Chapter 12

Leveraging Multilingual Identities in Computer Science Education

Sharin Jacob, University of California Irvine, USA

Leiny Garcia, University of California, Irvine, USA

Mark Warschauer, University of California, Irvine, USA

INTRODUCTION

In the last decade, considerable effort in the US has been dedicated to promoting equity in computer science education by broadening participation of students from marginalized and culturally diverse backgrounds. However, scholarship on promoting quality computer science instruction for linguistically diverse students is sparse (Jacob et al., 2018). As one of the fastest growing populations in US schools, language learners remain dramatically underrepresented in computer courses and careers (Martin, McAlear, & Scott 2015). This is likely due to lack of access to courses (Martin & McAlear, 2015), lack of diversity in the workforce (Royal & Swift, 2016), lack of representation in the media (Royal & Swift, 2016), and/or pervasive stereotyping in the field (Margolis, 2010). Given this limited experience and lack of relatable role models, students from diverse backgrounds may perceive members of the computer science discipline as being unlike themselves and lose interest in the field (Aish, Asare, & Miskioglu, 2018). This qualitative study explores how multilingual students leverage their identities during computer science instruction to support the development of computational thinking practices and promote interest in computing. Results from student semi-structured interviews indicate that identity expression supports positive social and academic outcomes for multilingual students in computing.

Background

Despite economic forecasts projecting a shortage of qualified candidates for computing jobs (BLS, 2015), underrepresentation of students from marginalized groups in these fields continues to be a critical issue (NSF, 2017). Underrepresentation in computer science is particularly alarming for students from linguistically diverse backgrounds. Schools, with a student population that comprises 11% or more English learners of the total student body, offer half as many computer science courses as schools serving fewer than 11% English learners (Martin & McAlear, 2015). In California, the site of this study, Latinos represent 54% of California K-12 enrollment (Ed Data, 2018), yet they made up only 22% of the students who took the Advanced Placement Computer Science Exam in the state (College Board, 2017).

Concerted effort has been dedicated to exploring the causes of such underrepresentations in computer science in the United States. Students from low socioeconomic status backgrounds have lower levels of computer and internet access (McFarland et al., 2017) and lack exposure to professionals in computer science fields (Royal & Swift, 2016). Limited access and exposure are exacerbated by inadequate media representation of diverse populations in computer science (Royal & Swift, 2016) and an overall dearth of student experiences related to computer science both at home and at school (Google & Gallup, 2015; Wang, Hong, Ravitz, & Moghadam, 2016). Given this limited experience and lack of relatable role models, students from diverse backgrounds may perceive members of the computer science discipline as being unlike themselves and thus lose interest in the field (Aish, Asare, & Miskioglu, 2018). Research findings indicate that increasing identification with computer science disciplines leads to increased persistence in pursuing computer science degrees (Mercier, Barron, & O'connor, 2006; Packard & Wong 1999). Efforts targeting marginalized students to promote identification with computer science have been laudable and numerous. While these efforts have typically targeted underrepresented groups such as women and students from diverse racial backgrounds, scant attention has been given to exploring approaches to computer science education that attract and retain multilingual students (see Jacob et al., 2018).

In addition to lack of representation, language learners face pervasive stereotyping in the field, which sends exclusionary messages, particularly with respect to who 'does' computer science. This stereotype is based on a fixed mindset, in which certain types of individuals, often those from privileged backgrounds, are endowed with particular talents and abilities (Margolis, 2010). Such inflexible thinking leads to the setting of expectations based on group characteristics, in which lower expectations limit the opportunities for participation and growth of students who do not fit these stereotypes. To address these issues within computer science education, initiatives such as *Computer Science for All* (Smith, 2016) recognize the systemic cultural and historical inequities that lead to differential learning opportunities for underrepresented students. These movements address systemic inequities by maintaining high expectations for all students regardless of geography, and for lessening inequality related to language, race, ethnicity and socioeconomic status. Equity-minded educators combat deficit-based views by embracing diversity and viewing students' backgrounds as repositories of knowledge that contribute to and enrich learning.

Conceptual framework for characterizing multilingual identities

In underscoring multiple modes of cultural identification, Norton (1997) made significant contributions to SLA research by challenging the dichotomous relationship underlying assimilation based models of identity. Drawing from the work of Pierre Bourdieu, Norton (1997) proposed the idea of Investment as a means for attaining 'symbolic capital' within a given institutional setting, such as classrooms. Multilingual students use language not only to express themselves but to position themselves within the world. Speech is a form of performativity in which these students navigate existing power relations, as well as subsequent modes of inclusion and exclusion (Norton, 2013). Both investment and resistance to cultural norms, then, exemplify the struggle for power undertaken by language learners who choose to obtain capital within specific social, cultural, and historical contexts. Because identities are not dichotomous—they are constantly shifting and often contradictory—language learners traverse multiple spaces as they navigate their relationship in the world.

