Teaching Elementary Computer Science through Universal Design for Learning

Maya Israel
University of Florida
Gainesville, FL
misrael@coe.ufl.edu

Gakyung Jeong
University of Illinois
Champaign, Illinois
gjeong4@illinois.edu

Meg Ray
Cornell Tech
New York, New York
meg.ray@cornell.edu

Todd Lash
University of Illinois
Champaign, Illinois
toddlash@illinois.edu

ABSTRACT

Given the academic diversity of today's classrooms, elementary teachers engaged in computer science (CS) and computational thinking (CT) instruction must create CS/CT experiences that are accessible and engaging to a broad range of learners, including those with disabilities. One method of developing inclusive instructional experiences is through the Universal Design for Learning (UDL) framework, wherein teachers proactively design instruction for the broadest range of learners. Doing so may be challenging as elementary teachers may not be familiar with the UDL framework or may not have experience with applying UDL within CS/CT instruction. The purpose of this qualitative study was to investigate how four elementary teachers provided UDL-based instruction to academically diverse learners during CS/CT instruction. Teachers received professional development and instructional coaching related to UDL within CS/CT education. Data included teachers' lesson plans, coaching logs, and teacher interviews which were qualitatively analyzed and triangulated. Data revealed that teachers generally addressed all three UDL principles, with an emphasis on two of the principles (multiple means of engagement and multiple means of representing content) above the third principle (multiple means of action and expression). They focused on breaking tasks into steps, emphasizing student choice, and presenting information in multiple ways. Findings revealed nuanced implementation differences among the teachers as well.

CCS CONCEPTS

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

KEYWORDS

Elementary CS, Universal Design for Learning, Students with disabilities

ACM Reference Format:
Maya Israel, Gakyung Jeong, Meg Ray, and Todd Lash. 2020. Teaching Elementary Computer Science through Universal Design for Learning. In *The 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20), March 11–14, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3328778.3366823

1 INTRODUCTION

When speaking with elementary teachers providing computer science (CS) instruction, it becomes clear that they want to reach all learners. They often describe their desire to increase equitable access for their learners [6]. However, these same teachers often describe a lack of information about effective pedagogical approaches to guide them in including all learners in such experiences. Because of limited strategies specifically designed to support students with disabilities in elementary CS, teachers may use strategies that they find effective in other content areas[15]. One such pedagogical approach that has been applied in CS education is Universal Design for Learning (UDL) [13]. The purpose of this study was to investigate how elementary teachers used the UDL framework to meet the needs of their learners with disabilities during CS instruction.

2 BACKGROUND

2.1 Students with disabilities in K-12 CS Education

Approximately 6.7 million children in K-12 public schools receive special education services [11] under the Individuals with Disabilities Education Act (IDEA), federal legislation that ensures that students with disabilities receive an education tailored to their individual needs in the least restricted environment. Another group of students receive supports under Section 504 of the Rehabilitation Act of 1973, which is a federal civil rights law that prohibits discrimination against people with disabilities in programs receiving federal financial assistance (such as schools). Students receiving Section 504 supports typically require accommodations such as extended time to complete tasks and technology-supports such as text-to-speech software for reading. The majority of students with

SIGCSE '20, March 11–14, 2020, Portland, OR, USA
© 2020 Association for Computing Machinery.
ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.1145/3328778.3366823

disabilities are taught alongside their peers in general education settings [11]. Thus, it is reasonable to expect that these learners will be included in CS activities available to students without disabilities.

A growing body of literature suggests that with the appropriate supports, students with disabilities can be successful in CS education [9]; [16]; [17]. Snodgrass, Israel, and Reese (2016), for example, showed that when supports implemented in other content areas (such as mathematics) were used in CS education, students with disabilities meaningfully participated in CS activities. These studies highlighted the need in the CS education research community to include disability as part of broadening participation discourse. As K-12 CS education continues to gain momentum, the study of pedagogical approaches that result in meaningful inclusion of students with disabilities is essential [8]; [13].

2.1.1 Theoretical Framework: Universal Design for Learning (UDL). UDL is often described as a planning process that teachers use to reduce barriers to learning so that all students, including those with disabilities, can access instruction [14]; [12]. UDL rests on the premise that rather than modifying "one size fits all" instruction for learners for whom that instruction is not accessible, teachers should proactively plan instruction that is accessible and engaging to a broad range of learners. UDL is comprised of three broad principles based on cognitive neuroscience research [4]: (a) multiple ways for students to engage in learning, (b) multiple ways for content to be presented to learners, and (c) multiple ways for students to express their understanding. These principles are further broken down into nine guidelines (three per principle) and accompanying checkpoints that provide context for how the principles can be applied to instructional planning (see http://udlguidelines.cast.org/). The principles, guidelines, and checkpoints, when viewed together, can help teachers focus on how to increase access and engagement within the instructional goals, methods, materials, and assessments that are used during instruction.

2.1.2 Related and Prior Work: UDL and CS Education. The UDL guidelines were developed by CAST (i.e., the Center for Applied Special Technology) and have been widely used as a foundational document for UDL implementation. Despite the focus on UDL in other content areas such as K-12 STEM instruction (e.g.,[1]; [7]) and postsecondary computing instruction (e.g., [2]), UDL has not often been applied in the context of elementary CS instruction. Hansen and colleagues (2016) found that UDL and differentiation were necessary to ensure equitable and accessible CS experiences to academically diverse elementary students [5]. In another study, Ray and colleagues (2018) investigated how K-8 teachers taught CS to a wide range of learners, including those with disabilities, through a UDL-based approach. They found that most of the teachers used some aspects of UDL, but implementation of UDL was generally limited to either offering students choice in projects (an aspect of multiple means of engagement) or project planning supports (an aspect of multiple means of action and expression). This limited interpretation of UDL leads to multiple questions about how much support teachers require in order to more fully implement UDL in the context of CS education. Beyond these studies, little is known about how to implement UDL in the context of early CS education.

Table 1: Teacher Information

Name	Grade	Class Setting	License	Years teaching
Lester	2	Inclusive co-taught	Special and General Ed	4
Rios	K-3	Self-contained special ed	Special ed	4
Aldridge	3	Inclusive co-taught	General Education	2
Robinson	4	Gen Ed and Eng. as a New Language (ENL)	General Ed	3

2.1.3 Purpose and Research Questions. The purpose of this paper was to investigate how elementary teachers who received professional development and instructional coaching in UDL-based instruction implemented UDL within their inclusive CS classrooms. The research question that guided this study was: What does UDL look like in the context of elementary CS education?

3 METHODS

This study utilized a basic interpretive qualitative research design [10] to examine the degree to which UDL was present in teachers' instruction. Data included (1) lesson plans and instructional artifacts (e.g., worksheets, anchor charts), (2) teacher interviews, and (3) instructional coaching logs.

3.1 Setting and Participants

This study took place in a large urban school district in the Northeast that had a K-12 CS for all initiative. In this study, participating teachers were recruited based on the following criteria:

(1) Teachers signed the informed consent forms,
(2) Teachers had a range of learner diversity in their classrooms (e.g., students with disabilities, English language learners),
(3) Teachers participated in initial professional development and a five-week UDL instructional coaching cycle, and
(4) Teachers agreed to implement aspects of UDL in their CS instruction during the five-week coaching cycle.

Of the available teachers in this school district, four were purposefully selected based on the inclusionary criteria and the student diversity in their classrooms. Table 1 provides information about the teachers in this study.

