Unpacking Research on Supporting Students with Disabilities in Elementary CS

Maya Israel
University of Florida
misrael@coe.ufl.edu
Main Problem:

There is very little research about the inclusion of students with disabilities in elementary CS education.
Multidimensional Approach to Inclusive Best Practices
(Crouch, Keys, & McMahon, 2014)

- **Organizational Dimension**
  - CS co-teaching, co-planning

- **Academic Dimension**
  - CS curriculum and instruction is accessible
  - Instructional supports in place

- **Assessment & Planning Dimension**
  - CS assessments address strengths & deficits

- **Social Dimension**
  - Students w/disabilities proportionally represented in CS
What we do know

(informed by Crouch et al., 2014)

3 Pillars of Inclusive CS Education

Belief in CS for All including Students with Disabilities

Policy Examples:
- K-12 CS Framework-Equity language
- CSTA Equity Standards for CS Teachers

Accessible Computational Tools & Curricula

Examples:
- Schanzer et al., (2019)
- Stefik & Siebert (2013)

Effective Pedagogical Practices/supports

Examples:
- Israel et al., 2018
- Ray et al., 2018
What we don’t know:

To what extent are students w/disab included in CS education?

How does the intersectionality of disab, gender, SES, culture, etc play into CS learning?

How can we best support teachers in meeting the needs of ALL their learners?

What instructional supports help students who struggle in CS education?
Tool and Curriculum Accessibility Barriers

- There is no such thing as “fully accessible” programming platforms for young learners:
  - Limited access with screen readers
  - Rely on visual representations

- Cognitive supports our typically outside the system (i.e., teachers)

- Many activities are open-ended
Individual Learning Barriers

- Struggle with programming **languages**
  - Challenges with decoding and comprehending the code
- Struggle with multi-step complex **problem solving**
  - Debugging a program that does not work
  - Strategically planning programs from the beginning to end
- Results = **Frustration and task abandonment** (Israel et al., under review)
Addressing Inclusion Barriers

- Use accessible tools (e.g., Bootstrap, Quorum).
- Provide individualized supports, accommodations, and strategies effective in other content areas (e.g., Snodgrass et al., 2016).
- Provide teachers with professional development on strategies and applications of Universal Design for Learning (UDL) in CS education (e.g., Israel et al., 2020).
- Use explicit instruction (Wright et al., 2019) in a balanced approach within open ended computational activities.
Our Approach to Studying Inclusive CS Education

Student-level deep analysis of computational processes

(e.g., Israel et al., under review)

Teacher-level analysis of PD and instructional practices

(e.g., Israel et al., 2020)
Studying Student Engagement and Learning

Data includes:

- Videos of students’ computational behaviors
- Student observations
- New: Eye tracking/gaze fixation
- Artifact-based interviews

Demo Observation

(I) How does the interaction with the peer or adult begin or continue?

Choose a path label
Collaborative Computing Path

Student Driven

- (8) Student clearly expresses how he or she needs help with a difficulty or problem
- (3) Student expresses a need for help, but is not explicit to the difficulty or problem
- (2) Student discusses computing (not problem solving)
- (3) Student engages in non-computing conversation
- (4) Student offers support to peer (the peer did not specifically ask for help)
- (5) Student said something that is unclear or inaudible
- (6) Student verbally addresses a person without expressing the offer or need for help, curiosity, excitement, accomplishment or non-computing conversation (e.g., ‘Hey you...’ or ‘Mrs. S...’ or ‘Stop that!’)
### Universal Design for Learning Guidelines + Computer Science / Computational Thinking

