# Developing the Systems Thinking and Computational Thinking Identification Tool

Jonathan Bowers, Michigan State University, bowersj8@msu.edu
Namsoo Shin, Michigan State University, namsoo@msu.edu
Linsey Brennan, Michigan State University, mcmull74@msu.edu
Emil Eidin, Michigan State University, Eidinema@msu.edu
Lynn Stephens, The Concord Consortium, lstephens@concord.org
Steve Roderick, The Concord Consortium, sroderick@concord.org

**Abstract:** We developed the Systems Thinking (ST) and Computational Thinking (CT) Identification Tool (ID Tool) to identify student involvement in ST and CT as they construct and revise computational models. Our ID Tool builds off the ST and CT Through Modeling Framework, emphasizing the synergistic relationship between ST and CT and demonstrating how both can be supported through computational modeling. This paper describes the process of designing and validating the ID Tool with special emphasis on the observable indicators of testing and debugging computational models. We collected 75 hours of students' interactions with a computational modeling tool and analyzed them using the ID Tool to characterize students' use of ST and CT when involved in modeling. The results suggest that the ID Tool has the potential to allow researchers and practitioners to identify student involvement in various aspects of ST and CT as they construct and revise computational models.

## Introduction

Many of our current societal and ecological challenges involve complex systems composed of interconnected elements. From global pandemics to climate change, these challenges require systems thinking (ST) to identify how various elements contribute to emergent effects in large-scale systems. ST enables individuals to investigate how a single part of a system can have broader impacts on the whole system (Meadows, 2008). Given the complexity of most systems, computational thinking (CT) is often required to approach these problems. CT is a sensemaking process where one decomposes a problem in a systematic way, translates it into an algorithm that can be interpreted by an information processing agent, and iteratively refines it based on new observations and new data inputs (Grover & Pea, 2018; Wing, 2006). Because both ST and CT are important for addressing problems involving complex systems, it is fruitful to consider their synergies for investigating phenomena (Shin et al., in press; Weintrop, 2016). ST and CT are also increasingly being emphasized as important elements of science education on a global scale, being incorporated into official policy documents in many countries including the U.S., the U.K., and Taiwan (Csizmadia et al., 2015; NGSS Lead States, 2013; So et al., 2020).

These efforts to include ST and CT as key aspects of science education create a need for developing new research tools for characterizing and monitoring student use of these types of thinking (Grover & Pea, 2018). One framework that recognizes the interconnected relationship between ST and CT is the "ST and CT Through Modeling Framework," which describes how student use of ST and CT can be supported through the construction of computational models (Bowers et al., 2022; Shin et al., in press). This framework seeks to clarify and expand the conceptualizations of ST and CT as proposed by the NGSS as well as demonstrate the synergy between ST, CT, and modeling (NGSS Lead States, 2013; Shin et al., in press). Given its focus on the interconnectedness of ST and CT, this framework provides a foundation for developing an instrument for observing student use of ST and CT as they construct and revise computational models. Such a tool may facilitate researchers in recognizing instances of and patterns in students' use of specific ST and CT aspects as they construct and revise models. In this paper, we first summarize our conceptualization of the three main components of our framework (ST, CT, and modeling) and how these components combine to form five computational modeling practices. We then describe how we developed a research tool based on this framework to explore student use of ST and CT as they constructed and revised computational models using a semi-quantitative computational modeling tool. Finally, we provide examples of how this tool can be used to identify and categorize student use of ST and CT.

## Theoretical Approach

Systems thinking is an approach to exploring a phenomenon as a network of elements that work together to create a system with emergent behavior that is more than the sum of its constituent parts (Arnold & Wade, 2015; Forrester, 1971; Meadows, 2008). We define an "element" as a key part of a system that can be independently described yet interacts with other aspects of the system to impact the overall behavior of that system. Many complex phenomena can be described as a series of interacting elements with feedback relationships and informational delays that often generate counterintuitive behaviors (Booth-Sweeney & Sterman, 2007; Cronin et al., 2009). To fully engage in ST, students need to move beyond simple linear causal reasoning to a system behavior perspective so that they can identify common structural patterns found within and across systems. Our framework identifies five major aspects of ST: (1) defining a system's structure and boundaries, (2) engaging in causal reasoning, (3) recognizing interconnections and identifying feedback, (4) framing problems or phenomena in terms of behavior over time, and (5) predicting system behavior based on system structure (Shin et al., in press).