All four teachers could be considered novice CS educators. This was Ms. Robinson's first year of teaching elementary CS. The other three teachers had one prior year of elementary CS. During this study, all four teachers taught CS once a week for a 45 minute block of time. The CS teaching activities varied among the teachers as they taught different grade levels and in different contexts. All began with unplugged activities such as those in *Hello Ruby.* With the exception of Ms. Rios, they then followed a similar trajectory

Table 2: Computing Activities

Computing activity	Lester	Rios	Aldridge	Robinson
Unplugged	x	x	x	x
Robotics	x	x	x	x
Code.org	x		x	x
Scratch	x	x	x	x
Codesters				x
Stations		x		

wherein they transitioned to Code.org Fundamentals for a couple of months followed by robotics activities with the Dash robot or Robot Mouse. They then transitioned to open-ended projects using Scratch. Ms. Robinson, in 4th grade, then moved onto Codesters to introduce students to Python. Ms. Rios followed a different learning trajectory wherein she spent approximately two months on unplugged activities tied to *Hello Ruby* and then transitioned to the Robot Mouse followed by Dash robot. In both robotics units, Ms. Rios explicitly taught the students how to use these devices. Once students were familiar with the activities, she set up stations wherein students cycled through different computational activities. All teachers addressed computational concepts related to sequencing and algorithms. Other than Ms. Rios, the teachers also focused on repetition. Ms. Robinson also focused on general syntax in Codesters, as well as on strings, integers, and making shapes using parameters in a function within coordinate plans.

3.1.1 Professional development (PD) and instructional coaching. Only Ms. Rios had experience with UDL prior to this study. Consequently, before the initial PD, the teachers were asked to complete a self-guided module that included an introduction to UDL, with videos and web resources such as an overview of the CAST website, the UDL guidelines, and materials created by this research team that connected UDL to CS education. The subsequent face-to-face PD focused on deeper exploration of how the UDL framework could be applied in lesson planning and implementation. A major aspect of this PD included showing teachers examples and providing them with time to discuss strategies and implementation steps. Next, the teachers worked with an instructional coach in a five-week coaching cycle on strategies for including students with disabilities and English as a new language learners (ENL) in CS education. The coach met with the teachers weekly to co-plan and co-teach with the teachers. The structure of this coaching included: (a) time to co-plan instruction, (b)co-teaching, and (c) lesson debrief. Co-planning involved a discussion of lesson objectives, ways of increasing the flexibility of instruction through the UDL framework, and discussions of lesson logistics such as the need for materials and lesson pacing. Co-teaching involved a combination of modeling pedagogical approaches by the coach and opportunities for the teacher to lead instruction. Lastly, lesson debrief included a discussion of lesson successes and challenges, discussion of strategies used, and next steps needed for co-planning.

3.2 Data Collection

This study utilized three data sources: (a) lesson plans and accompanying artifacts, (b) coaching log notes, and (c) teacher interviews. All materials were collected during the five-week coaching cycle in which the teachers participated.

3.2.1 Lesson plans and accompanying artifacts. Lesson plans and associated artifacts were collected by the instructional coach and uploaded to a shared cloud-based account. These lesson plans included lesson goals, assessments, and necessary materials for implementation. Lesson plans were collected both during the 5-week coaching cycle as well as after the coaching intervention ended.

3.2.2 Coaching logs. Weekly coaching logs were conducted across the five-week coaching cycle. These coaching logs consisted of Google Forms with prompts for areas of discussion between the teachers and the coach. Areas within the coaching logs included general instructional context (e.g., number of students, grade level), instructional goals, and inclusive strategies discussed or observed by the coach (e.g., evidence of UDL, differentiation, modeling, formative assessments). Data from these coaching logs were then exported into a spreadsheet for later analysis.

3.2.3 Teacher interviews. Semi- structured interviews were conducted with the teachers at the end of the school year. Example questions included: (a) Can you talk a bit about to what extent you used the UDL framework in your CS instruction, and (b) Can you talk a little bit about the experiences of your students with disabilities, without identifying specific students? All questions included follow-up probes to encourage specific examples and additional information. Each interview lasted between 30 and 60 minutes. All interviews were audio recorded and transcribed for analysis.