The ability to fluidly accommodate multiple identities is a uniquely valuable characteristic of digital spaces that benefits students from culturally and linguistically diverse backgrounds. The advent of the internet, social media and instant messaging have allowed for myriad forms of communication that transcend temporal and spatial boundaries (Warschauer, 1999). These spaces have opened new possibilities for language learning, fostering interaction in multiple languages (Warschauer, 2009) that traverse linguistic and cultural norms (Canagarajah, 2013). Given these opportunities, educational technologies broaden the ways in which students identify with the curriculum and present marginalized students with more opportunities to leverage knowledge learned outside of school to facilitate formal learning.

While digital spaces have transformed the ways in which individuals negotiate their identities, scant attention has been paid to identity construction through the creation and development of these digital spaces via computer programming. Despite this paucity of research, computer science continues to shape the digital terrain in which we find ourselves immersed. Given the rise of automation, artificial intelligence, and machine learning in our daily lives, computing is increasingly being used to solve everyday problems. In fact, thinking computationally is so embedded within social patterns of human communication and interaction that it arguably represents a fundamental literacy (diSessa, 2000) and can be leveraged to develop literacy (Jacob et al., 2018, Jacob & Warschauer, 2018; Peppler & Warschauer, 2012). Engaging language learners in using computational approaches to express their identities will prepare them for full participation in society while increasing career opportunities and enhancing potential for civic engagement.

What is computational thinking and why should we teach it to our students?

In her groundbreaking discussion in the *Journal of the Association for Computing Machinery*, Jeanette Wing (2006) coined the term computational thinking and described its significance in solving complex problems that extend across a broad array of disciplines. Computational thinking utilizes concepts that are essential to computing to systematically characterize problems, process information, and test solutions. By formulating thoughts and questions in a manner that can be interpreted by a computer, computational thinkers are able to instruct computers to carry out sets of tasks in a specified order to achieve desired results (Wing, 2006). While the exact definition of computational thinking is still a matter of debate (Barr & Stephenson, 2011; Grover & Pea, 2013), ISTE (2011) identifies algorithmic thinking, automation, abstraction, modularization and data analysis as being essential skills to carry out computational processes. Computational thinking is not to be confused with computer programming or coding; the latter involves the teaching and learning of computer programming languages. Rather, computational thinking is typically operationalized through computer programming but can be learned with or without the use of computers, the latter approach often dubbed as “unplugged” activities in computer science education.

To date, there is no consensus on the exact definition of computational thinking in the scholarly research (Barr & Stephenson, 2011; Grover, Pea, 2013). Some researchers define computational thinking according to key computational concepts involved (i.e., events, conditionals, loops, etc.) (Grover & Pea, 2013). Brennan and Resnick (2012) argue that focusing solely on concepts represents a limited perspective of computational thinking, proposing that computational thinking practices and perspectives provide a more robust understanding of the skill. These practices have been examined with children programming in Scratch, a media rich, block-based programming environment designed to engage novice programmers in coding (Resnick et al., 2009). This chapter borrows from Brennan and Resnick (2012) in examining student computational practices along four dimensions: 1) experimenting and iterating, 2) abstracting and modularizing, 3) testing and debugging and 4) reusing and remixing.

Each of these four practices is intended to examine the design process that students engage in as they develop their computational artifacts (Brennan & Resnick, 2012).

- The practice of experimenting and iterating involves viewing project design as a process that involves designing, developing, and testing algorithms in an iterative manner—after testing steps students revisit their programs to further develop their projects based on new experiences and ideas. Creativity is integral to this practice as students use their imaginations to design and refine their projects.
- Abstracting and modularizing involves the process of decomposition, or breaking a problem down into smaller, more manageable pieces. Abstraction and

modularization is typically used during the initial conception of a project as well as when organizing and arranging stacks of code to synchronize design features.

- Testing and debugging entails dealing with problems in code by identifying causes and finding solutions. In general, students who use Scratch learn debugging through trial and error, from their experience with other projects, and by receiving help from others (Brennan and Resnick, 2012).
- Reusing and remixing involves building on one another's' code and designs to create new, and potentially more complex projects. With it brings the question of how to give others credit for their work, which represents an understudied area.