Computational thinking has many definitions ranging from being grounded in mathematics and data analysis (NRC, 2012) to being an aspect of sensemaking centered on formulating questions through testing models and simulations (Schwarz et al., 2017; Weintrop et al., 2016) to thinking like a computer scientist (Grover & Pea, 2018; Wing, 2006). Synthesizing these approaches, we define CT as a form of sensemaking that uses an iterative and quantitative approach to decompose a phenomenon or problem to explore, explain, and predict the behavior of that phenomenon or to find a solution to a problem through the creation and iterative revision of algorithms (Shin et al., in press). Our framework identifies five major aspects of CT: (1) decomposing problems such that they are computationally solvable, (2) creating computational artifacts using algorithmic thinking, (3) generating, organizing, and interpreting data, (4) testing and debugging, and (5) making iterative refinements.

In addition to ST and CT, modeling forms the third component of our framework. Modeling is the process of creating a representation of a phenomenon such that the representation can be used to explain or predict the behavior of that phenomenon (Harrison & Treagust, 2000; Schwarz et al., 2009). From this perspective, models are viewed not just as the product of scientific inquiry but as essential tools for supporting scientific reasoning and sensemaking (Schwarz et al., 2009). Additionally, analyzing existing models can help one gain insight into different aspects of a phenomenon and predict its future behavior (zu Belzen & Krüger, 2010). Scientists and students often use models to represent their conceptualization of a phenomenon so that they can synthesize and communicate their ideas to others (Gilbert & Justi, 2016). Within our framework, students utilize both ST and CT approaches as they engage in the process of modeling.

While researchers (Berland & Wilensky, 2015; Wing, 2017) claim that CT and ST are intertwined and support each other, we view CT and ST as co-equal, yet distinct in the context of modeling because of their unique ways of approaching problems. CT focuses on designing solutions through computation while ST analyzes the various relationships among elements in a system (Shute et al., 2017). Our framework thus defines CT and ST as separate entities and identifies five computational modeling practices that combine aspects of ST and CT: (M1) characterize problem or phenomenon to model, (M2) define the boundaries of the system, (M3) design and construct model structure, (M4) test, evaluate, and debug model behavior, and (M5) use model to explain and predict behavior of phenomenon or design solution to a problem (Bowers et al., 2022). Students engage in these modeling practices as they construct, test, revise, and use their computational models. Students characterize the phenomenon (M1) as they discuss and unpack key elements of the phenomenon under study and as they learn about new elements of the phenomenon. Students define the boundaries of the system (M2) and design/construct model structure (M3) as they discuss which variables to add to their models and set relationships between these variables respectively. Once students have built their initial models, they can analyze the model output and should compare this output to real-world data or their emerging understanding of the phenomenon to identify and modify flaws in their model, thus testing and debugging of model behavior (M4). Finally, students use their models to construct explanations of the phenomenon or predict how the system will behave under different circumstances (M5). Each of these practices are supported by a combination of aspects of ST and CT (Table 1).

**Table 1**
*The computational modeling practices and associated ST and CT aspects*

| Computational Modeling Practice | Associated ST and CT Aspects |
|---|---|
| M1. Characterize Problem or Phenomenon | ST: Define a System<br>CT: Decompose Problems |
| M2. Define System Boundaries | ST: Define a System, Frame Phenomena in Terms of Behavior over Time<br>CT: Decompose Problems, Create Algorithmic Artifacts |
| M3. Design and Construct Model Structure | ST: Engage in Causal Reasoning, Recognize Interconnections and Feedback, Frame Phenomena in Terms of Behavior over Time |

| | |
|---|---|
| | CT: Create Algorithmic Artifacts |
| M4. Test, Evaluate, and Debug Model Behavior | ST: Define a System, Predicting System Behavior Based on System Structure<br>CT: Generate and Interpret Data, Test and Debug, Make Iterative Refinements |
| M5. Use Model to Explain and Predict Behavior of Phenomenon | ST: Predict System Behavior Based on System Structure, Engage in Causal Reasoning<br>CT: Generate and Interpret Data, Test and Debug, Make Iterative Refinements |

