these patterns and conditions to writing loops and if/then commands for Kibo" [Culture as Assets, Control Structures, Hardware/Software].

Tenet 5: "Barometers for Technological Success"

Analyses revealed the CRRAFT curriculum is strongly aligned with tenet five in the value it places on addressing questions of who creates, for whom, and to what ends. Providing students with "a learning context in which [they] appropriate the technical and research skills to dismantle the system…" (p. 428) is a flexible goal that provides educators with a way that is both age-appropriate and culturally relevant. Within the curriculum this is done through conversations with children about how to solve problems they have identified in their communities.

To this end, frequently co-occurring codes that relate to appropriating technical skills in computer science include *hardware/software*, such as *robots* and *computers*, and the *design process*, which comprises specific concepts such as *improve*, *imagine*, *plan*, and *create*. Co-occurring codes related to dismantling the system in ways that empower historically marginalized students include *problem-solving, purpose, project ownership*, and *funds of knowledge*. For example, in the Phase 3 Focal Experience, children are asked to reflect on and make connections between technology use and real-world problem-solving. Educators guide children in "discussing the cities and communities children live in and the problems robots (could) solve" [Purpose, Problem-Solving, Funds of Knowledge, Robots]. Similarly, educators ask children questions, such as "how can we use robot[s] to make life better for the people who live in our city," [Purpose, Funds of Knowledge, Imagine, Robots, Project Ownership]. Throughout this process, children are the ones who make decisions about what problems exist, which problems warrant solving, and how to solve them.

Study Significance and Contributions

Given the recent expansion of early childhood computer science education, the demand for increased representation in computer science, and evidence of CRC as a powerful framework for addressing opportunity gaps, the current study offers several timely insights to the field of teacher education. Study findings indicate that CRC theory is applicable to early childhood computer science education and can be used to guide educators who work with diverse student populations. However, educators should take into consideration that CRC enactment in early childhood contexts is not identical to CRC with older student populations.

First, the original tenets' emphasis on digital hardware, software, and techniques can be modified such that young learners engage in CT and computer science in non-digital ways, which aligns with research recommendations to emphasize "unplugged" platforms when first introducing computing concepts (Lee & Junoh, 2019). Even in the context of a culturally relevant robotics program, the curricular activities demonstrate the potential for non-digital tools and activities to support CT skills and help children draw personal connections to computer science topics. This finding also highlights a way to effectively reduce potential barriers such as materials costs and teachers' knowledge of sophisticated hardware and software.

Second, current study findings also point to developmental factors that warrant the adaptation of CRC theory in early childhood contexts. While Scott et al. (2015) emphasize the importance of supporting students' understanding of intersectionality as it relates to issues of equity and social justice, the CRRAFT curriculum reflects age-appropriate activities used to support foundational identity development and social awareness. That is, early educators can slightly modify CRC to help children learn about themselves, others, and themselves in relation to others by (1) offering ample opportunities for children to work with and learn from one another in the classroom and (2) creating a positive context for children to explore their own and other's cultural identities. Altogether, the present study draws attention to the ways CRC can be adapted to support young, diverse learners' CT and emerging sense of self as it relates to computer science. This gives early educators a powerful theoretical framework for promoting continued student participation, increased representation, and excellence in the field of computer science.

Acknowledgments

This work was supported with funding from the National Science Foundation, Computer Science for All Program (Award #2031394). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Bers, M. U. (2019) *Coding as a playground: Programming and computational thinking in the early childhood classroom* (2nd ed.). Routledge.

Bers, M. U., & Sullivan, A. (2019). Computer science education in early childhood: The case for Scratch Jr. *Journal of Information Technology Education: Innovations in Practice*, *18*, 113–138.

Carley, K. M. (1990). Content analysis. In *The Encyclopedia of Language and Linguistics*, (Vol. 2, pp. 725-730), Edinburgh, UK: Pergamon.

