Mixed Methods for the Assessment and Incorporation of Computational Thinking in K-12 and Higher Education

KANSAS STATE UNIVERSITY
Computer Science

Joshua Levi Weese

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, and funded by Laboratory Directed Research and Development project 16-ERD-019.

National Science Foundation (DUE-1347821)
Title: WIDER: Data Explorer and Assessment Resources for Faculty
9/15/2013 – 9/14/2017

Abstract

A movement to include computer science in K-12 curriculum standards has sparked a significant interest in computational thinking (CT). This paper describes current and future work in the development of visual programming curricula for teaching CT at the K-12 level and self-efficacy surveys for evaluating the effectiveness of the curricula at fostering CT. Current work on a comprehensive system for automated extraction of assessment data for descriptive analytics and visualization is also described. To complement attitude surveys, a translation of Scratch to Blockly is proposed. Data on student programming behaviors at the collegiate level will be collected and quantitatively analyzed to help assess CT in support of self-efficacy.

Problem Statement

This dissertation aims to provide a clear and practical definition of computational thinking, as well as effective pedagogy and mixed methods for assessing CT. The research questions aligned with this statement are as follows:

1. Do the computer science principles chosen to be included in computational thinking align with student learning outcomes?
2. Can the attitude surveys created reliably assess student ability in computational thinking?
3. Are the computer science principles taught through outreach curricula reflected in student confidence in CT?
4. Is there a link between qualitative and quantitative analysis of computational thinking? If so, does quantitative analysis support or refute self-efficacy as a method of measuring CT?
5. Can student personas be extracted from traditional physics assessments, and if so, can the method be generalized to apply to CT assessments?

Computational Thinking

"The process of recognizing aspects of computation in the world that surrounds us, and applying tools and techniques from computer science to understand and reason about both natural and artificial systems and process.[1]"

Concepts
- Algorithmic Thinking
- Abstraction
- Problem Decomposition
- Data
- Parallelization
- Control Flow

Practices[2]
- Being Incremental and Iterative
- Testing and Debugging
- Reusing and Remixing

Perceptions[2]
- Expressing
- Connecting
- Questioning

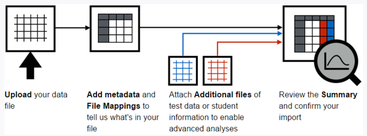

Figure 1 The process of uploading assessment data in the Data Explorer

Data Explorer[3]

The Data Explorer is an analytics portal that gives educators the ability to gain insights from assessment data. The focus of this research is to facilitate the importing, validation, integration, and visualization of assessment data. Using online information retrieval and visualization through an interactive display, educators interact with their data, as well as data from others in a usable, efficient, and secure way. Users will also be able to compare data against other educators to understand where their students lie.

Blockly Scratch

Program logging and analysis, including compilation and programming process, is present in current text-based programming literature, like Blackbox[4] however, this has not been expanded to block-based languages. This project aims to expand Google's Blockly with Scratch features to gather fine grained data of students' learning process in outreach programs and introductory computer science courses at the collegiate level. This provides a vehicle for machine learning and critiquing applications for improving pedagogy and the student learning experience.

Blockly Scratch

Data Explorer beta by PhysPort

Outreach

Learning how to program, especially for those who have no experience in computer science fundamentals, requires a significant amount of overhead. This work focuses using Cognitive Load Theory and Problem-Based Learning to allow students to focus on learning how to think computationally while also becoming computer programmers.

Mission to Mars[5]

Students in 5th and 6th grade participated in a summer STEM Institute in 2015 and 2016. This curriculum infused CT concepts in context of sending an autonomous rover to Mars, as well as in theme of the recent movie, *The Martian*. Using Scratch, students learned a variety of foundational CS theory, including:

- Artificial Intelligence
- Search algorithms
- Scientific simulation
- Hexadecimal and binary

Mighty Micro Controllers

During the summer of 2016, 7th-9th graders in the STEM Institute were able to get hands-on experience in circuitry, sensors, and microcontrollers. Students were able to use Scratch to program an Arduino Uno to create anything from LED arrays to motion sensors. Computational thinking infused in this curriculum included:

- Data collection, representation, and analysis
- Abstraction
- Control flow
- Engineering design process

KANSAS STARBASE

Figure 2 Kansas STARBASE exposes 5th and 6th graders to STEM disciplines, including engineering, robotics and computer science.

Video Game Design[5]

Students in 7th-9th grade participated in a summer STEM Institute in 2015. By using video game design principles, students learned advanced programming techniques through Scratch, improving their confidence in CT and highlighting AI, pair, and parallel programming.

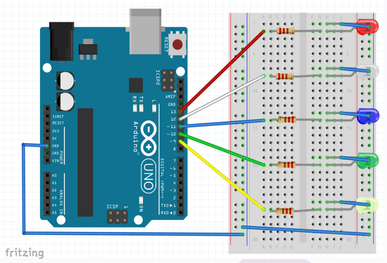

Figure 3 A wiring diagram, made in Fritzing, showing one of the simple LED arrays students completed in Mighty Micro Controllers.

Results

More than 80% of students have used some programming language; this shows that all students are being exposed as much to computer programming at home or school as those who participated in outreach programs. However, the *level* of exposure is reflected in the self-efficacy in CT concepts.

Computational Thinking	No Previous Attendance in STEM Programs	Previous Attendance in STEM Programs	Mission to Mars	Game Design
Executes a sequence of commands	.352	.632	.496	.546
Uses loops to repeat commands	.641	.785	.683	.779
Responds to events	.259	.539	.584	.231
Parallelism	.482	.656	.556	.621
Conditional commands	.498	.508	.582	.408
Perform math operations	.265	.387	.481	.162
Perform Boolean operations	.606	.626	.626	.608
Store, update, and retrieve values	.405	.550	.429	.568
Ask user for input	.292	.694	.537	.522
Iterative development	.331	.417	.322	.456
Frequent tests/debugging	.519	.481	.533	.452
Share and collaborate with programs	.337	.573	.445	.517
Break program into parts	.537	.412	.448	.482

Table 1 Effect size from pre to post-survey from the 2015 summer STEM Institute.

Future Work

The 2016 summer STEM Institute, as well as Starbase, is currently being analyzed to improve the CT survey. The Data Explorer will be modified to accept custom assessments (beyond just Physics), and will also include improved data extraction and intelligent clustering for visualization. Blockly Scratch is still in development, but will be on schedule to go live for Spring 2017.

References

1. Royal Society. (2012). Shut Down or Restart? The way Forward for Computing in UK Schools. The Royal Academy of Engineering.
2. 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.
3. Weese, J. L., & Hsu, W. H. (2016, June), WORK IN PROGRESS: Data Explorer – Assessment Data Integration, Analytics, and Visualization for STEM Education Research Paper presented at 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana. 10.18260/p.27215
4. Altadmri, A., & Brown, N. C. (2015). 37 Million Compilations: Investigating Novice Programming Mistakes in Large-Scale Student Data. Proceedings of the 46th ACM Technical Symposium on Computer Science Education - SIGCSE '15, (pp. 522-527).
5. Weese, J. L., & Feldhausen, R., & Bean, N. H. (2016, June), The Impact of STEM Experiences on Student Self-Efficacy in Computational Thinking Paper presented at 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana. 10.18260/p.26179