Although the science education community has established ST, CT, and modeling as key learning goals, we know relatively little about how to support students in these practices. We used the ST and CT Through Modeling Framework to develop a research tool that could help researchers investigate student use of ST and CT as they build, test, and revise models. We hypothesize that such a tool could help researchers identify which aspects of ST and CT students use more frequently or find challenging. Therefore, we investigate these research questions: How can one characterize patterns of student use of specific aspects of ST and CT as they construct and revise models? Which aspects seem to be more challenging for learners? To address these questions, we developed the ST and CT Identification Tool (ID Tool) to classify instances of students using aspects of ST and CT as they build, test, and revise models.

## Methods

### Study context and data sources

The data used to develop and evaluate our ID Tool came from a high school chemistry unit on evaporative cooling designed to meet NGSS learning goals and enacted at a Midwestern U.S. STEM school. We designed this unit around Project-Based Learning (PBL) principles (Krajcik & Shin, 2022) in which students explore the phenomenon of evaporative cooling, use a driving question and a driving question board and conduct investigations to address the driving question. This unit also centered on students building and revising models of phenomena using an open-source semi-quantitative computational modeling tool called SageModeler (Figure 1). SageModeler is a modeling tool that allows students to construct semi-quantitative models without using formal programming language (https://sagemodeler.concord.org). Students can test these models using a simulation function to generate model output and using graphs constructed from the output or imported from real-world data. To collect data on students building and revising their models, we used 15 hours of screencasts from five pairs of students for a total of 75 hours. Screencasts record the students' actions on their laptop screens and record student audio while they are building and manipulating their computational models.

### Instrument development

Content validity refers to the extent all aspects of our framework align with the literature. For content validity, we conducted an extensive literature review of CT, ST, and modeling, and deconstructed each practice into smaller aspects and sub-aspects (Shin et al., in press). We examined specific aspects of ST and CT to define how students should be able to use their knowledge through five modeling practices. During the development processes, our research team – including experts in science, learning sciences, learning technology, and science education – defined, reviewed, and revised the modeling process and specified aspects of ST and CT in the context of modeling through discussing disagreements and ambiguities, continuing (or updating) our literature review, and teachers' and students' data collected from implementation. Our research team expanded on this work, developing a theoretical framework describing how specific aspects of ST and CT are applied through five distinct modeling practices. These processes confirmed the theory-based modeling process outlined in the framework, ensuring that the ST and CT aspects were operationalized to monitor student involvement in these aspects while modeling.

Construct validity is the extent to which the indicators of our ID Tool measure our intended constructs. To accomplish construct validity, our approach focuses on defining indicators (evidence) clearly and comprehensively and describing measurable (observable) behaviors that are present when learners are utilizing the desired ST and CT aspects through modeling. We first decomposed the various aspects of ST and CT associated with each modeling practice into smaller sub-aspects. For example, the computational modeling practice of "test, evaluate, and debug model behavior" (M4) is supported by the ST aspects of "defining a system" and "predicting system behavior based on system structure" along with the CT aspects of "generating, organizing, and interpreting data," "testing and debugging," and "making iterative refinements" (Table 1). These ST and CT aspects can in turn be broken down into more specialized sub-aspects. Within the CT aspect of "testing and debugging" we identified three key sub-aspects associated with the modeling practice of "test, evaluate, and debug model behavior": "detecting issues in an inappropriate solution," "fixing issues based on the behavior of the artifact," and "confirming the solution using multiple starting conditions."

We then identified specific learner-generated behaviors or knowledge products that can be associated with one or more of these sub-aspects of ST and CT. These behaviors were operationalized as indicators. In this study, we are specifically focusing on indicators associated with "testing and debugging" as the literature strongly suggests that students often have difficulty fully participating in its associated ST and CT aspects (Grover & Pea, 2018). The six indicators associated with the (M4) modeling practice behavior are listed below, along with their respective ST and CT aspects and sub-aspects.

- **4A**. *Analyzing and Sensemaking through Discourse*. ST: defining a system (redefining model structure) and predicting system behavior. CT: testing and debugging (detecting faults and fixing faults)
- **4B**. *Analyzing Model Output: Simulations*. ST: Not applicable (NA). CT: interpreting data (generating data, analyzing data), testing and debugging (detecting faults, confirming solutions), and iterative refinement (verifying solutions)
- **4C**. *Analyzing Model Output: Graphs*. ST: NA. CT: interpreting data (generating data, analyzing data), testing and debugging (detecting faults, confirming solutions), and iterative refinement (verifying solutions)
- **4D**. *Analyzing and Using External Data*. ST: NA. CT: interpreting data (generating data, analyzing data) and iterative refinement (verifying solutions)
- **4E**. *Using Feedback*. ST: defining a system (redefining model structure). CT: testing and debugging (fixing issues) and iterative refinement (making modifications and verifying solutions)
- **4F**. *Reflecting upon Iterative Refinement*. ST: defining a system (redefining model structure). CT: testing and debugging (fixing issues) and iterative refinement (making modifications and verifying solutions).

The modeling research team also reviewed the indicators to determine if they plausibly indicated student use of individual aspects of ST and CT. Once these indicators were reviewed, we further refined and developed them into a four-part classification system (ranging in ascending order from Level 1 to Level 4) to explore the sophistication of student use of these aspects. We then conducted an interrater reliability test for these indicators, which demonstrated a 91.7% agreement (Cohen's Kappa, .87) between two independent coders.

## Data analysis and findings

To code students' collaborative interactions as they built a computational model with SageModeler, we analyzed screencast data using Atlas.ti to organize the data according to the four levels of the ID Tool and to determine the relative frequency of each of these six indicators. The patterns of each group as well as among groups were analyzed to characterize how students used ST and CT aspects during modeling and which aspects seemed challenging for learners. Below we summarize how student behaviors served as evidence for ST and CT by matching indicators from the ID tool with observations from our study.

The computational modeling practice of "test, evaluate, and debug model behavior" occurs as students evaluate their models and consider changes they need to make so that their models more accurately reflect their understanding of the phenomenon. Given the various approaches students can take to evaluating and revising their models, we have identified six observable indicators as evidence that students are involved in different aspects of this modeling practice as listed above as 4A-4F. As students worked on refining their models with their partners, they often discussed specific model relationships and/or broader model behavior. For example, when evaluating their early model, two students had this conversation regarding relationships between variables:

> "Student 1: As the molecular energy increases, that makes the molecular spacing of the substance increase. That's good. And then the spacing of the air molecules increases, this [molecular spacing] stays the same until it [spacing of the air molecules] gets small as there's not a lot of space. Student 2: Makes sense. Student 1: We can change this [the relationship between spacing of air molecules and molecular spacing]. Student 2: Yeah, I don't think that makes sense. Student 1: Is it the other way then? Student 2: Maybe?"

This conversation is an example of indicator 4A, students participating in analyzing and sensemaking of model structure through discourse. At Level 1, students verbalize the changes they made to their model or describe the area of their model they believe needed further revisions, but do not provide reasoning for making these changes or why an area of their model needs improvement. If a student verbalizes reasoning but does not participate in a mutual dialogue with their partner, they are considered performing at Level 2. To progress to Level 3 requires that students engage in a back-and-forth dialogue by providing reasoning for making key changes to their model. Evidence for Level 4 would be a student exchange in which they consider how changes to their model would impact the behavior of the model. Because the students in this example provide a brief amount of reasoning as they are trying to justify the relationship between the spacing of air molecules and molecular spacing and both students contribute to the discourse (although Student 2 does so in a minimalist manner), we consider this to be evidence of students participating at Level 3 for Indicator 4A.