Current Study

Through funding from the National Science Foundation, we have formed a collaborative network of university, K-12 researchers and practitioners to promote computational thinking for students in grades 3-5 in a way that better meets the needs of language learners. CONECTAR (*Collaborative Network of Educators for Computational Thinking for All Research*), represents the first phase of a larger effort aimed at validating and disseminating instructional materials for broader implementation and assessment. Phase one is exploratory in nature and seeks to iteratively develop and test instructional materials and data collection instruments in upper elementary classrooms to better meet the needs of multilingual students. The work is situated in a southern California school district, where the majority of students are low-income (89.7%), Latino/a (93%), English learners (overall 41% but higher in elementary grades at 62.7%). The goals of this project are:

1) To investigate the teaching and learning of computational thinking in grades 3-5 in the districts and nationally, and their potential for successful engagement of diverse learners.
2) Based on this, to develop and pilot instructional materials that align with the Common Core State Standards—specifically related to English Language Arts—that best meet the needs of the predominantly Latino/a language learners.
3) To iteratively pilot test these instructional materials for broader implementation and assessment.
4) To establish a successful *Researcher Practitioner Partnership* (RPP) that can ultimately serve as a model for other RPPs in support of *Computer Science for All*.

This chapter is part of a larger study that uses principles of Design-Based Implementation Research (DBIR) to study, design, implement and refine a computational thinking curriculum developed in collaboration with researchers, teachers, administrators and curriculum specialists. The study utilizes qualitative methods to explore how multilingual students leverage their identities while engaging in the computing curriculum. In addition, the study explores how students' identities can be leveraged to support the development computational thinking practices. Data collection includes semi-structured interviews that ask students to analyze their *About Me* Scratch projects. The *About Me* project is geared towards self-expression: Students are encouraged to share facts about themselves with their peers in Scratch. The interview engages students in analyzing their projects according to four computational thinking practices: 1) experimenting and iterating, 2) abstracting and modularizing, 3) testing and debugging, and 4) reusing and remixing (Brennan & Resnick, 2012).

Research questions

Under the umbrella of these broader aims, we ask the following research questions:

- How do multilingual students leverage their identities while engaging in computing and computational thinking?
- How does multilingual identity expression support the development of computational thinking practices?

METHODOLOGY

Sampling procedures

This study took place in six upper elementary classrooms (3rd through 5th grade) across the Santa Ana Unified School District (SAUSD). Santa Ana Unified School District, with about 56,000 students is the seventh largest school district in California, and one of the districts in the United States with the highest percentage of Latino/as (93%), low-income learners (89.7% receiving free or reduced-price lunch), and 41% language learners (62.7% in the elementary grades). SAUSD is seeking to improve student academic achievement and interest in STEM through programs that support instructor innovation and emphasize integration of STEM and English language arts curricula.

Teachers were selected based on their experience teaching computer science to upper elementary students. In the larger study, maximum variation sampling was used to select 28 students (four from each classroom) to be interviewed based on differences in programming experience and linguistic proficiency. Of these 28 students, a convenience sample of eight students were selected for this chapter to explore the relationship between identity and computational thinking for multilingual students.

Instruments: Semi-structured interviews

Semi-structured interviews were adapted from artifact-based interviews developed by the ScratchEd team at Harvard's Graduate School of Education. They were adapted from a rubric that assesses students' fluency with computational thinking practices, including experimenting and iterating, testing and debugging, reusing and remixing, and abstracting and modularizing. Open-ended questions that evoke a more conversational tone were modified from the existing rubric to capture more in depth data on student experiences while developing their computational thinking artifacts.

Procedures

The semi-structured student interviews were administered to eight students after they finished their *About Me* Projects. A team of four researchers were trained to ask open ended questions and make active decisions about when to probe or ask follow-up questions, shift topics, and/or modify the original protocol to accommodate new themes. Researchers used an interview protocol to conduct the semi-structured interviews, and audio-recorded student responses. The researchers also video-recorded students' screens as they were demonstrating their project to gain a better understanding of the block-based commands that corresponded to students' verbal responses. Audio-recordings were transcribed using Temi software and again by a team of researchers. MAXQDA was used to code the interviews and conduct reliability checks.

Data analysis

RQ1: How do multilingual students leverage their identities while engaging in computing*?*

We used open and axial coding (Saldana, 2016) to identify how students leveraged their identities to engage in computing and computational thinking. Our coding scheme was guided by existing theory on identity development (Norton, 1997, Ricento, 2005; Wortham, 2006) and inductively from student responses as they pertained to the research question. The coding scheme was formulated initially by the first author. Weekly meetings were conducted with the second author to and iteratively refine the codes and categories. Through multiple iterations, a codebook was devised to answer our first research question. The coding scheme addressed how students leveraged their identities to engage in designing their computational artifacts.

