The Decoding Approach to CT Integration

BY IRENE LEE

Scheller Teacher Education Program
education arcade

The Scheller Teacher Education Program (STEP) Lab
Massachusetts Institute of Technology

IRENE LEE

Made possible by the generous contribution of the National Science Foundation STEM+C Program (award numbers 1934126, 1934039, and 1934112)

Introduction

This paper was written for researchers and educators interested in integrating Computational Thinking (CT) into disciplinary education. We describe a new construct called “Decoding”: why it was developed, which theories and practices it builds upon, how it was developed, how it compares and contrasts with other CT integration approaches, and how it is being implemented and studied in a variety of educational settings. We hope this provides the reader with the foundations for incorporating Decoding in future teaching and research. The research endeavor is one of exploration, inquiry, and discovery. Much of what we now think about Decoding has taken form in the years 2019-2023. The theory of Decoding: what it is and how it might work on a cognitive level, has come together over the four years within three different research projects. We are eternally grateful to the communities of students, teachers, and STEM education researchers who helped us learn about Decoding as they took part in professional development workshops, out-of-school time summer camps, and/or classrooms where the Decoding Approach was implemented and tested.

The Decoding Approach to CT Integration

The integration of computational thinking in education has been promoted by both national computer science education and science education associations (Computer Science Teachers Association, 2011; National Governors Association, 2010; National Research Council, 2012; NGSS Lead States, 2013). Over the past 10 years, researchers have embarked on projects to engage students in CS and CT through integration in core subjects as a means to prepare students with the knowledge, skills, and practices for future endeavors in science and computing fields (Basu et al., 2016; Lee et al., 2011; Schanzer et al., 2018; Sengupta et al., 2013; Swanson et al., 2019). This strategy of integrating CT in disciplinary classes aims to bypass some of the difficulties associated with offering stand-alone CS/CT courses as a new subject to an already crowded school day.

There are many potential benefits of using an integrated approach in Science classes. Since science classes are mandated for all students, integrating CS and

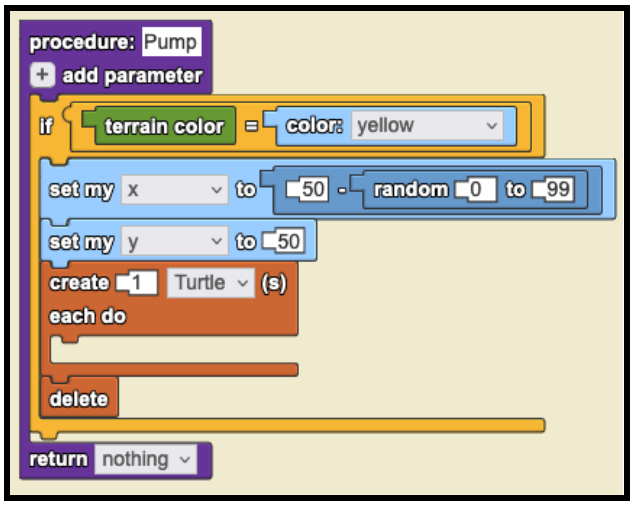
CT in Science classes serves as a way to introduce all students to CS in an equitable fashion (Lee et al., 2017). Interjecting CS and CT into science classes drives the modernization of science curriculum to reflect modern scientific practices such as computer modeling and simulation (Foster, 2006; Sengupta et al. 2013; Tedre and Denning, 2016; Weintrop et al., 2016). The integration also aspires to raise teachers' and students' awareness of the relevance of CS and CT in science and how CS and CT have been key to scientific discovery. Importantly, the integration of CS and CT in science has potential to deepen students' understanding of scientific processes through investigation of mechanisms, or parts of a system working together to generate a phenomenon, abstracted in computer models (Arastoopour Irgens et al., 2020; Schwarz et al., 2014, Wilkerson et al., 2015, 2018).