3.3 Data Analysis

This study utilized a basic interpretive qualitative methodology [10]. A coding scheme was developed based on the UDL guidelines and checkpoints to ascertain the UDL implementation used by the teachers. Due to the specificity of the UDL guidelines and checkpoints developed by CAST, it was feasible to assign consistent codes based on UDL in the data. Once initial coding was complete, two researchers compared codes and clarified areas of disagreement. Interrater agreement was then computed using Cohen's Kappa coefficient. Cohen's Kappa is a measure of interrater agreement used for qualitative/categorical data and is considered more robust than percent agreement as it accounts for agreement between raters that might occur by chance [3]. The research team reached an interrater reliability which remained above .98 across all data sources. Using the Cohen's Kappa coefficient, scores between .8 and .99, indicate near perfect agreement. The researchers also tabulated the frequency of data in each UDL checkpoint and guideline to ascertain which areas of UDL were emphasized above others. This analysis was not intended to suggest that *more is better.* Rather, it was intended to unpack gaps in UDL implementation.

4 RESULTS

4.1 UDL Principles in CS Instruction

Data revealed that all four teachers implemented UDL within their instruction, although implementation varied among the teachers. Additionally, although tabulating instances of UDL does not correspond to understanding UDL implementation, this tabulation provided one lens into UDL areas of emphasis. This analysis revealed 145 UDL practices tied to the UDL guidelines and checkpoints. The distribution of these instances across principle was: Multiple means of engagement (n=63), Multiple means of representation (n=51), and multiple means of action and expression (n=31). This analysis revealed that primary emphasis was on engaging learners and representing content in flexible ways, with less emphasis on options for expression of understanding. Again, this frequency count does not indicate quality of UDL implementation. Rather, it showcased UDL checkpoints that were emphasized.

4.1.1 Multiple means of engaging and motivating learners. : Teachers addressed this UDL principle to the greatest extent as compared to the other two UDL principles. In fact, all four teachers addressed all components of this principle. First, they all offered choice in the development of their students' computational artifacts. However, the teachers also added a level of accountability to that choice so that students would need to think about their choices and show intentionality for those choices. For example, when Ms. Aldridge's students worked with peers on Scratch projects, she encouraged them to express their intentionality for their choices using sentence starters with four prompts: (a) My goal is to . . . (b) My first step is . . . (c) My next step will be . . . And (d) My last step will be . . . This type of activity not only heightened salience, but also helped the students with goal setting, which is a critical component of increasing student engagement.

Second, the teachers attempted to optimize relevance for the students by adding several real-world applications that were familiar to the students within the computing activities. An example of this approach occurred when Ms. Aldridge taught a lesson using the Dash Robot (https://www.makewonder.com/dash/) wherein students were asked to create a map and develop a story that directed Dash to go to different areas of the map. Once their stories and maps were created, the students programmed Dash to follow the map according to the story they wrote. Many of the students created maps of their neighborhoods or familiar areas; thus, this activity was personalized and contextualized in their lives.

Last, all the teachers encouraged sustained persistence. The primary way that they addressed this UDL guideline was by fostering student collaboration with their peers as a community of learners. Collaborative opportunities varied among the teachers, but they all encouraged collaboration and offered a structure in which the collaborations should occur. For example, Ms. Lester created an unplugged activity to teach decomposition wherein students decomposed the steps to a hand gesture dance. In her directions, she stated, "Work with your partner to create, decompose and record the code for a hand jive. One partner will create the hand jive and the other partner must decompose it and write the code." These directions were clarified to the students. Additionally, she provided the students with a template wherein the students would write their steps. The lesson plans also included facilitated group discussions and partner/pair programming. Ms. Aldridge introduced a pair programming protocol in assigned pairs and included explicit collaboration rules (e.g., when to switch roles), individual students' roles as navigator and driver, and explicit directions for each student's responsibility (coaching debrief). The teachers also all indicated that they purposefully paired students. For example, Ms. Lester initiated peer collaboration as both an instructional strategy and a classroom management strategy. She explained that by placing students in small groups that were heterogeneous and included students of different abilities, she was able to include all students in the CS activities. Ms. Aldridge, similarly, encouraged peer modeling and paired students based on knowledge of their strengths and weaknesses (Teacher interview).