<table>
<thead>
<tr>
<th>Access</th>
<th>Multiple Means of Engagement</th>
<th>Multiple Means of Representation</th>
<th>Multiple Means of Action &amp; Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provide options for</td>
<td>Affective Networks</td>
<td>Recognition Networks</td>
<td>Strategic Networks</td>
</tr>
<tr>
<td>Recruiting Interest</td>
<td>The “WHY” of learning</td>
<td>The “WHAT” of learning</td>
<td>The “HOW” of learning</td>
</tr>
<tr>
<td>• Give students choices</td>
<td>• Model computing using</td>
<td>• Give access to modeled code</td>
<td>• Provide teacher’s codes as</td>
</tr>
<tr>
<td>(choose project,</td>
<td>physical representations</td>
<td>while students work</td>
<td>templates</td>
</tr>
<tr>
<td>software, topic)</td>
<td>as well as through an</td>
<td>independently</td>
<td>• Include CS Unplugged activities</td>
</tr>
<tr>
<td>• Allow students to</td>
<td>interactive whiteboard,</td>
<td>• Provide access to video</td>
<td>that show physical relationship of</td>
</tr>
<tr>
<td>make projects</td>
<td>videos</td>
<td>tutorials of computing tasks</td>
<td>abstract computing concepts</td>
</tr>
<tr>
<td>relevant to culture</td>
<td>• Give access to visual</td>
<td>• Select coding apps and</td>
<td>• Use assistive technology</td>
</tr>
<tr>
<td>and age</td>
<td>settings (such as font size</td>
<td>websites that allow the</td>
<td>including larger/smaller mice,</td>
</tr>
<tr>
<td>• Minimize possible</td>
<td>&amp; contrast) and that are</td>
<td>students to adjust</td>
<td>touch-screen devices</td>
</tr>
<tr>
<td>common “pitfalls” for</td>
<td>compatible with screen</td>
<td>visual settings (such as font</td>
<td>• Select coding apps and websites</td>
</tr>
<tr>
<td>both computing and</td>
<td>readers.</td>
<td>size &amp; contrast) and that are</td>
<td>that allow coding with keyboard</td>
</tr>
<tr>
<td>content</td>
<td>• Provide options for</td>
<td>compatible with screen readers.</td>
<td>shortcuts in addition to dragging</td>
</tr>
<tr>
<td>• Allow for differences</td>
<td>Perceptual</td>
<td></td>
<td>&amp; dropping with a mouse</td>
</tr>
<tr>
<td>in pacing and</td>
<td>• Provide teacher’s codes as</td>
<td></td>
<td></td>
</tr>
<tr>
<td>length of work</td>
<td>templates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sessions</td>
<td>• Include CS Unplugged</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Provide options to</td>
<td>activities that show</td>
<td></td>
<td></td>
</tr>
<tr>
<td>increase or decrease</td>
<td>physical relationship of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sensory stimulation</td>
<td>abstract computing concepts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(for example listening</td>
<td>• Use assistive technology</td>
<td></td>
<td></td>
</tr>
<tr>
<td>to music with</td>
<td>including larger/smaller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>headphones or using</td>
<td>mice, touch-screen devices</td>
<td></td>
<td></td>
</tr>
<tr>
<td>noise cancelling</td>
<td>• Select coding apps and</td>
<td></td>
<td></td>
</tr>
<tr>
<td>headphones)</td>
<td>websites that allow coding</td>
<td>• Provide options for</td>
<td></td>
</tr>
<tr>
<td>• Allow for differences</td>
<td>with keyboard shortcuts in</td>
<td>Physical Action</td>
<td></td>
</tr>
<tr>
<td>in pacing and</td>
<td>addition to dragging &amp;</td>
<td>• Provide teacher’s codes as</td>
<td></td>
</tr>
<tr>
<td>length of work</td>
<td>dropping with a mouse</td>
<td>templates</td>
<td></td>
</tr>
</tbody>
</table>

References


Extra slides
About recent UDL in CS Ed PD this Study

Primary RQ: What does UDL look like in elementary CS education?

Methodology: Interpretive qual. triangulating lesson plans/artifacts, coaching logs, and interviews.

<table>
<thead>
<tr>
<th>Name</th>
<th>Grade</th>
<th>Class Setting</th>
<th>License</th>
<th>Years teaching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lester</td>
<td>2</td>
<td>Inclusive co-taught</td>
<td>Special and General Ed</td>
<td>4</td>
</tr>
<tr>
<td>Rios</td>
<td>K-3</td>
<td>Self-contained special ed</td>
<td>Special ed</td>
<td>4</td>
</tr>
<tr>
<td>Aldridge</td>
<td>3</td>
<td>Inclusive co-taught</td>
<td>General Education</td>
<td>2</td>
</tr>
<tr>
<td>Robinson</td>
<td>4</td>
<td>Gen Ed and Eng. as a New Language (ENL)</td>
<td>General Ed</td>
<td>3</td>
</tr>
</tbody>
</table>
Data Analysis

- Coding: UDL checkpoints
- Categorized data by similarities/differences across teachers
- Interrater reliability among the two coders: Cohen’s Kappa (above .98)

Provide options for Executive Functions (6)

- Guide appropriate goal-setting (6.1)
- Support planning and strategy development (6.2)
- Facilitate managing information and resources (6.3)
- Enhance capacity for monitoring progress (6.4)

Provide options for Recruiting Interest (7)

- Optimize individual choice and autonomy (7.1)
- Optimize relevance, value, and authenticity (7.2)
- Minimize threats and distractions (7.3)
CT & CS Activities

- Combination of plugged & unplugged activities
- Begin with unplugged, transition to simple plugged, and finally to more sophisticated plugged activities
Results... MORE CHECKPOINTS  BETTER

<table>
<thead>
<tr>
<th>Engagement (n=63 instances)</th>
<th>Representation (n=51 instances)</th>
<th>Action and Expression (n=31 instances)</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Choice in computational artifacts (with accountability)</td>
<td>*Options for perception (e.g., customize display of info)</td>
<td>*Focus on executive function</td>
</tr>
<tr>
<td>*Personally relevant/real-world applications</td>
<td>*Explicit directions, visual cues, modeling</td>
<td>-Goal setting</td>
</tr>
<tr>
<td>*Student collaboration</td>
<td>*Less for on supports for language and symbols</td>
<td>-Planning support</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Progress monitoring</td>
</tr>
<tr>
<td></td>
<td></td>
<td>*Less focus on options for physical action or options for expression and communication</td>
</tr>
</tbody>
</table>
Significant differences in UDL implementation across the teachers.

UDL is contextual.
**UDL checkpoints DO NOT hold equal weight in individual lessons: They are context and student dependent!**