In addition to listening to student discourse, we observe students using model output features to test their model's behavior. SageModeler offers two ways of generating and analyzing model output: manipulating variables using the simulation tool and generating graphs. Both actions produce observable indicators (4B and 4C, respectively). The simulation tool allows students to manipulate the relative amount of each input to test its impact on model behavior (Figure 1A). If students adjust the relative amount of one or more input variables, but do not verbalize their interpretation of this process, we consider this evidence of some attempt to interpret data, even if only at Level 1. Once students begin verbalizing their interpretation of the testing process (either by identifying specific flaws in their model or by stating that their model is functioning in accordance with their expectations), we can map their progress to Level 2. Identifying Levels 3 and 4 requires that students participate in a meaningful back and forth dialogue with either their partner, one or more peers, or a teacher. If the conversation focuses on the smaller aspects of model behavior (centering on a single causal chain), we consider this evidence of Level 3 for interpreting data using the simulation tool, while evidence of Level 4 would require a more holistic discussion of the model (e.g., focusing on how multiple causal chains impact each other).

Students can also use SageModeler to generate graphs that analyze the relationships between two variables from their model (Figure 1B). If students unsuccessfully attempt to make a graph of two variables from the model output, we consider this evidence of Level 1 for indicator 4C. If students successfully make a graph of two variables, but do not discuss their interpretation of this graph, their behavior can be categorized as Level 2. Evidence for Level 3 requires students to participate in a dialogue where they discuss their interpretation of the graph and its implications for those two variables in isolation from the broader context of the model. If students consider the implications this graph has for both these two variables and broader model behavior, this is considered evidence for Level 4 for generating and analyzing data, detecting faults, and confirming and verifying a solution. In Figure 1A, these students used the simulate feature to look at their model output, but neither student verbalized this process in a meaningful way, so we inferred that the students were performing at Level 1 for indicator 4B. The students who made the graph in Figure 1B verbalized their interpretation of the graph, stating, "This graph shows that as IMF increases the potential energy increases but then plateaus. That makes sense to me." This is evidence of Level 3 performance for indicator 4C.

Figure 1A
*Student use of "Simulate" feature in SageModeler.*

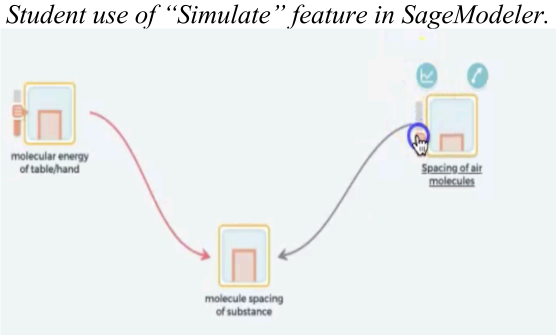
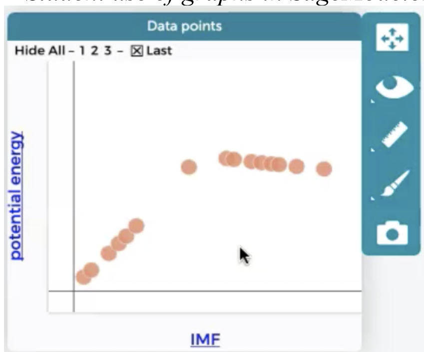



Figure 1B
*Student use of graphs in SageModeler.*




Just as students can analyze their model output to see if their model behaves according to their understanding of the phenomenon, students can also examine external data sources to verify if their models accurately describe the phenomenon as evidenced by indicator 4D. When students superficially refer to the existence of data or loosely reference dubious data sources, they are at Level 1. At Level 2, students reference external data (from real-world observation or specific information provided by instruction or readings) to inform or justify changes made to their models, but do not actively compare these data to their model output. Once students progress to comparing specific pieces of real-world data to their model output, they can be said to be engaging at Level 3. This is particularly evident if they input quantitative external data into the modeling program and directly compare it to their model output (Figure 2). Finally, if students compare and contrast their external data to model output and discuss the validity of the external data, this is evidence of Level 4 performance. In

Figure 2, the students input real-world data from an experiment into SageModeler but did not actively compare these data to their model output, indicating a Level 2 performance.

