CT IN ELEMENTARY SCIENCE TEACHER EDUCATION

Developing a Framework for Integration

Dr. Diane Jass Ketelhut & Lautaro Cabrera
University of Maryland, College Park
Our Project

- CT Integrated into Elementary Science Methods Course
  - *Focused on four sessions*
  - *Final assignment: CT-infused science lesson*
  - *Modest results: preservice teachers used CT terms loosely*

- Professional Development experience
  - *Science Teaching Inquiry Group in Computational Thinking (STIG<sup>CT</sup>)*
  - *Pre-service and in-service teachers learn and work together, including mentor-mentees pairs*
  - *Researchers and teachers co-design CT-infused science lesson plans*
  - *Introduce teachers to CT concepts through elementary school science activities*
Framework Development

- For both the course and the STIG\textsuperscript{CT}, we iteratively developed a framework for integrating CT into elementary science.
- The framework guided participant learning, discussion around CT, and integration of CT into lesson plans.
- Different versions of the framework were accompanied by different results in how teachers integrated CT.
FRAMEWORK YEAR 1
Our Framework Iterations: Year 1

■ Drew from multiple sources:
  – *Weintrop et al.* (2016): CT practices specifically for science and math
  – *CSTA & ISTE* (2011): inclusion of dispositions and attitudes
  – *Barr & Stephenson* (2011): use of concrete examples

■ Created our own examples of each CT Practice (from Weintrop et al.)
Data Practices
- Collecting Data
- Creating Data
- Manipulating Data
- Analyzing Data
- Visualizing Data

Modeling & Simulation Practices
- Using Computational Models to Understand a Concept
- Using Computational Models to Find and Test Solutions
- Assessing Computational Models
- Designing Computational Models
- Constructing Computational Models

Computational Problem-Solving Practices
- Preparing Problems for Computational Solutions
- Programming
- Choosing Effective Computational Tools
- Assessing Different Approaches/Solutions to a Problem
- Developing Modular Computational Solutions
- Creating Computational Abstractions
- Troubleshooting and Debugging

Systems Thinking Practices
- Investigating a Complex System as a Whole
- Understanding the Relationships within a System
- Thinking in Levels

Operational Definition
CT is a problem-solving process that includes (but is not limited to) the following characteristics:

- Formulating problems in a way that enables us to use a computer and other tools to help solve them
- Logically organizing and analyzing data
- Representing data through abstractions such as models and simulations
- Automating solutions through algorithmic thinking (a series of ordered steps)
- Identifying, organizing, and implementing possible solutions with the goal of realizing the most efficient and effective combination of steps and resources

CT Vocabulary and Progression Chart

- Data Collection
- Data Analysis
- Data Representation

Weintrop et al. (2016)
CSTA & ISTE (2011)
Year 1 “Framework” Challenges

■ The framework language was sometimes inaccessible or overwhelming for teachers—it was based on CS terminology
  – *E.g.*, *algorithmic thinking* or *computational abstraction*

■ Hard to differentiate CT practices from other more common scientific practices
  – *E.g.*, *CT data collection* vs. *science data collection*
Using Data Programming

Computational Simulations Systems Thinking from a CT Perspective

- Using data with computational devices
- Finding patterns and relationships in data sets
- Sorting data
- Creating graphs or charts
- Breaking down problems into smaller parts
- Coding
- Creating step by step instructions to solve a problem
- Identifying quantifiable parts of a system
- Considering numerical relationships within a system
- Considering how changes to the quantifiable parts contribute to results of the system

- Test, Adjust to Improve, Retest, Readjust to Improve...

- Using computational simulations
- Assessing computational simulations

- Student's Favorite Colors

- Analyzing data from computational simulations

- Analyzing data from computational simulations

- Analyzing data from computational simulations

- Analyzing data from computational simulations
Unified sources into one framework

Using Data
- Finding patterns and relationships in data sets
- Sorting data
- Creating graphs or charts
- Collecting data with computational devices

Programming
- Breaking down problems into smaller parts
- Coding
- Creating step by step instructions to solve a problem
- Test, Adjust to improve, Retest, Readjust to improve...

Computational Simulations
- Assessing computational simulations
- Using computational simulations

Systems Thinking from a CT Perspective
- Identifying quantifiable parts of a system
- Considering numerical relationships within a system
- Considering how changes to the quantifiable parts contribute to results of the system
Reduced number of practices

**Using Data**
- Finding patterns and relationships in data sets
- Creating graphs or charts
- Sorting data
- Collecting data with computational devices

**Programming**
- Breaking down problems into smaller parts
- Coding
- Creating step by step instructions to solve a problem
- Test, Adjust to improve, Retest, Readjust to improve...

**Computational Simulations**
- Assessing computational simulations
- Using computational simulations

**Systems Thinking from a CT Perspective**
- Identifying quantifiable parts of a system
- Considering numerical relationships within a system
- Considering how changes to the quantifiable parts contribute to results of the system
Simplified language to avoid CS jargon

Using Data
- Finding patterns and relationships in data sets
- Creating graphs or charts
- Sorting data
- Collecting data with computational devices

Programming
- Breaking down problems into smaller parts
- Creating step by step instructions to solve a problem
- Test, Adjust to Improve, Retest, Readjust

Computational Simulations
- Assessing computational simulations
- Using computational simulations

Systems Thinking from a CT Perspective
- Identifying quantifiable parts of a system
- Considering numerical relationships within a system
- Considering how changes to the quantifiable parts contribute to results of the system

Formerly “Algorithmic Thinking”
Differentiated CT from science practices

Added a quantifiable or numerical component

Computational Simulations

Using Data

Programming

CT Perspective

Identifying quantifiable parts of a system

Considering changes to the quantifiable parts contribute to results of the system

Considering numerical relationships within a system
Preliminary Results

- With the new framework, teachers are feeling more comfortable integrating CT
  - *Both in written reflections and self-efficacy measures*
- They are more successfully integrating CT into their lesson plans than in Year 1
  - *The instances of CT in their lesson plans more closely resembled the CT practices of the framework*
- Mentors and mentees are benefitting from working together
  - *Different but complementary expertise*
Remaining Challenges

- Almost no teachers integrated Systems Thinking from a CT Perspective. Are these practices appropriate for the elementary level?

- Simplifying language to avoid CS jargon may have led to some superficial uptake
  - Sometimes “step-by-step instructions” meant following any type of procedure was considered CT
Moving Forward

- How are teachers implementing the lessons they design?
  - What are the instances of CT that are developmentally appropriate, work within school structures, and teachers feel comfortable integrating?

- Which CT practices are making it into the Elementary classroom?
  - How is the framework guiding the design and implementation of lessons?