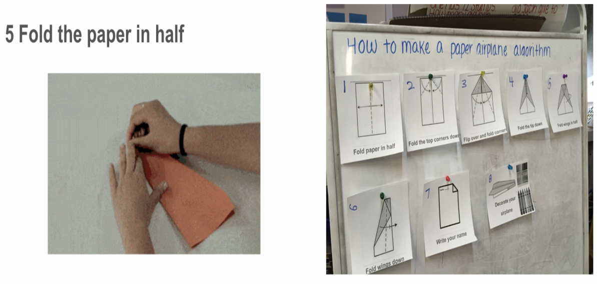
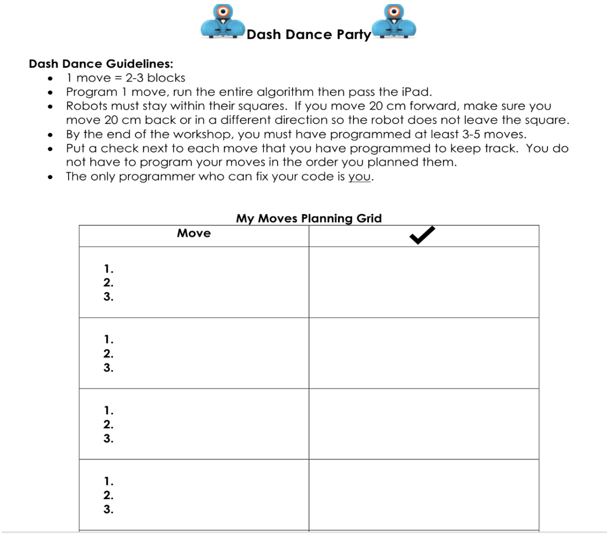


Figure 1: Example of Multiple Means of Representation.

4.1.2 Multiple means of representing content. : All the teachers recognized the need to present information to students through multiple pathways. Of the three representation UDL guidelines, the most common was providing options for perception (e.g., customizing the display of information) and options for comprehension (e.g., guiding information processing and visualization and maximizing transfer and generalizability). There was less focus on options for language and symbols such as clarifying syntax and structure, which is likely a result of greater emphasis on pictorial and auditory representations of content. Representation options often involved giving directions orally, providing visual cues, and modeling required tasks. For example, Ms. Rios explained that when she taught a CS unit using the *Hello Ruby* book series, she displayed the book on the interactive whiteboard and enlarged the text. Similarly, Ms. Lester taught the students about algorithms through the process of designing paper airplanes. Her lesson plan began with the following steps: (a) Tell students: We are going to learn an algorithm today for making paper airplanes; (b) Show students the paper airplane slides, (c) Demonstrate making a paper airplane under the document camera; (d) Put all the images up on the board in random order; (e) Label each step with a number from 1-6. Figure 1 provides images of both the demonstration of the paper airplane on the document camera and the steps required to create the paper airplane on the board, once they were placed in correct order.

Additionally, given the large number of ENL learners in the classrooms, teachers often talked about the need to provide multiple options for presenting content to address the needs of students' whose English language skills were still emerging. For example,



Dash Dance Party

Dash Dance Guidelines:

- 1 move = 2-3 blocks
- Program 1 move, run the entire algorithm then pass the iPad.
- Robots must stay within their squares. If you move 20 cm forward, make sure you move 20 cm back or in a different direction so the robot does not leave the square.
- By the end of the workshop, you must have programmed at least 3-5 moves.
- Put a check next to each move that you have programmed to keep track. You do not have to program your moves in the order you planned them.
- The only programmer who can fix your code is you.

My Moves Planning Grid

Move	✓
1. **2.** **3.**	
1. **2.** **3.**	
1. **2.** **3.**	
1. **2.** **3.**	

Figure 2: Student Checklist

Ms. Robinson explained, "There's a lot of translation. I use Google Translate often. There are visual models. There will be pictures of certain aspects of the CS project or a demonstration before they do it, so they can see the project."