RQ2: How does multilingual identity expression support the development of computational thinking practices?

The semi-structured interviews were examined using deductive and inductive qualitative coding (Saldana, 2016). The generation of first cycle codes in this study is situated within a procedural, deductive, frame of analysis (Boyatzis, 1998; Cummins, Hu, Markus, Montero, 2015; Schensul, LeCompte, Nastasi, & Borgatti, 1999). In this approach, protocol coding was used to discover how students engaged in computational thinking practices when developing their *About Me* projects. This process consisted of reading the data multiple times to categorize computational thinking practices and subcategories within each practice. The Harvard ScratchEd Student Artifact analysis rubric was used to guide the deductive coding. Student responses were categorized according to their reported use of four computational thinking practices: 1) experimenting and iterating, 2) abstracting and modularizing, 3) testing and debugging, and 4) reusing and remixing. Because the goal of analysis was exploratory, that is, to identify emergent themes in participants' responses about their computational thinking practices, researchers also generated new codes in the process when the existing codes from the rubric did not fit with particular excerpts of text. Credibility of qualitative findings was ensured through triangulation, member checks, and negative case analysis (Lincoln & Guba, 1985).

FINDINGS

The nature of the *About Me* project allowed for students define their identities by expressing multiple experiences, interests, and characteristics across a variety of contexts. Specifically, students mentioned hobbies and favorite items or activities that they enjoy as well as stories about their lives, families, and communities. The multiple modes of identification were translated and represented on the Scratch site through the customizable features of the program; sprites (created animated caricatures), backgrounds and sounds. An example can be seen with students like Ricardo, Abby, and Eduardo (pseudonyms). First we have Ricardo who decided to display his various interests for his About Me project, which include playing basketball, video games and enjoyment of mathematics:

> Ricardo: "So I decided like, what sprites is like, I added like the things that I liked, like my favorite um thing is basketball, so added a basketball player [inaudible] and like a little thing. And then since my favorite subject is math I added a plus sign, a division, and a minuses right there and I added a controller for like the gaming part."

We can see that Ricardo chose sprites that he felt would directly represent his interests. Most of his sprites were provided by the Scratch library, however, he also designed his own sprite to resemble a gaming console and indicate his love for video games. In addition to Ricardo's method of identity expression, students also remade an experience they associated themselves with. Below we have Abby whose identity piece involved the recreation of a party event:

> Abby: " I want it to make [the project] a little bit extreme. So for the background I just put like a little party because I love going to parties and then [the background] was perfect because it had a bunch of balloons and I was wondering how I can make it like even funner than it looked. So then, um, I got some color changing blocks and I was, just want like an adventure for the blocks. [...] [The code block] is the change color effect by 25 [...] then I went to my events and under 'Control' and I got a 'Repeat' and 'Forever' block and, and um, I wanted it to last forever, but then I did the 'When flag is clicked.' It's opposed to do a rainbow background … "

While Abby ultimately incorporated sprites that directly symbolized some of her characteristics, she also attempted to recreate her interests in fun party experiences for her project. To her, having a background that shifts between all colors, or in other words a rainbow background, in a continuous manner symbolized her conception of a lively event. Because of her desire to express this enjoyable experience, Abby incorporated computational concepts such as events and loops to create this special color effect.

Similar to Abby, Eduardo added additional elements to his project to provide a full representation of his identity. Here we have him explain his efforts to set a certain tone for his project as he describes his favorite weekend activities with his siblings.

> Eduardo: “I think it would have been like a little bit like fun and like a little bit of smooth, like, and not too much [...] like just like a pretty like just smooth [...] I didn't know you can actually play music until I saw the sound and I clicked on it and there was like buttons and all that and I didn't know what they did at first. I just started exploring. And then I heard, I saw like this little button, like I saw all these and I didn't know what they did. So, um, I went to sound cause I saw them and then I clicked one of these buttons and then I clicked on here and there was a bunch of sounds and it clicked one that was my favorite and I clicked on it. And then once I got it I went to scripts and sounds and then I saw which one was the sound. For example, ‘play sound’, ‘chill out,’ dance ‘chill out.’ And then I'm put it in my blocks.”

In the efforts to have a project that truly portrays his experiences, which in this case included the use of music, Eduardo was intrinsically motivated to explore coding blocks that he never tried before. While he focused on finding the right music to set a “smooth” tone, he practiced the skill of modifying coding parameters and incorporating code it into a larger project.