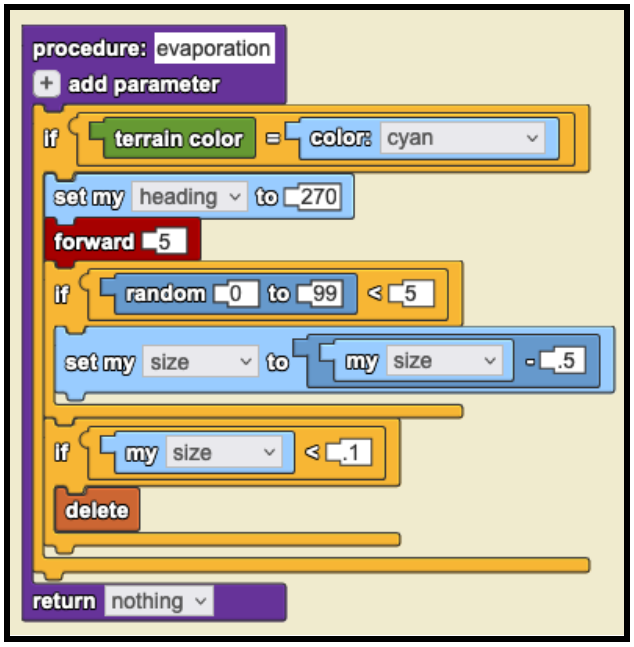
Yet, past efforts to integrate CS/CT in science through a "programming first" in which students learn computer programming to enable them to build models of phenomena have had mixed results. While there are numerous potential benefits of this approach, there have been substantial challenges encountered during its implementation. There is a steep learning curve for science teachers to learn CS and pedagogies for engaging students in CT-rich activities, and gain confidence in teaching computer programming. Science teachers often do not see the "fit" between building computer models and goals for science learning. They have difficulty finding instructional time in a crowded science curriculum to teach CS content and CT practices, especially computer programming. Furthermore, since CS and CT topics and skills are not yet tested in standardized assessments, their value is not currently measured or used to assess teaching. It is within this context that the Decoding Approach was developed as a way to deeply engage in CT without the barriers encountered with expecting students to program models.

A Short History of Decoding

The first conception of Decoding emerged within Project GUTS: Growing Up Thinking Scientifically, an NSF-funded afterschool program that engaged teachers in professional development and youth in making and then using computer models and simulations as tools for scientific inquiry. In the article “Computational Thinking for Youth in Practice”, Lee et al. (2011) described a three-stage progression for engaging youth in Computational Thinking (CT) that was found to be successful in Project GUTS. The progression, called Use-Modify-Create (or UMC for short), describes a pattern of engagement that was seen to support and deepen youths’ acquisition of CT. The Use-Modify-Create progression is based on the premise that scaffolding increasingly deep interactions will promote the acquisition and development of CT. In the “use” stage, students are consumers of someone else’s creation. For example, in the context of a computational science investigation, students run experiments using pre-existing computer models as experimental test beds. Over time they begin to “modify” the model to incorporate new features or change existing features. Through a series of modifications and iterative refinements, the student is in the “create” stage and what was once someone else’s model now becomes one’s own.

While there were many successes in supporting students’ CT with the UMC approach, we found that there was a lot of room for improvement. Through early research on Project GUTS (Lee, 2009), we found that student learning gains were primarily in the areas of CT and agent based modeling, and not in grade-level appropriate science content learning. One observation was that students, when making modifications to their models, were not seeking to make scientifically valid models; imaginary scenarios wherein agents carried out magical behaviors were equally valued by students. Researchers also noticed that student modifications varied in their sophistication. There were modifications that reflected students’ comfort with making changes to code. Student modifications ranged from changing the value of a variable in code (changing a number) to implementing whole new procedures to implement agents’ behaviors. Additionally, there were modifications that ranged from non-mechanistic, such as changing the color or shape of an agent or some other purely visual change that would not impact how the system works, to mechanistic, such as changing an agent’s conditional behavior in a way that can impact the generation of a phenomenon. This type of





mechanistic change was evidence of students' ability to think mechanistically about phenomena. Thus, through the decoding approach, we were attempting to increase students' science learning through their gaining of a mechanistic understanding of processes in scientific models (what causes what) rather than through engaging students in creating their own models through programming.

Prior to modifying models within the Project GUTS curriculum, it was necessary for students to explore and understand the code within the model prior to making modifications. We initially called this type of analysis "Decoding" but have since refined its definition. At that time, when students were asked to decode, we were asking them to describe how an agent's state, typically its position and direction, changed as the code was executed. For example, in Project GUTS' Water Resources module, students were asked to explain the code and what it caused the water droplet to do. In reviewing these blocks of code, we ask students to play the role of the water droplet and interpret the code and act according to what the code instructs you to do. As each block is executed, the student was supposed to describe where the water molecule was in space, as x- and y-coordinates, and which direction it was heading as described below.

When asked to decode this procedure at the left, we expected students to note that the water droplet will get moved when it encounters the yellow part of the pump. The water droplet gets re-positioned at the top of Spaceland with a random x position and a y position at y = 50 (at the top of the window) and a new water droplet is created before the existing water droplet gets deleted. We did not ask

what this process represented in science (that the water droplet gets pumped, (used for some purpose), then goes into the atmosphere.)

Then, in another procedure called “evaporation” shown to the bottom left, we expected students to note that if the water droplet is on a cyan colored part of the terrain, it was to turn so it headed down, and take 5 steps forward. Additionally, there was a small chance that the water droplet will get smaller in size, and if the water droplet gets very small, it gets removed. It is important to note that we did not ask students to link these encoded processes to anything in the real-world. We could have explicitly linked the terrain being cyan to the sky, the heading down and taking steps forward to be precipitation, and the reduction of the size of the water droplet as evaporation.

Through investigating learning outcomes of the UMC progression (Lee, Hsiao & Anderson, 2020), and responding to the feedback provided by science teachers who were implementing the Project GUTS curriculum, it became clear that while the Project GUTS curriculum was a powerful means to engage in CS and CT, it was not significantly improving students’ science learning. Thus, researchers and curriculum designers sought to deepen the links between CT, computer modeling, and science in the curriculum and to identify opportunities that were being overlooked.

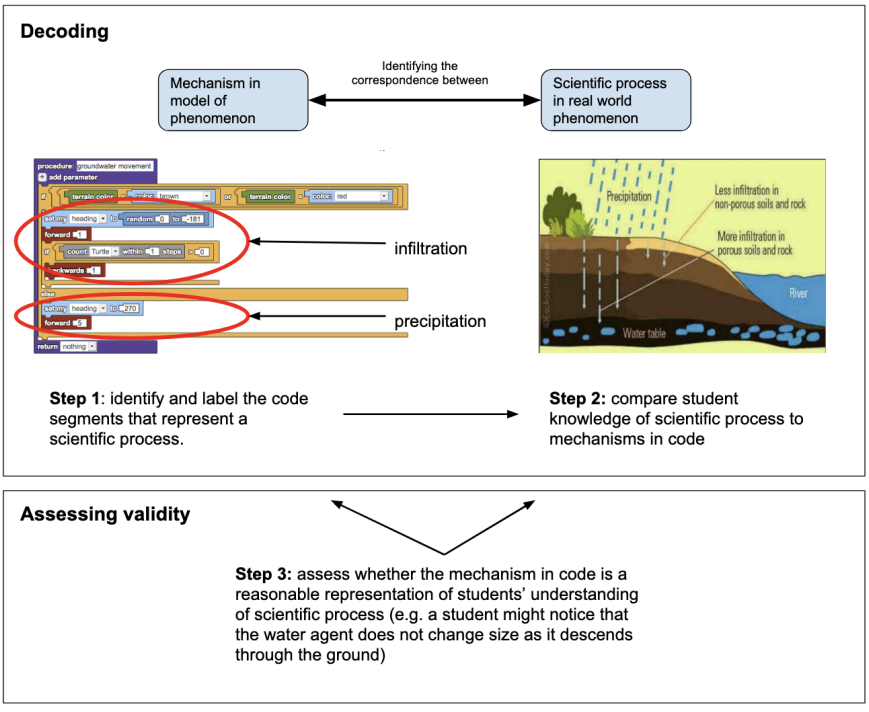


Figure 1. Diagram steps in decoding.

Decoding Explained

In 2019, Lee hypothesized that to enrich students' science learning within a modeling and simulation context, students could engage in identifying the scientific process is being represented in code, examining how the code embodies the scientific processes, and assess the extent to which the coded mechanism reflects what the student knows about the process in science being modeled rather than engaging in programming to create models. This became the new conceptualization of Decoding as making the connection between a mechanism in code and the scientific process it represents. For example, in Figure 1 above, a red oval marks the code that causes an agent to descend (the mechanism in code), matched with the process of precipitation (the process in science being represented). Validation of the model is a subsequent step that requires decoding or the mapping between mechanism in code and process in science it represents. Furthermore, Lee posited that Decoding could expand the UMC progression to form a Use-Decode-Modify-Create progression (Martin et. al., 2020). This expanded progression has potential to enhance students' science learning as well as increase their CS/CT learning.

The steps in Decoding are described using the example from Project GUTS Module on Water Resources (GUTS, 2018) in Figure XX above. Overall, Decoding is operationalized as identifying and assessing the correspondence between mechanisms in code and processes in science. The first step is to identify and label code segments that represent a scientific concept such as infiltration or precipitation. The second step is to compare the identified mechanism with the process in science it is attempting to represent. This comparison, is the basis for assessing the validity of the representation: If the mechanism in code matches the process in science, then one can say the model is "valid"; while if the mechanism in code and the process in science are not aligned, then we can say that the model lacks validity based on what we know about the real world phenomenon being modeled. For example, a student may identify the precipitation code in the model then deem that it does not adequately represent precipitation (rain falling) because rain rarely falls straight down to the ground. We caution that the assessment of a model's validity is contingent on having a correct understanding of the processes in the real world phenomenon as well as having a correct understanding of the encoded mechanism. This proposed practice of validation

mimics the professional practice of model validation in computer modeling and simulation. Scientists and modelers seek to ensure that the model being validated includes the necessary abstractions to simulate a phenomenon, and that the model was programmed correctly and to specification. With respect to a model's validity, Gräbner (2018) coined the phrase "mechanistic adequacy" to describe when a model has the capacity to represent the mechanistic features/structure of the target natural phenomenon.