4.1.3 Multiple means of action and expression. Of all the UDL principles, this principle had the least amount of evidence in the data. However, of the three guidelines within this UDL principle, all four teachers implemented checkpoints related to supporting students' executive function (e.g., guiding goal setting, supporting planning, and providing progress monitoring). The other two guidelines (options for physical action; options for expression and communication) were less prevalent.

First, all the teachers addressed executive function needs by emphasizing the use of rubrics and checklists for assessment, to provide students with guidelines and expectations for their computational artifacts, and to provide feedback to the students on their progress.

Ms. Lester, for example, wanted her students to program the Dash robots to dance. In her directions and accompanying handout, she defined the expectations and provided a checklist for the students (see Figure 2). In this checklist, for example, she wrote, "You must have programmed at least 3-5 moves. Put a check next to each move that you programmed to keep track. You do not have to program your moves in the order you planned them."

The second way that the teachers implemented this principle was by focusing on progress monitoring as a way to guide student progress. All four teachers enhanced capacity for progress monitoring by providing explicit and directed student feedback throughout the computational lessons. This feedback was often tied to the rubrics and checklists discussed above. This aspect was substantiated as check-ins with students, discussions of rubrics, and built-in assessment questions into the lesson plans.

Ms. Robinson indicated that she encouraged peer feedback as part of developing computational artifacts. She hoped to promote students' growth mindset among her students. To encourage effective peer feedback, she uses a peer feedback form with "stars" and "wishes" so that students can give peers positive feedback and suggestions. Along with these peer components, she also included a reflection, so that the students could make a plan for implementing the suggestions offered by their peers.

Finally, Rios discussed how she used the Dash robot with iPads to help students with social-emotional communication. She explained:

> "One child will say, 'This is my problem'[and use] the robot to communicate. We were shocked at how they were able to [use the robot to]say, 'Well, you're not my friend because you didn't play with me during lunch.' They were able to have back and forth conversations through the Robot."

In this way, students communicated their understanding through computing. In another example, Ms. Lester provided English language learners with sentence stems to talk about what they were doing during CS activities. She also encouraged students to act out code and use pseudocode to demonstrate understanding. She explained that in an unplugged activity where students filled grids using arrow pseudocode, "They practiced with their bodies. They got to design different arrow codes and then perform it with their bodies...they got to practice with dry erase markers and take turns in designing arrow code." Thus, both Ms. Rios and Ms. Lester found ways to increase the flexibility in which students demonstrated their understanding during computing activities.

4.2 UDL Implementation Differences

Although there were similarities in UDL implementation between teachers, there were also meaningful differences. Due to the difference in students in the classes, the teachers emphasized different aspects of UDL in their instruction. For example, although all teachers emphasized engagement, Ms. Lester, had the most even distribution of UDL practices across the three UDL principles. Interestingly, she also had the fewest instances of UDL checkpoints in her data (n=25), although she implemented the checkpoints that she utilized consistently throughout her instruction. The teachers with the largest number of evidence of UDL principles were Ms. Aldridge (n=42) and Ms. Robinson (n=48). Ms. Rios had 30 instances of UDL across the data. Again, this tabulation provided an initial analysis of UDL implementation but was much less useful than the qualitative analysis of *how* UDL was implemented. Below are an example activity each teacher implemented, anticipated barriers, and UDL practices to address those barriers.

Ms. Lester: Activity: Paper airplane algorithm activity. Anticipated learning barriers: Physically folding the paper accurately; Following either oral or pictorial directions alone; and High cognitive load in new and challenging activities. UDL approaches: (a) Options for pre-folded paper for all learners, (b) Modeling alongside posting pictorial representations, (c) Think-aloud stories about managing frustration, and (d)Heterogenous collaborative groups with expectations for peer support.

Ms. Rios: Activity: Robot stations. Anticipated learning barriers: Understanding abstract concepts; Attention with single-modality instructions; Completing complex multi-step activities. UDL approaches: (a) Station activities with paraeducator support; (b) Social-emotional learning built into CS activities, (c) Expression options based on student interest; and (d) Key vocabulary and concepts reviewed prior to station work.

Ms. Aldridge: Activity: Dash Robot *Around Town*. Anticipated learning barriers: Understanding implicit directions; Task completion with debugging code that does not work as intended; Maintained attention during complex multi-step activities; and Limited English language comprehension for bilingual students. UDL approaches: (a) Collaboration guidelines sentence stems provided and modeled; (b) At least two formats for directions (pictorial, auditory); (c) Team problem-solving protocols, (d) Flexible groupings based on student choice, and (e) Use of Google Translate.

Ms. Robinson: Activity: Open-ended Scratch. Anticipated learning barriers: Preference to always work independently; Discouraging comments by some students about peers' activities; Limited English language for bilingual students. UDL approaches: (a) Collaboration guidelines provided and modeled; (b) Pair programming implemented in homogeneous pairs (so both students fully participate); (c) Student choice provided; (d) Extension activities provided for all students; and (e) Model for students toggling Scratch blocks between home language and English.

There was no "one way" to implement UDL. Rather, implementation addressed the range of variability in the teachers' classrooms: For example, Ms. Lester and Ms. Robinson had ENL learners in their classrooms. Ms. Robinson and Ms. Aldridge also had students receiving gifted supports. Ms. Rios had students with more significant intellectual support needs. Therefore, even the same UDL checkpoints were actualized differently in different settings.

5 DISCUSSION

The study investigated how elementary teachers implemented UDL within academically diverse CS classrooms. Findings supported findings from other studies that investigated the inclusion of students with disabilities in CS education. That is, teachers attempted to support all learners including, students with disabilities, and attempted to meaningfully include all learners [5]; [15] [16]; [17]. However, in this study, as well as in Ray and colleagues (2018), doing so in the context of CS education required significant PD and support from a coach familiar with both CS education and UDL.

A major question answered by this study was how teachers used the UDL Framework to better engage academically diverse learners in their inclusive CS instruction. Prior to this study, only one study examined teacher implementation of UDL in this context. That study ([13]) showed that teachers interpreted the UDL framework narrowly. A major implication of that study was that the teachers required significantly more support to implement UDL-based CS instruction. This current study attempted to minimize the likelihood of attaining similar results by providing teachers with PD and coaching in using the UDL framework both in planning and in implementation. This study revealed that the teachers generally taught in a manner consistent with the UDL framework. However, similar to Ray et al., (2018), when looking at which components of UDL the teachers implemented, some areas were emphasized above others. Specifically, among the three UDL principles, multiple means of action and expression was emphasized to a lesser extent than multiple means of engagement or multiple means of representation. Second, UDL looked different across the classrooms as teachers made decisions about the needs of their students. The three UDL principles (broken into 9 guidelines and accompanying checkpoints) address different student needs. Hence, when looking at UDL implementation, teachers had examine the goals for the CS instruction, barriers to achieving those goals, and which guidelines and checkpoints might address those barriers to learning. Simply put, the UDL checkpoints were not intended to hold equal weight as they are context- and student-dependent. There is no known "effective proportion" of the three UDL principles and the guidelines and checkpoints within them. Teachers should look at the goals for their CS instruction, possible barriers that students may face in meeting those goals as well as strengths that the students bring to the CS classroom, and then choose the checkpoints that could leverage students' strengths and minimize barriers to learning. Lastly, although UDL was implemented in the context of CS instruction, the implementation of UDL did not look significantly different than implementation in other content areas. There was no CS-specific implementation strategies. Rather, teachers applied UDL based on an understading of their learners and the goals they had for their instruction. They most often attempted to (a) make CS content relevant to their learners, and (b) present complex concepts and academic language to their learners in multiple ways.