Despite the different approaches in expression, Ricardo, Abby, and Eduardo focused on their chosen forms of identity which then informed the coding functions they deemed important for any user to understand their lives. As their coding experiences were guided by their chosen forms of identity, students leveraged their identities to engage in four key computational thinking practices:

Experimenting and iterating

Students recalled instances where they discovered, explored, and tinkered with coding blocks as they created functions for specific elements of their projects. Here we have Melanie, who shared her process in programming a sprite to display information about her interests.

> Melanie: "Yeah. So I was on this sprite. And I have my things but I didn't know that you had to- when you change the sprite you had to change um this- like right here you had to change them right here. So what I did is I just added more blocks onto this [code blocks] and just kept on doing it and I'm like, why is this doing this? And then I noticed this, and that's how I figured out how to do this. And then I did like 'Basketball is my favorite sport' because I love basketball. I said 'Llamas are my favorite animals' and 'My favorite book is The Land of Stories: The Enchantress Returns.' And 'My home is cozy.' And 'My favorite food is spaghetti.' "

Melanie experimented with coding blocks by including them in her program and looking for any potential outcomes of her program. In her case, she wondered how despite the inclusion of blocks a certain action was still occurring which led her to wonder about "changes" in parameters that she might need to control. As a result, she had a program where her sprites can display speech bubbles to "say" corresponding facets of her identity. Similarly, Omar experimented with his code by understanding the outcomes as he attached or detached certain coding blocks in his program. Below he discussed how he understood the importance of event blocks while trying to make his sprite "glide" on his screen.

> Omar: "Yeah, so once, I used [event blocks], but then I took them off and just left like the 'Glide' block [...] Um [the code blocks] didn't work and they like - they had these things [event blocks] on, but then I took them [event blocks] off and I left like these

> blocks alone and then I clicked the flag, but nothing happened [...] [Event blocks] - they like really do stuff or you just click on it and it happens."

Although Omar was using an event block at the start of his sub program designed to make his sprite glide, he did not fully understand its purpose. It was not until he was attempting to debug his code to execute the gliding function that he experimented with the blocks and realized how Event blocks fully work.

Abstracting and modularizing

As the purpose of the About Me project was to engage students in self-expression, each element of their identity that they chose to express contributed to students' thinking of the main functionalities and sub programs that would ultimately inform users of their identities. Eduardo's project revolved around a description of his relationships, and so he decompartialized his project as such:

> Eduardo: "I organized [my script] by putting, I'm not like in sections but like one by one, like about my family then about me. And then about my friends, so like one, two, three. Like first about my family's second about me, then my brothers and then my friend."

Many students used facets of their identity to practice abstraction or to decide which components to include in their projects and which to leave out. Students then identified how to express these elements in a sequential fashion by modularizing their code. In Abby's case, she incorporated her initials in her project through the provided sprites Block-a and Block-o.

> Abby: "So I looked at one of the letters and I was thinking to myself what can I make

with these letters? So then I clicked on one of the letters and I wanted it to turn because it was an O, and like, I was thinking of like various kinds of [movements] and then I clicked one of [the movement blocks] and I knew that turn 15 degrees and the other way turn 15 degrees would make like a circle and it would turn around and I wanted it to turn it around forever. So I clicked the 'Forever', forever block and the 'Repeat.'"

Students also repeated their modularized programs for some elements of the projects to help them express other elements. Here we have Marcos who expressed his identity using direct representation of Sprites, which included dogs, food and his brothers.

Marcos: "[...] I did the same thing [for a sprite] like the dog [sprite]; when, I put press D and then right here [in the script] says "When D clicked" to let it say [my interests] over here [in the stage] …"

Marcos created a sub program that had users press a specific letter key in order to activate a sprite to explain a specific part of his identity. He decided to incorporate this same sub program that he initially created for his dog sprite and used it to explain his other interests.

Testing and debugging

With their end goal in mind, students debugged towards achieving a level of functionality that allows for identity expression. Students shared debugging strategies that they implemented.

Marcos: [To make the sprite spin around] I tried "Move 10 Steps." Then I put the-the "Repeat Forever" 10 Steps. But it didn't work.

> Camila: “So with the background, I wanted it to play sound. So I clicked all of these together, and with one of [the sound blocks] and with one of [the change color blocks] together [the program] started to glitch. When the sound played in the middle [of the program], it start to glitch. And [the sprite] would turn a little bit grey, and when I took [the sound blocks] out and put them in this form it started to work [...] It probably needed like another [event] block to like get it because probably one wasn't enough. ”

Students generally debugged their programs via trial and error, revealing the experimental and iterative nature of their programming experience.

