Supporting Students with Learning Disabilities in K-8 Computer Science Instruction

Project TACTIC: Teaching All Computational Thinking through Inclusion and Collaboration

Maya Israel, Ph.D.

Department of Special Education; University of Illinois at Urbana-Champaign

For more information: Email: misrael@illinois.edu; Lab Website: http://ctrl.education.illinois.edu ; Twitter: @misrael09

Creative Technology Research Lab

RATIONALE

Although the Computer Science (CS) for All movement has focused on broadening participation of women and people from different cultural and socioeconomic backgrounds, disability has only recently been discussed within this context (Ladner & Israel, 2016).

Through a National Science Foundation STEM+C Project, we are studying:

1) Methodologies for studying engagement and learning within CS education;

2) Challenges faced by students with learning disabilities and other struggling learners within CS/CT instruction in grades 3-8; and

3) Instructional strategies that address those challenges.

Research Questions

1) What benefits and challenges do students with learning disabilities face during K-8 computer science activities?

2) What strategies can teachers implement to support students with learning disabilities in K-8 computer science activities?

3) How can we reconcile teaching approaches between CS instruction and intensive intervention supports?

CHALLENGES OF STUDENTS WITH LEARNING DISABILITIES IN CS

-Generalizing from teacher model to own work

-Decoding and comprehending the code/blocks

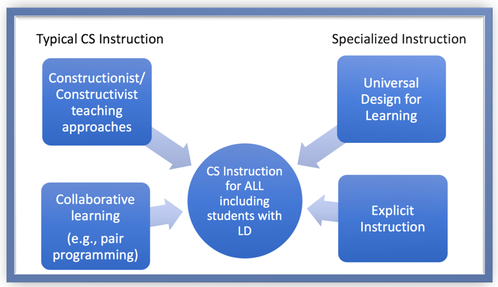
-Understanding "big ideas" such as conditional logic

-Multi-step complex problem solving & debugging

-Lack of persistence

-Lack of opportunity (being excluded from CS opportunities)

*Lack of agreement about instructional approaches between CS methods and individualized student needs



EXPLORATORY METHODOLOGY

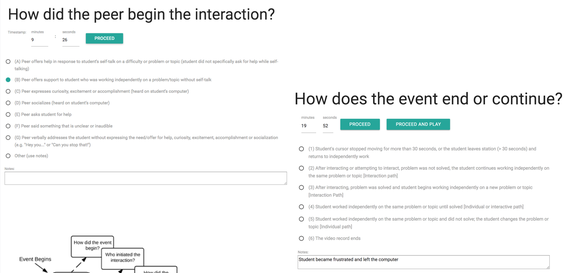
Participants: Teachers and students in grades 3-8 with disabilities or receiving Tier 2 RTI supports

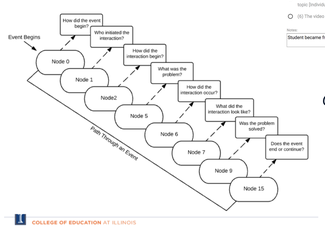
CS/CT Curricula:

Scratch: Block-based, visually-intuitive programming language. Students drag and connect blocks of commands to create computational artifacts such as stories, games, and animations.

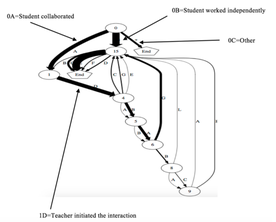
Code.org Code Studio: Block-based, visually-intuitive programming language. Students complete increasingly complex puzzles until they master skills that allow them to freely create programs within the Code Studio Play Lab

Collaborative Computing Observation Instrument: (C-COI)