Figure 2
*Students inputting external data*

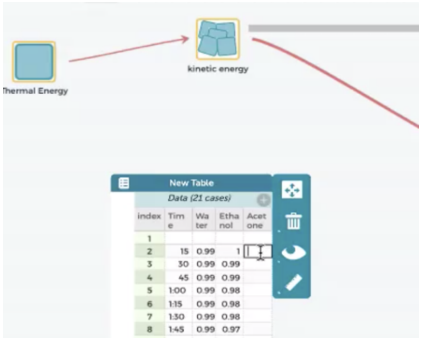
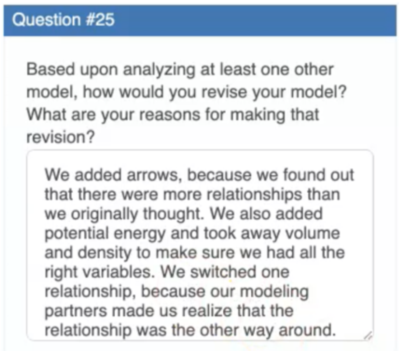



Figure 3
*Student's written reflections on iterative refinement*




Another important way students can receive feedback on their models is through discussions with peers or a teacher, which can inform further revisions, allowing students to engage in using feedback to inform model revisions (Indicator 4E). If the feedback students receive does not inform any changes to their model or prompt further analytical discourse, they are at Level 1. Note that if the feedback they receive is inappropriate and students do not discuss why this feedback was inappropriate their behavior would still be indicative of a Level 1 performance. Students who use this feedback to make changes to their models but have neither a discussion with their partner before making these changes nor test their models after making these changes are at Level 2. Once students use this feedback to either spark an analytical discussion or analyze their model's behavior after making recommended changes, they are operating at Level 3. If students then address the originator of the feedback or have a conversation with another student group about why they made these changes or what new insights have emerged from their testing and debugging of these changes, their behavior can be categorized as being at Level 4. For instance, one of their peers asked another pair to remove "density" as a variable from their model, arguing that it was not necessary to explain the phenomenon. This pair of students then removed the density variable but did not discuss why they were removing this variable.  We classify this performing at Level 2.

Finally, students should be given opportunities to reflect on the changes they have made to their models. Students can participate in the process of reflecting upon iterative refinement (4F) through discussion or through writing as seen in Figure 3. Student level of expertise is suggested by the depth and richness of the insights they exhibit into their own revision process. When students give surface-level feedback on the quality of their models at a given point in time, without considering the changes they have made or the reasons for making these changes, they are performing at Level 1. To infer Level 2 performance, students list specific changes that they have made to their models, but do not provide any detailed reasons for making these changes. Evidence for Level 3 performance requires that students reflect upon specific changes to their models and explain their reasoning behind these changes. Finally, students performing at Level 4 emphasize broader changes that have occurred to their models over a longer period (often across multiple revisions) and provide an explanation as to how their model has evolved. In Figure 3, a pair of students list the changes they have recently made to their models and give specific reasons for making these changes (peer feedback and changes in conceptual understanding). As this reflection focuses on more immediate changes and not broader patterns, it is evidence of Level 3 performance. Overall, these results support Research Question 1 (How can one characterize patterns of student use of specific aspects of ST and CT as they construct and revise models?) as they demonstrate how our ID Tool can be used to identify and classify specific instances of students using ST and CT as they are testing and debugging their models.