Reusing and remixing

While there are many existing online resources for users, students revealed that their main source of help were their peers. They reported instances in which their peers inspired their design choices or helped them fix pieces of their code.

> Melanie: “I looked at my friends so my friend Andrea's, and it gave me ideas on to make it moving and to, to make it move. The llama. And I'm like, oh I need to move it because it would just be that and I didn't really want that. I wanted to make it more special. So that's when I decided to make the llama move. It wasn't really my project that I created that gave me ideas for this, but it was my friends.”
>
> Abby: “Well, I didn't know how to do that, so I asked my friend, he showed me how to do that. and it was with with like when space key press, and I got one of those, and he told me to use whichever, like the up Arrow or the J or any letters or numbers on it and I

wanted and I wanted the letter J, and because it's right here in my name. And when that happened it would say my nickname is JoJo."

Outside of peer support, some students still showed use of external online resources such as tutorials and videos. An example here is Camila who watched tutorial videos and explored existing projects through Scratch's library to gather ideas on how to make her project interactive. In an effort to make her fruit salad, a symbol of her love for eating fruit, she watched "a video about Scratch and it showed how to swirl the letters and I tried the fruit salad on that." Additionally, she selected a sprite that represented herself, and decided to make it interactive by having it move with arrow keys.

Camila: "Because um I was going to try and move it to another direction (interviewer: mmmm) because I wanted to click these two (presses on left and right arrow keys), but it wouldn't work, it would just go (points towards the right direction). I tried to um use different events for it could move to another direction, but it wouldn't really work."

Although these actions do not directly help her specifically express her identity, the motivation to make her project interactive led to the reuse and remixing of code.

Additional findings

Aside from personal experiences informing their learning of computational concepts, students established personal expectations and goals for their project that were influenced by the target audience - their peers.

When asking Marcos what he wanted the project to say about himself, he further emphasized how he wanted to "[...] do my stuff good and my friends can like it." He acknowledged that many design decisions was inspired by his desire to create a program that his peers would appreciate. He said he would "try blocks that I never tried to see what it would do. Then I realized all of the blocks to do…(I: Ok, nice)... To make my thing creative." This external motivation for the project was expressed by other students as well, like Eduardo who, "just wanted [the project] ... like as fun as like, like when people see it, like from one of my classmates here to just make it like, as like, as fun as I can be and that's it."

Despite having a focus on expressing their self defined identity, students still considered the user experience and presentation of the project, like the student Ana below.

Ana: "For the Giga I put 'When the flag is clicked' [code block]."

Interviewer: "Why did you do that?"

Ana: "[...] When you present [the program], there's usually a flag in the middle and I would find it like a little weird if it like didn't say to click on the Giga, and I didn't think of putting that so when flag clicked-when the flag is clicked it would probably work well."

DISCUSSION AND CONCLUSIONS

The *About Me* Scratch project appeared to produce positive social and academic outcomes for these multilingual students. These positive outcomes can be attributed to the bridging of out of school and in school learning environments by drawing on students' identities to create a 'third space' for student participation. They disrupt the traditional roles between students and teachers to create more horizontal, symmetrical spaces in which teachers and students co-navigate their official and informal identities. In doing so, students begin to see the parallels between their existing skills and the curricula, which not only engages them in computational thinking practices, but likely serves to increase their identification with the discipline.

Under an Acculturation model of identity development, students' rich linguistic and conceptual backgrounds are viewed as obstacles to overcome as they assimilate to formal learning environments. On the other hand, when identities are viewed as multiple and fluid, students are encouraged to utilize their home and school-based resources interchangeably to navigate academic and technical spaces. Instructional materials that draw upon students' identities facilitate parallels between academic content and what students already know, thereby increasing their opportunities to identify with the curriculum as well as communicating their own identities through the development of their projects.

Considerable effort has been dedicated to leveraging students' existing resources to increase identification with STEM fields. These efforts include drawing from students' existing family expertise to promote science and engineering practices Kafai, Searle, Martinez, & Brayboy, 2014), diversifying the curriculum through the simulation of multicultural artifacts (Eglash, Gilbert, and Foster, 2013), and engaging students in action based initiatives that seek to solve problems within their local communities (Goldman et al., 2009). As we have seen with the computational thinking curriculum described in this study, drawing upon students identities contributes to deeper learning and enriches and broadens the discursive practices students utilize to engage with the curriculum.

Beyond leveraging students' identities, the programming environment Scratch has the unique capacity for providing language neutral spaces for language learners (Tofel-Grehl et al., 2016). In addition to translating features, the multimodal nature of the interfaces reduces the linguistic load for language learners and enables them to more readily access content. Thus, computational thinking practices have the potential to represent an equalizer for linguistically diverse students, providing a variety of entry points for student participation that leverage, instead of tax, their existing resources (tofel-Grehl et al., 2016).