We present Decoding as a new approach to CT integration that is different from existing approaches. It contrasts with a "programming-first" approach to the integration of CS and CT in Science wherein students would learn to build models through programming. Decoding is also an alternative to teaching via simulation-only wherein the student manipulates a model through a user interface and does not have access to the code underlying the model that generates the simulation. While these visualizations can be helpful illustrations of phenomena, they limit students' assessment of the validity of the model based upon its encoding of scientific processes and limit students' exposure to CT and CS. Additionally, the development of the "Decoding Approach" responded to teachers' feedback that they needed more explicit links between computer modeling and science learning, and they did not have time to teach programming within a science class.

Decoding deepens science learning by engaging students in Mechanistic Reasoning about how and why phenomena occur. Mechanistic Reasoning focuses on the underlying and often invisible mechanisms that cause a phenomenon. It is characterized by hypothetical models of the mechanism being formed and then tested, evaluated, and revised against observations of the phenomenon (Russ, 2008). Determining why a phenomenon occurs requires reasoning about what entities, activities of entities, and causal links that generate an outcome or phenomenon (Machamer, Darden, & Craver, 2000). This practice of analyzing a model for entities, activities, and causal mechanisms is mechanistic reasoning, a central component in science inquiry (Russ, Scherr, Hammer, & Mikeska, 2008). In many scientific fields, explanations require mechanistic descriptions of phenomena, and the discovery and description of mechanisms is key to the practice of science (Machamar et al., 2000; Burian, 1996; Bechtel and Richardson, 1993; Crick, 1988; Brandon, 1985; Kauffman, 1971; Wimsatt, 1972). Students engage in mechanistic reasoning when they are immersed in modeling activities and

pushed to provide explanations of how and why a particular phenomenon happens (Russ et al., 2008; Schwarz et al., 2014).

Seeking to advance students' Mechanistic Reasoning is not new in science education, Russ et. al. (2008) codified MR in a framework that organizes students' inquiry into key elements such as Identifying Entities; Identifying Properties of Entities; Identifying Actions; Identifying Chaining or Cause and Effect relationships either forwards (this led to that) or backward (that happened so there must have been this). Schwarz et. al. (2014) later consolidated the many descriptors used by Russ into three levels of MR where in Level 1 addresses only surface characteristics / non-mechanistic description of WHAT happened; Level 2 addresses co-occurrence or sequencing of actions describing HOW the process or phenomenon played out over time / space; and Level 3 attends to WHY (causal) the phenomenon or process occurred. In our research, we use MR to categorize and reflect on student thinking during decoding. Russ' and Schwarz's frameworks are used in the identification and analysis of mechanistic reasoning in student interviews and classroom utterances.

Connecting CT and Decoding through Mechanistic Reasoning

In theory, CT and Decoding activate complementary cognitive activity and in particular, Mechanistic Reasoning (MR). By engaging students in identifying abstractions such as "who are the agents and what are their properties?"; identifying automations such as "How do the agents interact with other agents and/or the environment, and how do the processes unfold over time?"; and thinking about causality such as "why did the change happen (what caused the phenomenon to happen)?", CT and Decoding address the key components of MR but from different perspectives. Whereas a designer's CT is concerned with questions like "what are the agents I need to include in this model?", an analyst's decoding is concerned with "what agents did the designer of this model include?" Thus, in essence, CT is to design what Decoding is to analysis. Decoding involves understanding how someone else, namely the designer, defined a problem or phenomenon to investigate, and developed a model to solve a problem or understand a phenomenon. (See Table 1)

In the context of Modeling and Simulation	
Computational thinking concerns:	**"Decoding" concerns"**
Taking a 1st person perspective as a modeler	**Taking a 3rd person perspective as an analyst**
Abstraction: What should I include in my model? What should I leave out?	Abstraction: What did the model's creator include in this model? What did the model's creator leave out?
Automation: How should I implement the behaviors in my model? What should happen in the main loop? (forever loop)	Automation: How were behaviors implemented in this model? What did the creator make happen in the main loop?