Using our ID Tool, we examined the screencasts of five student groups during an evaporative cooling unit. We then compared the relative amount of time (as determined by the number of 10-minute intervals where students were involved in at least once in a respective indicator) these students spent participating in each of the six sets of behaviors we viewed as indicators of involvement with the modeling practice of testing and debugging (Table 2). Incidents were recorded for each 10-minute block and data from all five groups were aggregated to compare the relative amount of time coded for the presence of each indicator. Time points where students were not exhibiting any indicators were excluded from this data set and students could exhibit multiple indicators within one 10-minute block. The results suggest that students spent a large portion of their time using discourse-based strategies to analyze their models (as seen by their high use of Analyzing and Sensemaking of Models through

Discourse [4A], 59.5%) and often utilized the simulation features present within the modeling program. However, these students seem less likely to use external data sources to drive their revision process ([Indicator 4D], 15.1%) and even more hesitant to use graphs to analyze their model output ([Indicator 4C], 3.2%). This suggests that additional scaffolds are likely needed to support student participation in these activities. It is important to note that although students were more likely to participate in Analyzing and Sensemaking of Models through Discourse (Indicator 4A), many exhibited performance only at Level 1 and Level 2 behaviors, which might indicate that performing at higher levels for these indicators were more challenging for them. For instance, we observed several instances where the student in charge of the cursor dominated the sensemaking discussion, while their partner provided minimal feedback or verbal sensemaking support. Overall, these results address Research Question 2 (Which aspects seem to be more challenging for learners?) by suggesting that aspects associated with Indicator 4C are more challenging for students or are less supported by either the curriculum or their teachers compared to aspects associated with Indicators 4A or 4B.

Table 2
*Relative time spent participating in each indicator for all five groups*

| Indicator | Total time Spent* | Relative Percentage |
|---|---|---|
| 4A. Analyzing models: Discourse | 75 (G1:13, G2: 24, G3: 16, G4: 9, G5: 13) | 59.5 |
| 4B. Analyzing Model Output: Simulations | 48 (G1: 4, G2: 15, G3: 12, G4: 5, G5: 12) | 38.1 |
| 4C. Analyzing Model Output: Graphs | 4 (G1: 0, G2: 0, G3: 4, G4: 0, G5: 0) | 3.2 |
| 4D. Analyzing External Data | 19 (G1: 6, G2: 6, G3: 3, G4: 2, G5: 2) | 15.1 |
| 4E. Using Feedback | 36 (G1: 11, G2: 5, G3: 3, G4: 11, G5: 6) | 28.6 |
| 4F. Reflecting upon Iterative Refinement | 41 (G1: 3, G2: 12, G3: 11, G4: 5, G5: 10) | 32.5 |
| Total | 223 (G1:37, G2:62, G3: 49; G4:32 G5:43) | |

Note: * Total number of 10-minute coding blocks where indicator is present at least once. G: group. GX: total coding blocks where indicator is present. Due to students often participating in multiple indicators in a single 10-minute block and the “relative percentage” referring to the percentage of coding blocks where this indicator is present, the sum of the relative percentages does not add up to 100%.

## Conclusions and implications

These results demonstrate how the ID Tool can be used to characterize patterns and challenges of student use of specific aspects of ST and CT as they construct and revise models. While the instrument described in this paper focuses on aspects of ST and CT used during model revision, we have also developed other indicators for ST and CT that need to be validated. A draft of these additional indicators can be found at https://tinyurl.com/2ft6rkza**.** Building off the ST and CT Through Modeling Framework, our ID Tool seeks to connect abstract ideas of student cognition with concrete indicators that can be observed through screencasts, classroom videos, or direct observation of students in classrooms. As each indicator is grounded in specific aspects and sub-aspects of ST and CT, it can be used to track how students are using ST and CT in various learning activities across disciplines. Therefore, this tool can be used to develop future research instruments such as teacher and student interview protocols and classroom observation instruments as well as assist with creating ST and CT integrated learning activities. While the ID tool is primarily designed for research use, it can be modified to be used by teachers to help them identify moments where students are using ST and CT. Overall, our ID Tool represents an important step in developing a meaningful instrument for monitoring student use of ST and CT while constructing and revising models in realistic classroom settings. Further validity studies based on students’ data in various learning contexts are needed to iteratively revise this ID Tool as an evidence-based principled tool to observe student use of ST and CT as they construct and revise computational models. We are also in the process of utilizing this tool to further investigate how students utilize ST and CT aspects within a computational modeling context. Given the increased need and benefits to incorporate ST, CT, and modeling into science education, there is a growing demand for tools that can support researchers, curriculum developers, and teachers in classifying instances of students use of these practices. Our research efforts on the ID Tool seek to further research in ST, CT, and computational modeling and promote the integration of these three research fields to support student learning.