Although, little research has focused on leveraging students' *linguistic* resources to promote computing and computational thinking for language learners, from our interviews, we see that students use their everyday sense making capabilities to describe their projects. To this end, students' discourse choices reveal a lack of academic and discipline-specific language. While theoretical frameworks for exploring the relationship between computational thinking and literacy have been established (Jacob & Warschauer, 2018), scant attention has been paid to promoting discipline-specific literacy in the field of computer science. Future research should focus on empirical studies that examine how students existing resources can be leveraged to promote the development of academic literacy as well as the language forms and functions of computer science (see Jacob et al., 2018).

By engaging students in using their existing tools to access content while infusing content with artifacts and themes from their lives, responsive computing education sends an alternative and compelling message regarding who 'does' computer science. This has the potential to shift teacher and student biases to engender a more inclusive approach to computer science education. The perception of exclusivity in computer science runs in stark contrast to the pervasiveness of computing and computational thinking within society and the workforce. If computational thinking is indeed a fundamental literacy required for full participation in society, then the myth that only certain, prodigious talents can 'do' computer science needs to be dispelled. This can be achieved by integrating culturally responsive computing education into existing K-12 curricula that leverages students' identities to engage them in computational thinking. When students are able to utilize their existing skills to engage in computer science, they develop a sense of competence and ultimately the belief that they can succeed as professionals in the field.

REFERENCES

Aish, N., Asare, P., & Miskioğlu, E. E. (2018, March). People Like Me: Providing relatable and realistic role models for underrepresented minorities in STEM to increase their motivation and likelihood of success. Paper presented at the *2018 IEEE Integrated STEM Education Conference (ISEC)*, 83-89, Princeton, NJ.

Amaral, O. M., Garrison, L., & Klentschy, M. (2002). Helping English learners increase achievement through inquiry-based science instruction. *Bilingual research journal*, *26*(2), 213-239.

Aspray, W. (2016). Participation in computing: The National Science Foundation's expansionary programs. New York, NY: Springer.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48-54.

Boyatzis, R. E. (1998). Transforming qualitative information: Thematic analysis and code development. London, UK: Sage.

Bureau of Labor Statistics. (2015). Computer and Information Technology. Occupational Outlook Handbook, 2014-2015. Retrieved from: https://www.bls.gov/ooh/computer-and-information-technology/home.htm

Canagarajah, A. S. (2013). Negotiating translingual literacy: An enactment. *Research in the Teaching of English*, *48*(1), 40-67.

College Board (2017). AP program participation and performance data 2017. [Data file]. Retrieved from https://research.collegeboard.org/programs/ap/data/participation/ap-2017.

Cummins, J., Hu, S., Markus, P., & Montero, M. K. (2015). Identity texts and academic achievement: Connecting the dots in multilingual school contexts. *TESOL Quarterly*, *49*(3), 555-581.

diSessa, A. (2000). *Changing minds: Computers, learning and literacy.* Cambridge, MA: MIT Press.

Ed Data (2018). California public schools. [Data file]. Retrieved from https://www.ed-data.org/state/CA.

Eglash, R., Gilbert, J. E., & Foster, E. (2013). Broadening Participation: Toward culturally responsive computing education. *Communications of the ACM*, *56*(7), 33-36.

Goldman, J., Shilton, K., Burke, J., Estrin, D., Hansen, M., Ramanathan, N., ... & West, R. (2009). Participatory Sensing: A citizen-powered approach to illuminating the patterns that shape our world. *Foresight & Governance Project, White Paper*, 1-15.

Google & Gallup (2015). *Images of computer science: Perceptions among students, parents and educators in the U.S.* Retrieved from https://services.google.com/fh/files/misc/images-of-computer-science-report.pdf

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38-43.

International Society for Technology in Education (ISTE), & Computer Science Teachers Association (CSTA). (2011). Operational definition of computational thinking for K-12 education. Retrieved from http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf

Jacob, S., Nguyen, H., Tofel-Grehl, C., Richardson, D., & Warschauer, M. (2018). Teaching computational thinking to English learners. *NYS TESOL Journal, 5*(2), 12-24.

Jacob, S., & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Computer Science Integration, 1*(1), 1-19.

Kafai, Y., Searle, K., Martinez, C., & Brayboy, B. (2014, March). Ethnocomputing with electronic textiles: Culturally responsive open design to broaden participation in computing in American Indian youth and communities. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 241-246). ACM.