Table 1. Comparison of the concerns in Computational Thinking vs. Decoding.

The assessment of the validity of the model is a step beyond decoding. To assess the validity of a model, one asks whether the abstractions made were correct, and was the implementation of the abstraction and automation faulty? comparing the coded mechanisms with the real world scientific processes as a way of determining the mechanistic validity of a model. Mechanistic validity concerns how the code does or does not represent the real world phenomenon in terms of processes (rather than or in addition to assessing whether the output data or the visualization produced by running the model is similar to data collected and phenomenon observed in the real world).

Over the past year, researchers developed a new hypothesis of how CT and decoding relate to each other where decoding is defined as comparing mechanisms in code with processes in science they represent. We hypothesized that the Mechanistic Reasoning used during CT in the context of modeling and simulation is the same as the Mechanistic Reasoning used in Decoding although from a different perspective. In CT, one uses MR from the perspective of a designer while designing a model, whereas in Decoding, one uses MR from the perspective of an analyst when analyzing and seeking to validate a model. (See figure 2).

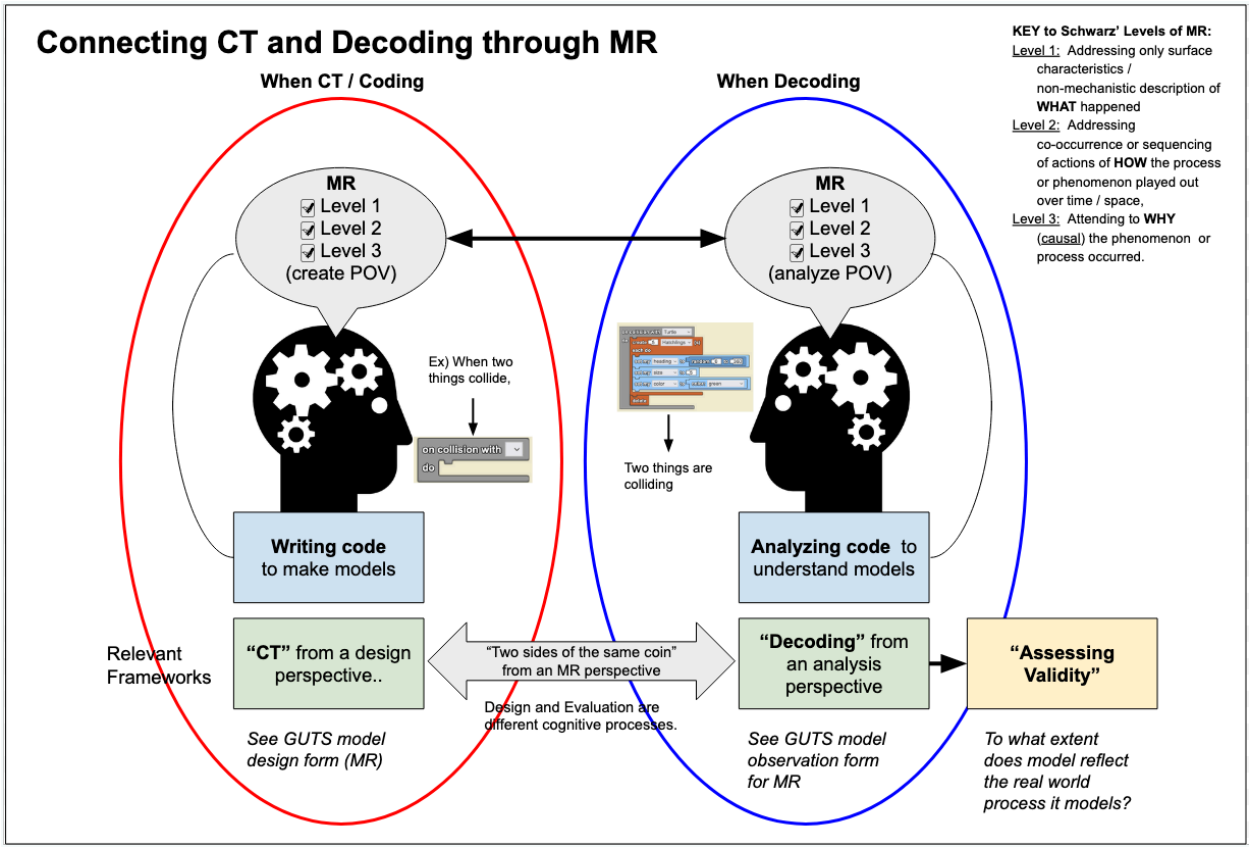


Figure 2. Connecting CT and Decoding through Mechanistic Reasoning.