## References


Arnold, R. D., & Wade, J. P. (2015). A definition of systems thinking: A systems approach. *Procedia Computer Science, 44,* 669-678.

Berland, M., & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. Journal of Science Education and Technology, 24(5), 628–647.

Booth-Sweeney, L. B., & Sterman, J. D. (2007). Thinking about systems: student and teacher conceptions of natural and social systems. *System Dynamics Review: The Journal of the System Dynamics Society, 23*(2-3), 285-311.

Bowers, J., Damelin, D., Eidin, E., & McIntyre, C. (2022). Keeping Cool With SageModeler. *The Science Teacher*, *89*(4).

Cronin, M. A., Gonzalez, C., & Sterman, J. D. (2009). Why don't well-educated adults understand accumulation? A challenge to researchers, educators, and citizens. *Organizational Behavior and Human Decision Processes, 108*(1), 116–130.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking: A guide for teachers.* Computing at School. Swindon, UK.

Forrester, J. W. (1971). Counterintuitive behavior of social systems. *Theory and Decision, 2*(2), 109-140.

Gilbert, J. K., & Justi, R. (2016). *Modelling-Based Teaching in Science Education (Vol. 9).* Cham, Switzerland: Springer.

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. In *Computer Science Education: Perspectives on Teaching and Learning in School,* 19 -34. New York, NY: Bloomsbury.

Harrison, A. G., & Treagust, D. F. (2000). A Typology of School Science Models. International Journal of Science Education, 22(9), 1011–1026.

Krajcik, J., & Shin, N. (2022). "Project-Based Learning." In R. K. Sawyer (Ed.), *Cambridge Handbook of the Learning Sciences 3rd Edition*, New York: Cambridge University Press.

Meadows, D. (2008). Thinking in systems: A primer. Chelsea Green Publishing. White River Junction, Vermont.

National Research Council (NRC). (2012). A framework for K-12 science education: Practices, crosscutting concepts, and core ideas. National Academies Press. NGSS Lead States.

Next Generation Science Standards (NGSS) Lead States: For states, by states. (2013). Washington, DC: The National Academy Press.

Schwarz, C. V., Passmore, C., & Reiser, B. J. (2017). *Helping students make sense of the world using next generation science and engineering practices*. NSTA Press.

Schwarz, C. V., Reiser, B. J., Davis, E. A., Kenyon, L., Acher, A., Fortus, D., Shwartz, Y., Hug, B., & Krajcik, J. (2009). Developing a learning progression for scientific modeling: Making scientific modeling accessible and meaningful for learners. *Journal of Research in Science Teaching*, *46*(6), 632–654.

Shin, N., Bowers, J., Roderick, S., Mclntyre, C., Stephens, L., Eidin, E., Krajcik, J., & Damelin, D. (in press). A framework for supporting systems thinking and computational thinking through constructing modeling. *Instructional Science.*

Shute, V. J., Sun, C., Asbell-Clarke, J. (2017). Demystifying computational thinking. Educational Research Review, 22(1), 142–158.

So, H. J., Jong, M. S. Y., & Liu, C. C. (2020). Computational thinking education in the Asian Pacific region.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. Journal of Science Education and Technology, 25(1), 127–147.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Wing, J. M. (2017). Computational thinking's influence on research and education for all. Italian Journal of Educational Technology, 25(2), 7–14. doi:10.17471/2499-4324/922

zu Belzen, A., & Krüger, D. (2010). Modellkompetenz im Biologieunterricht [Modeling competence in biology classes]. *Zeitschrift für Didaktik der Naturwissenschaften, 16*, 41–57.


## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. DRL-1842035 and DRL-1842037.