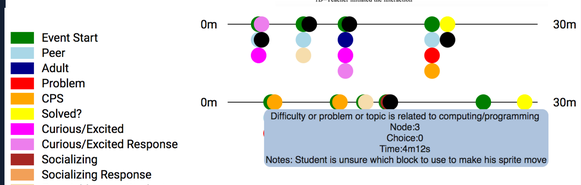




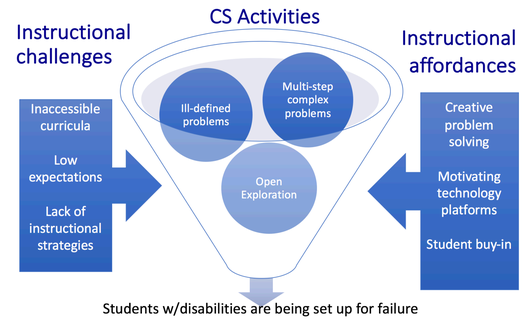
C-COI Nodes



Example C-COI Directed Graph



This visualization is used to understand differences between student interaction, either longitudinally or between different students completing the same task.



Data Sources

1) Classroom observations using a structured observation protocol

2) Teacher Interviews

3) Teacher Lesson Plans

4) Collaborative Computing Observation Instrument (C-COI; Israel et al., 2016): Used alongside video screen capture of students as they engage in CS/CT. It has 16 broad categories for: Challenges; Time on task; Help seeking; Independent and collaborative problem solving; Socializing interactions

Promoting Strategic Problem Solving

Balancing Explicit Instruction and Open Inquiry and Universal Design for Learning

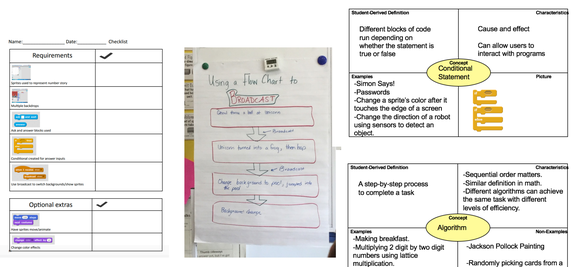
(1) Problem solving protocols and checklists: Help with self regulation and problem solving during complex CS activities.

(2) Breaking tasks into smaller subtasks and modeling those subtasks

(3) Providing immediate/formative feedback to students as they work through subtasks

(4) Teaching CS vocabulary explicitly

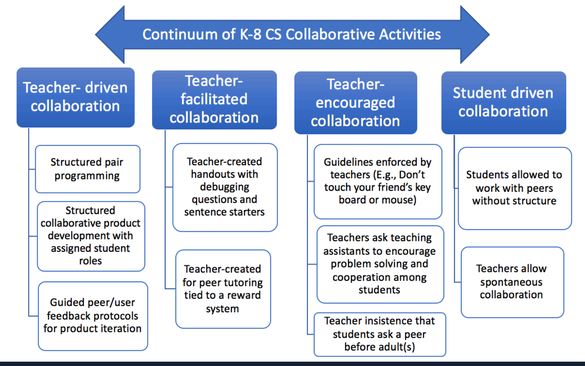
(5) Offering constrained choice in project development



PROMOTING COLLABORATION

Collaborative problem solving in computer-supported collaborative learning (CSCL) is complex and relies on sophisticated social and cognitive skills (Hesse et al., 2015)

Scripted Conversations Scripted conversations can be used to support student learning and social interactions (Koller et al., 2016). These have not been studied with students with disabilities. Collaborative Discussion Framework (CDF): students and peers in the study were taught the CDF (Park & Lash, 2014) to support adaptive help seeking and collaborative problem solving



INCREASING TEACHER CAPACITY

1) Instructional coaching:
   Co-planning, co-teaching, resource sharing, reflection
2) Professional development:
   In the CS software and pedagogical practices
3) Resource development--TACTICal Teaching Briefs:
   -Role of paraeducators
   -Universal Design for Learning
   -Vocabulary instruction
   -Promoting effective collaboration

ACKNOWLEDGEMENTS

Research reported in this presentation was supported by the National Science Foundation STEM+C program [Award # 1639837].

NSF

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN