Computing Computational Thinking: Towards the Automatic Recognition of Computational Thinking in Real Time

Submitted by Kyu Han Koh, Doctoral student, in Department of Computer Science
Faculty Advisor: Dr. Alexander Repenning, Professor in Department of Computer Science

Since the 1990’s there have been multiple efforts to fix the broken pipeline at the K-12 level in computer science (CS) education [1-3], and most of those efforts have focused on the student motivational factor [4, 5]. The results of many studies in computer science education indicate that student motivation in computer science has been successfully increased by those efforts [4, 5], but most of them failed to rigorously assess student learning. I believe that this biased tendency of CS education research was caused by failing to have the right instrument to measure student learning of content at the semantic level. In other words, the right assessment instrument should be able to assess not only assess acquired students’ knowledge but also guide students to the next level of learning. Student learning may be measured with existing tools such as grading rubrics, but it is extremely time consuming and has a limited functionality to provide necessary educational feedback from students’ learning progressions.

In order to solve these challenges, I propose a method of automating the grading process. The method I propose, Computational Thinking Pattern Analysis (CTPA) (Figure 1) [6], is designed to measure student-learning skills and represent student content learning at the semantic level through phenomenological analysis in real-time. This concept uses a Latent Semantic Analysis (LSA) [7] inspired technique, multiple high dimensional cosine calculations to analyze semantic meanings of a given context with several pre-defined subjects/phenomena. Theoretically, this idea can be applied to several different domains such as natural language processing and visual end user programming. Therefore, this idea can be employed to build a learning assessment tool for computer science (CS) and/or computational thinking (CT) [14] education where visual programming is widely adopted.

**Figure 1. CTPA Graph Example. This graph is an actual student’s example of a game of Frogger.**
As part of my dissertation, I designed, developed, and studied the Scalable Game Design Arcade (Figure 2) [15], a cyberlearning infrastructure where middle school to grad school students share (play, download, leave comments, and rate each other on) their AgentSheets [3] projects including games and STEM simulations. The Scalable Game Design Arcade (SGDA) has collected more than 10,000 projects from 7,000 students over the past three years. All collected AgentSheets projects through SGDA are analyzed by CTPA.

![Scalable Game Design Arcade](image)

Figure 2. The Individual Page of SGDA. Individual page illustrates the screenshot of the game (upper left), Run and Download button (upper right), the game’s similarity score compared to four tutorial games (middle right), a similarity score matrix showing similarly programmed games to the submitted game, and the CTPA graph (bottom right & left).

The analyzed results of CTPA have shown promise in providing educational assessments of learning trajectory [8], learning transfer [6], and programming divergence [9].

The main focus of this research aims at evaluating the CTPA to prove its usefulness as an educational instrument for STEM education. To evaluate CTPA’s reliability (to verify its consistency of measuring outcomes) and CTPA’s validity (to verify that CTPA actually measures what we attempt to assess) as a program analysis tool for educational purposes, I plan to employ the following research plan.

**Research Plan**

Since the early stage of CTPA development, the CTPA has been validated formally and informally with traditional Computer-Automated Scoring (CAS) system validation approaches and other educational validation approaches.

Yang et al. classified CAS system validation approaches into three different categories: [10].
• Approaches focusing on relationships among scores generated by different scorers
• Approaches focusing on relationship between test score and external measures
• Approaches focusing on scoring processes

And I plan to add one additional category to this CAS system validation approach:

• Approaches focusing on predictive validity

For a good validation approach for CTPA, the purpose of CTPA and the consequence of the interpretation of CTPA should be addressed. The original CTPA was not designed as a computer-automated scoring system, but it was close to a program analysis tool that could present students’ problem solving skills and programming skills. It demonstrates what kinds of computational thinking (pattern) skills were used to implement AgentSheets [3] projects to indicate students’ problem solving skills and how well the projects were implemented (similarly programmed to the given tutorials) to show students’ programming skills. Nevertheless, the interpretation of CTPA has showed its possibility as a CAS system with the development of CTPA applications such as Learning Skill Scores calculation [8]. Also, it is possible to convert CTPA into a holistic score with CTPA applications. It means that the consequence of the interpretation of CTPA is able to translate CTPA analysis into a computer-generated score. Therefore, the consequence of the interpretation of CTPA is similar to the ones of other CAS systems. It is able to provide instructional and diagnostic feedback including self-assessment.

CTPA can be considered as not only a CAS system but also a program phenomenological analysis tool. So the CTPA should be validated from several different aspects with several different validation approaches.

**Category 1: relationships among scores generated by different scorers**

My approach will assess the correspondence between CAS-system-generated scores and scores assigned by human scorers.

With this kind of validation approach, there is a high possibility of variations due to intra-rater inconsistency and inter-rater differences. To solve this issue, more than 3 scorers/experts will participate in this category evaluation, and their rating will be classified into three different categories; pure programming ratings, pure design ratings, and overall ratings. Also, the correlation between CTPA-generated scores and expert-generated scores will be obtained by Pearson correlation and Spearman rank correlation, and their correlations values’ significance will be assessed too.