We found it helpful to use Schwarz et al's Levels of MR framing (2014), to assess Levels of MR used across both CT and decoding. To test if our hypothesis about the relationship between CT and Decoding is correct, we expect to see Level 1 concerns (addressing only surface characteristics or non-mechanistic descriptions) occur in both CT and decoding. In other words, when students plan to build a model or analyze a model made by someone else, do they attend to the entities needed or those already in existence, the attributes of those entities, and the actions of the entities? We also should see Level 2 concerns addressing the co-occurrence or sequencing of actions that describe how the process of phenomenon plays out over time and space. In other words, when students plan or analyze a model, do they describe sequences of actions they see (but without making causal inferences)? And at Level 3, are concerns pertaining to why the phenomenon or process occurs seen both when planning a model and when analyzing a model? If so, we can say that CT and Decoding activate the same type

of thinking, MR, only from different perspectives. If so, we could claim that decoding offers the same opportunities as CT/coding to engage students in mechanistic reasoning. But unlike CT, decoding specifically asks students to determine if the coded mechanism matches the processes in science it is modeling, and thus may invoke, apply, and reinforce students' scientific knowledge. Our theory has helped focus our analysis going forward. Mechanistic reasoning is being used as an analytical framework to compare thought processes involved in building models vs. decoding models across three STEM+C projects. We are applying our coding scheme to capture incidence of MR in the context of building and decoding models.

Supporting Mechanistic Reasoning with Agent-based Modeling

Thinking of a phenomenon in terms of mechanisms that generate it is a scientific practice that has a natural analog in agent based modeling. Describing the activity of entities and activities that produce a phenomenon when decoding an agent based model is a mirror process to describing the agents and mechanisms that need to be implemented to reproduce a phenomenon when creating an agent based model. The mechanism represented as a procedure in code is an instantiation of a theory of the way something works. Researchers have analyzed students' mechanistic explanations in the context of learning about ecology (Dickes et al., 2015; Danish et al., 2011; Hmelo-Silver, Eberbach, & Jordan, 2014). Dickes et al. (2018) found that the lens of mechanistic reasoning can be productively used to identify the process of third-grade students' conceptual development of interdependence in an ecosystem as they engaged in embodied and computer modeling activities. This mirrors findings from prior analyses of children's progressive refinement of explanations about ecosystems through the lens of mechanism (Machamer, Darden, & Carver, 2000; Russ, Scherr, Hammer, & Mikeska, 2008).

Agent based modeling tools like NetLogo and StarLogo Nova were found to support mechanistic reasoning (Hsiao and Lee, 2019). Through analysis of artifact-based interviews, different patterns of usage of agent-based modeling tools yielded different mechanistic understandings. It was found that only when

learners (in this case, middle school teachers) examined the simulation and the underlying code did subjects provide mechanistic explanations of why phenomena emerged. This usage pattern, of moving between the simulation running and the underlying code, enabled subjects to build a web of causal connections across individual and aggregate-levels in complex systems phenomena.

Sharing the Theory of Decoding

To familiarize CT integration researchers, field investigators, and facilitators with this Theory of Decoding we designed an educational sequence that drew on what they already were familiar with. First, we reviewed the roots of the Decoding Approach and the problem of practice it addresses. Then we reviewed an operational definition of decoding as a set of steps teachers could use to engage students in analysis of a model for the processes in science being represented and also assessing a model's validity. Next we reviewed mechanistic reasoning (MR). Though we had introduced it before, it was unclear to what extent people understood it or saw its utility. In order to make MR relatable, we needed to connect it to artifacts and processes that were familiar to the audience. Since many of our team members are familiar with the Project GUTS curriculum, we used it to explain MR.

We handed out Project GUTS' model design and model observation forms and asked the team members to review the questions therein. Then we showed how the questions on the model design and model observation forms were aligned with Russ et al's MR framework with a few exceptions. We showed that there was a close alignment between the Project GUTS' questions about the agents, their behaviors, the environment, etc. and the concerns of MR, namely the entities, the properties of entities, actions of entities, etc. We also pointed out that chaining in MR does not appear in the Project GUTS forms, but is similar to cause and effect relationships in science.

Next, we mapped Russ et al's MR framework to Schwarz et al's Levels of MR. With that understanding, we delved into the Theory of Decoding as seen in figure XX. above. In the diagram, the left red oval represents CT in the context of model

making, while the right blue oval represents Decoding in the context of analyzing a model. Importantly, we show how all three levels of MR are activated in each type of thinking (CT and Decoding). We claim that the two types of thinking are mirrors to one another. This theory was shared with the attendees of the Decoding Conference in November 2023. We heard from participants that the way we introduced MR was relevant and helpful for them.