5.1 Limitations and Future Research

Findings from this study must be interpreted in light of several limitations. The primary limitation was the sample size. Although this study investigated teachers with a broad range of classroom contexts (self-contained special education, inclusive co-taught classrooms, and classrooms with a high number of students who were ENL learners), there were only four teachers. Additionally, teacher characteristics such as number of years teaching and licensure were not studied as part of the analysis. Future research should investigate UDL implementation in a broader range of classrooms and also investigate to what extent teacher background and licensure might influence the way they implement UDL within their CS classrooms. A second limitation of the study was the limited focus on the instructional coaches that helped the teachers implement UDL in their classrooms. Because instructional coaching was the primary mechanism to help support the teachers in implementing UDL in their classrooms, future research should investigate the experiences of the coaches alongside the teachers.

REFERENCES

[1] James D Basham and Matthew T Marino. 2013. Understanding STEM education and supporting students through universal design for learning. *Teaching Exceptional Children* 45, 4 (2013), 8–15.

[2] Sheryl Burgstahler and Rosalie J Russo-Gleicher. 2015. Applying universal design to address the needs of postsecondary students on the autism spectrum. *Journal of Postsecondary Education and Disability* 28, 2 (2015), 199–212.

[3] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

[4] Center for Applied Special Technology. 2011. Universal Design for Learning guidelines version 2.0.

[5] Alexandria K Hansen, Eric R Hansen, Hilary A Dwyer, Danielle B Harlow, and Diana Franklin. 2016. Differentiating for Diversity: Using Universal Design for

Learning in Elementary Computer Science Education. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. ACM, 376–381.

[6] Maya Israel, Jamie N Pearson, Tanya Tapia, Quentin M Wherfel, and George Reese. 2015. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education* 82 (2015), 263–279.

[7] Margaret E King-Sears, Todd M Johnson, Sheri Berkeley, Margaret P Weiss, Erin E Peters-Burton, Anya S Evmenova, Anna Menditto, and Jennifer C Hursh. 2015. An exploratory study of universal design for teaching chemistry to students with and without disabilities. *Learning Disability Quarterly* 38, 2 (2015), 84–96.

[8] Richard E Ladner and Maya Israel. 2016. For all in computer science for all. *Commun. ACM* 59, 9 (2016), 26–28.

[9] Zuzanna Lechelt, Yvonne Rogers, Nicola Yuill, Lena Nagl, Grazia Ragone, and Nicolai Marquardt. 2018. Inclusive Computing in Special Needs Classrooms: Designing for All. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 517.

[10] Sharan B Merriam. 2002. *Qualitative research in practice: Examples for discussion and analysis*. Jossey-Bass Inc Pub.

[11] National Center for Education Statistics. 2017. Children and youth with disabilities. https://nces.ed.gov/programs/coe/indicator{_}cgg.asp

[12] Kavita Rao. 2015. Universal design for learning and multimedia technology: Supporting culturally and linguistically diverse students. *Journal of Educational Multimedia and Hypermedia* 24, 2 (2015), 121–137.

[13] Meg J. Ray, Maya Israel, Chung eun Lee, and Virginie Do. 2018. A Cross-Case Analysis of Instructional Strategies to Support Participation of K-8 Students with Disabilities in CS for All. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM, New York, NY, USA, 900–905. https://doi.org/10.1145/3159450.3159482

[14] David H Rose and Anne Meyer. 2002. *Teaching every student in the digital age: Universal design for learning*. ERIC.

[15] Melinda R Snodgrass, Maya Israel, and George C Reese. 2016. Instructional supports for students with disabilities in K-5 computing: Findings from a cross-case analysis. *Computers & Education* 100 (2016), 1–17.

[16] Matthew S Taylor. 2018. Computer Programming With PreK-1st Grade Students With Intellectual Disabilities. *The Journal of Special Education* (2018), 0022466918761120.

[17] Sarah Wille, Jeanne Century, and Miriam Pike. 2017. Exploratory Research to Expand Opportunities in Computer Science for Students with Learning Differences. *Computing in Science & Engineering* 19, 3 (2017), 40–50.