**Category 2: relationship between test score and external measures**

In this category, my study is designed to evaluate the relationship between the CAS-system-generated scores and Computational Thinking Pattern Quiz [11]. The Computational Thinking Pattern Quiz is another computational thinking assessment tool to measure students’ understanding of computational thinking (pattern) skills through phenomenological comparisons.

Computational Thinking Pattern (CTP) Quiz has a certain set of questions that involves video of real-life phenomena relating to patterns students programmed in their class implementation period. All CTP Quiz questions are open-ended questions, and students are able to answer freely why and how they match computational thinking patterns in the quiz videos and computational thinking patterns from the AgentSheets project they programmed. Basically this quiz works as a student interview that can judge students’ understanding of computational thinking patterns. This CTP Quiz results will be evaluated by several computational thinking experts’ agreements to check its internal consistency.
Additionally, I will assess the differential agreement between CTPA scores and ratings given by people with different levels of domain expertise including students, teachers, and researchers as Landauer et al. did for their CAS system [12]. They reported that their CAS-generated scores showed higher correspondence with ratings given by scorers with higher expertise than with those given by scorers with lower expertise [10, 12].

**Category 3: Approaches focusing on scoring processes**

Converting CTPA into a holistic score is not the only way to validate CTPA. CTPA can be validated with approaches focusing on scoring processes. For example, when a game of Frogger is graded, a grader will look the functionalities of Frogger; Frogger can be moved with key controls (cursor control), a tunnel creates cars or trucks (generation), etc. Thus, CTPA also should be able to check the functionalities of computational thinking patterns as a human scorer does. To do so, each canonical pattern of CTPA should be matched to the basic requirement of a human grader’s or a student’s perception. This process that refines canonical computational thinking patterns can be done with statistical approaches or human expert agreement. Currently, the canonical computational thinking pattern is defined by human expert agreement, but I plan to add a statistical approach, which captures an average model for each computational thinking pattern from a project submission database, as well.

**Category 4: Approaches focusing on predictive validity**

The last validity subject is predictive validity, which could be a possible clue to learning progressions interpretation. I will determine predictive validity by calculating R-squared value comparing student (CTPA-computed) skills before the final project and student (CTPA-computed) skills of the final project. If they are highly related, then we can say that students who gain more experienced through pre-final project assignments are expected to show better skills in the final project too.

**Intellectual Merit:** Semantic assessment in CS/CT education would be able to provide better individual feedback and faster learning assessment to students and teachers by measuring student skills and challenges and analyzing learning contents at the semantic level. This kind of feedback can be used to determine when and how teachers can expand students’ learning capability in accordance with the theories of the Zone of Proximal Development and Flow [13]. A validated CTPA will contribute to the study of learning theory, professional development, and educational data mining by providing empirical data in order to refine the current conceptual framework of educational systems.

**Broader Impact:** This research suggests a method that can assess students’ learning skills, provide effective learning guidelines, and compute students’ learning outcomes. This type of method, which cannot be found widely, can be used to create cyberlearning systems that help large numbers of teachers and students to learn computational thinking. This research is currently being applied to build a cyberlearning infrastructure, which is used for the Educational Game Design class, which has been held in computer science department since the spring of 2010 and will be held in ATLAS from the fall of 2013.

**A timeline for completion of the Project**

May 2013 – Data aggregation. In the first month, I plan to put together the data sets necessary to answer some research questions described in my dissertation description document. Much of the data has already been collected, but it needs to be collated in an appropriate format for answering each question. This phase is also exploratory, which will help tweak the framework in my dissertation synopsis to better represent the patterns found in the data.
May - June 2013 – Survey and experiment deployment for validity evaluation. Also in the first month, I plan to deploy the necessary design interventions that will help get the necessary data that I do not have already. Most of these experiments are already designed and just need to be tweaked.

Category 1 validation – From 3rd week of May to 4th week of May.

Category 2 validation – From 3rd week of May to 2nd week of June.

June 2013 – Category 3 validation/Software quality assurance – Based on the result of validity evaluation, I plan to update the current algorithm and technique of CTPA to reduce false positive/negative of CTPA and increase the validity of CTPA.

July - August 2013 – Data analysis. I plan to create the necessary visualizations and statistical analyses to examine some research questions outlined in my dissertation synopsis.

Learning Trajectory – From 1st week of July to 4th week of July.

Social Programming Divergence – From 1st week of August to 4th week of August.

Category 4 validation – 4th week of August.

August - September 2013 – Results. I plan to put together the findings in a structured form, and integrate all the findings into a cohesive document accessible to a broad audience.

November 2013 – Draft – a first draft of the project report.

March 2014 – Final version.
References