Testing the Theory of Decoding

The Theory of Decoding is being tested in three NSF-funded STEM+C projects around Decoding:

- Massachusetts Institute of Technology's Making Sense of Models (MSM), NSF STEM+C award #1934126, developed and tested an interdisciplinary curriculum for 6th grade students offered during the regular school day that interweaves math, science and CT through decoding models of scientific phenomena.
- American Museum of Natural History's Decoding Urban Ecosystems (DecodeNYC), NSF STEM +C award #1934039, designed a new curriculum and offered summer institutes for middle school students that engaged them in decoding at each step in a Use-Modify-Create progression.
- EDC's Computational Science Pathway Option for Massachusetts High School Students (Science+C), NSF STEM+C award #1934112, developed three new +C courses for high school students that introduced them to how scientific models embed processes in science and how they can be used and modified to test hypotheses about real world phenomena.

The project's principal investigators incorporated decoding as an approach to deepen both CT and disciplinary learning. Within each project, Decoding was defined similarly as mapping between mechanisms in code and processes in science, but conceptualized and instantiated Decoding in each curriculum differently:

In the MSM curriculum, Decoding was a mirror to CT and the mechanism learned in a math context was transferred to a science context. Initially in each unit, MR was used by 6th grade students when they were using CT to build simple

mechanisms in the math portion of the curriculum, then analogous MR was used when decoding a scientific model that uses the same mechanism that was seen before in math. For example, a bounce or stick mechanism was coded in the math portion of a 6th grade class wherein when a ball impacts a surface it either bounces off (if some condition exists) or sticks to the surface (if some other condition exists). Then in the science portion of class, students learn about reflection and absorption, link it to the bounce and stick mechanisms learned earlier, and transfer the understanding of the bounce and stick mechanisms to the reflection or absorption of solar energy based on the surface's color in a scientific model.

In the Decode-CT curriculum taught in the DecodeNYC summer program, 7th and 8th grade students engaged in decoding models at each phase in the Use-Modify-Create progression. In the use phase, while running a model to produce a simulation, students looked for the code that instantiated specific behaviors. In the modify phase (while working in code), reading and decoding models provided students with the knowledge of how the code works and how it represented processes in science as a preliminary step before making modifications. Likewise, in the create phase (while working in code), students were supported in reading and decoding existing scientific models before incorporating new actions and behaviors from the existing model into their own model.

In the three high school Science+C curricula (Biology+C, Chemistry+C, and Physics+C) a Use-Decode-Modify progression was used to engage students in decoding. The first lesson of each unit, the "Use" lesson, linked the science concept to a real world phenomenon or scenario selected for its familiarity to students, reviewed key science concepts, and investigated how the model when run as a simulation mimicked the phenomenon, or not. In the second lesson, the "Decode" lesson, students uncovered where and how concepts from science were instantiated in code and discussed potential refinements (modifications) that could be made to the model to make the simulation more realistic or to add additional functionality. Finally, in the third lesson, students were asked to choose a modification to make to the model and instructions on how to make the modification (given a choice of levels of scaffolding they preferred), then were guided as they modified then tested the model's code.

Additionally, expert facilitators from the Project GUTS (NSF ITEST #0639637) community contributed to the description of how Decoding can be integrated into the Project GUTS curriculum to deepen science learning.

Conclusions

Research on CT integration in disciplinary settings has stagnated around a two approaches namely the “programming first” that has numerous barriers to implementation such as the poor fit with science learning; the lack of preparation of science teachers to teach programming; and the immense amount of time it takes to support students as builders of models, and the use-only approach wherein simulations are run to generate data and/or illustrate scientific concepts, This approach limits students’ ability to look “under the hood” and learn how mechanisms in code generate phenomena in the simulation and how those mechanisms in code might correspond to processes in science. The Decoding approach was proposed as a solution to CT integration that bypasses barriers of the “programming first” approach and that deepens science learning by making strong links to science content and processes through decoding of phenomena.

But in the Decoding approach we anticipate there will be tradeoffs as well. While students get to inspect the code that runs simulations and map it to processes in science, without prior learning experiences in computer science, interpreting the code as processes is difficult for most students. Additionally, to validate a model, students will need a solid knowledge of how the phenomena is generated in the real world. Students will have less agency in the decoding approach than in the programming first approach, and this may limit the popularity of the decoding approach with students. Our pilot teachers have embraced the Decoding Approach but require extra professional development to understand mechanistic reasoning and how it can be used to bridge code and science to promote the understanding of processes in both domains (code and science). We firmly believe that implementing the Decoding approach requires a commitment on behalf of the school or district to allow teachers to implement the curriculum fully to derive the learning benefits of Decoding. This paper aims to inform researchers and practitioners about the Decoding approach with the hopes that others will incorporate and test the Decoding approach with K-12 students.