LeCompte, M. D., Schensul, J. J., Nastasi, B. K., & Borgatti, S. P. (1999). Enhanced ethnographic methods: Audiovisual techniques, focused group interviews, and elicitation. New York, NY: Altamira Press.

Lincoln, Y. S., & Guba, E. G. (1985). Naturalistic inquiry (Vol. 75). London, UK: Sage.

Lynch, M., & Macbeth, D. (1998). Demonstrating physics lessons. In J. G. Greeno & S. V. Goldman (Eds.), *Thinking practices in mathematics and science learning* (pp. 269-297). Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers.

Margolis, J. (2010). *Stuck in the shallow end: Education, race, and computing*. Boston, MA: MIT Press.

Martin, A., McAlear, F., & Scott, A. (2015). *Path not found: Disparities in access to computer science courses in California high schools*. Retrieved from https://eric.ed.gov/?id=ED561181

McFarland, J., Hussar, B., de Brey, C., Snyder, T., Wang, X., Wilkinson-Flicker, S., Gebrekristos, S., Zhang, J., Rathbun, A., Barmer, A., Bullock Mann, F., and Hinz, S. 2017. The condition of education 2017 (NCES 2017- 144). U.S. Department of Education. Washington, DC: National Center for Education Statistics.

McKinsey Global Institute. (2017). *Jobs lost, jobs gained: Workforce transitions in a time of automation.* Retrieved from https://www.mckinsey.com/

Mercier, E. M., Barron, B., & O'connor, K. M. (2006). Images of self and others as computer users: The role of gender and experience. *Journal of computer assisted learning*, *22*(5), 335-348.

Minner, D. D., Levy, A. J., & Century, J. (2010). Inquiry-based science instruction—what is it and does it matter? Results from a research synthesis years 1984 to 2002. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, *47*(4), 474-496.

Moje, E. B., Ciechanowski, K. M., Kramer, K., Ellis, L., Carrillo, R., & Collazo, T. (2004). Working toward third space in content area literacy: An examination of everyday funds of knowledge and discourse. *Reading research quarterly*, *39*(1), 38-70.

National Science Foundation. (2017). Women, minorities, and persons with disabilities in science and engineering. [Data file]. Retrieved from https://www.nsf.gov/statistics/2017/nsf17310/data.cfm

Norton, B. (1997). Language, identity, and the ownership of English. *TESOL quarterly*, *31*(3), 409-429.

Norton, B. (2013). *Identity and language learning: Extending the conversation*. Britsol, UK: Multilingual matters.

Packard, B. L., & Wong, E. D. (1999). *Future images and women's career decisions in science.* Retrieved from https://eric.ed.gov/?id=ED430805

Peppler, K., & Warschauer, M. (2012). Uncovering literacies, disrupting stereotypes: Examining the (dis)abilities of a child learning to computer program and read. *International Journal of Learning and Media*, *3*(3), 15-41.

Ricento, T. (2005). Considerations of identity in L2 learning. *Handbook of research in second language teaching and learning*, *1*, 895-910.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67.

Royal, D., & Swift, A. (2016, October 18). U.S. minority students less exposed to computer science. *Gallup.* Retrieved from http://www.gallup.com/poll/196307/minority-students-less-exposed-computer-science.aspx

Saldaña, J. (2015). *The coding manual for qualitative researchers*. London, UK: Sage.

Schumann, J. H. (1976). Social distance as a factor in second language acquisition. *Language learning*, *26*(1), 135-143.

Smith, M. (2016, January 30). Computer Science For All. [Web log comment]. Retrieved from https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

Vélez-Ibáñez, C. G., & Greenberg, J. B. (1992). Formation and transformation of funds of knowledge among US-Mexican households. *Anthropology & Education Quarterly*, *23*(4), 313-335.

Wang, J., Hong, H., Ravitz, J., & Moghadam, S. H. (2016, February). Landscape of K-12 computer science education in the US: Perceptions, access, and barriers. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 645-650). ACM.

Warschauer, M. (1999). Electronic literacies. Language, culture, and power in online education. Mahwah, NJ: Lawrence Erlbaum.

Warschauer, M. (2009). Digital literacy studies: Progress and prospects. In *The future of literacy studies* (pp. 123-140). London, UK: Palgrave Macmillan.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Wortham, S. (2006). *Learning identity: The joint emergence of social identification and academic learning*. Cambridge: Cambridge University Press.

Zwiep, S. G., & Straits, W. J. (2013). Inquiry science: The gateway to English language proficiency. *Journal of Science Teacher Education*, *24*(8), 1315-1331.