Hsieh, H. F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative health research*, *15*(9), 1277-1288. doi: 10.1177/1049732305276687

Lee, J., & Junoh, J. (2019). Implementing unplugged coding activities in early childhood classrooms. *Early Childhood Education Journal*, *47*(6), 709-716. https://doi.org/10.1007/s10643-019-00967-z

Scott, K. A., Sheridan, K. M., & Clark, K. (2015). Culturally responsive computing: A theory revisited. *Learning, Media and Technology*, *40*(4), 412–436. https://doi.org/10.1080/17439884.2014.924966

Scott, K. A., & White, M. A. (2013). COMPUGIRLS' standpoint: Culturally responsive computing and its effect on girls of color. *Urban Education*, *48*(5), 657-681. doi:10.1177/0042085913491219

State of Computer Science Education: Equity a Diversity. (2019). Retrieved from https://advocacy.code.org

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725. doi:10.1098/rsta.2008.0118

Wyeth, P. (2008). How young children learn to program with sensor, action, and logic blocks. *Journal of the Learning Sciences*, *17*(4), 517–550. https://doi.org/10.1080/105084008023

Primary Codes	Definition	Sub-Codes & Definitions (if applicable)
Algorithms and Sequences	Children learn about or engage in the process of creating a list of steps that can be used to perform a task	**Order Matters:** The importance of sequential order is explicitly stated **Algorithm and/or Sequence Cognition:** Students are asked to recall, imagine, etc. the steps in an algorithm/sequence without actually performing those steps.
Design Process	Children learn about or engage in some or all steps of the engineering design process including ***ask***, ***imagine***, ***plan***, ***create***, ***test***, ***improve***, and ***share***.	
Modularity	Children learn about or engage in the process of deconstructing complex problems or algorithms into smaller, more manageable parts.	
Control Structures	Children learn about or engage in the use of "building blocks" of computer programming	**Loops:** instructions to repeat an action or sequence **Conditionals:** if/then instructions that indicate under what circumstances to carry out an action or sequence
Hardware and Software	Children learn about the interactions between "***hardware***" and "***software***" components of computers and Robotics	More specific codes are applied when lessons/activities involve specific types of hardware such as ***computers*** and/or ***robotics***
Representation	Children learn about or engage in the use of symbols to convey meaning, especially as it relates to programming languages.	
Debugging	Children learn about or engage in the process of identifying, troubleshooting, and resolving issues when an algorithm, program, or approach to a problem does not function as intended	
Promoting a Sense of Belonging	The curriculum integrates opportunities for children to take ownership and share their voices during CS and robotics activities	**Computer Science**: The curriculum helps children make sense of what CS is and what it means to do CS **STEM:** The curriculum helps children make sense of what STEM is and what it means to do STEM **Project Ownership:** The activity involves children co-constructing/building things of their own design and/or the lesson explicitly mentions the importance of fostering a sense of student ownership.| **Purpose:** The curriculum outlines the purpose of learning and doing CS, including the important functions CS serves (personal, social, etc). **Self-Efficacy:** curriculum emphasizes building children's sense of competence and confidence in doing STEM and/or CS
Funds of Knowledge	The curriculum affords opportunities to connect CS and robotics activities/lessons to children's personal interests, experiences, and backgrounds.	**Cultural Funds:** lessons and/or activities leverage ***cultural practices*** and/or ***linguistic funds*** relevant to students' lives outside of school. This code may also be applied when the curriculum explicitly describes ***cultural differences as assets.*** **Home-Based Funds:** lessons and/or activities leverage children's home experiences, especially those related to ***home routines*** like brushing teeth, getting ready for bed, etc. **School-Based Funds:** lessons and/or activities leverage children's experiences at school such as ***school routines*** and/or explicit connections to non-CS ***curriculum and standards*** **Childhood Funds:** lessons and/or activities reflect ways of knowing and being specific to early childhood. This is demonstrated through clear evidence of ***developmentally appropriate practices*** and/or an emphasis on ***play-based learning***
Digital vs. Non-Digital Tools	Activity enactment involves the use of ***plugged*** technology such as educational robotics and/or ***unplugged*** materials such as non-digital coding games.	
Student-Directed	Activity enactment centers children as active participants in learning and/or involves children sharing their own ideas. This might involve the strategic use of ***questions*** to elicit children's ideas and encourage deeper thinking	
Teacher-Directed	Activity enactment centers around the teacher explaining content and/or objectives while children in a passive role	
Academic Knowledge and Skills	The curriculum references and/or draws connections to knowledge, skills, concepts, objectives, etc., that represent the established goals of the educational institution such as ***literacy*** and ***math***, important skills such as ***collaboration*** and ***problem-solving***, and valuable attributes such as ***persistence.***	

Figure 1. Final Codebook Used for Content Analysis.
Note. CS = Computer Science. STEM = Science, Technology, Engineering, and Math