Acknowledgments

The author would like to thank the National Science Foundation for its generous support of this line of research through the following awards: MIT's Making Sense of Models, #1934126; AMNH's Decoding Urban Ecosystems, #1934039; EDC's Computational Science Pathway Option for MA HS Students, #1934112, as well as NSF DRK12 #1503383 / 1639069; and NSF ITEST #0639637. The author would also like to thank Eric Klopfer, Judy Perry and for making space for this work at the MIT STEP Lab.

REFERENCES

Arastoopour Irgens, G., Dabholkar, S., Bain, C., Woods, P., Hall, K., Swanson, H., Horn, M., and Wilensky, U. (2020). Modeling and Measuring High School Students' Computational Thinking Practices in Science. Journal of Science Education and Technology, v29 n1 p137-161.

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. Research and Practice in Technology Enhanced Learning, 11(13), 1–35.

Bechtel, William and Robert C. Richardson (1993), Discovering Complexity: Decomposition and Localization as Strategies in Scientific Research. Princeton: Princeton University Press.

Brandon, Robert (1985), "Grene on Mechanism and Reductionism: More Than Just a Side Issue", in Peter Asquith and Philip Kitcher (eds.), PSA 1984, v. 2. East Lansing, MI: Philosophy of Science Association, 345-353.

Burian, Richard M. (1996), "Underappreciated Pathways Toward Molecular Genetics as Illustrated by Jean Brachet's Cytochemical Embryology", in Sahotra Sarkar (ed.), The Philosophy and History of Molecular Biology: New Perspectives. Dordrecht: Kluwer, 671-85.

Computer Science Standards: Revised 2011. Technical Report. Association for Computing Machinery, New York, NY, USA.

Crick, Francis (1958), "On Protein Synthesis", Symposium of the Society of Experimental Biology 12: 138-167.

Danish, J.A., Peppler, K., Phelps, D. et al. Life in the Hive: Supporting Inquiry into Complexity Within the Zone of Proximal Development. J Sci Educ Technol 20, 454–467 (2011). https://doi.org/10.1007/s10956-011-9313-4

Danish, J., Saleh, A., Andrade, A. et al. Observing complex systems thinking in the zone of proximal development. Instr Sci 45, 5–24 (2017). https://doi.org/10.1007/s11251-016-9391-z

Dickes, A. & Farris, A. (2019). Beyond Isolated Competencies: Development of Computational Literacy in an Elementary Science Classroom. In: Sengupta, P., Shanahan, M.-C., & Kim, B. (Eds). Critical, Transdisciplinary and Embodied Approaches in STEM Education . (pp 131 - 150). Springer: New York.

Dickes, A., Farris, A., and Sengupta, P. (2020). Sociomathematical Norms for Integrating Coding and Modeling with Elementary Science: A Dialogical Approach. Journal of Science Education and Technology 29(1) 35-52. DOI: 10.1007/s10956-019-09795-7.

Dickes, A., Sengupta, P., Farris, A. V., & Basu, S. (2016). Development of mechanistic reasoning and multi-level explanations in 3rd grade biology using multi-agent based models. Science Education, 100(4),734–776.

Foster, I. (2006). A two-way street to science's future. Nature, 440(March 23), 419. https://doi.org/10.1038/440419a.

Graebner, Claudius (2018) 'How to Relate Models to Reality? An Epistemological Framework for the Validation and Verification of Computational Models' Journal of Artificial Societies and Social Simulation 21 (3) 8 <http://jasss.soc.surrey.ac.uk/21/3/8.html>. doi: 10.18564/jasss.3772

Hmelo-Silver, C. E., Eberbach, C., & Jordan, R. (2014). Technology-supported inquiry for learning about aquatic ecosystems. Eurasia Journal of Mathematics, Science & Technology Education, 10(5), 405 – 413.

Hsiao, L., Lee, I., & Klopfer, E. (2019). "Making sense of models: How teachers use agent-based modeling to advance mechanistic reasoning." British Journal of Educational Technology 50, 5 (July 2019): 2203-2216 ©2019 British Educational Research Association

Kauffman, Stuart A. (1971), "Articulation of Parts Explanation in Biology and the Rational Search for Them", in Roger C. Buck and Robert S. Cohen (eds.), PSA 1970. Dordrecht: Reidel, 257-272.

Lee, I. (2011). Final Project Report for NSFAYS Award #0639637 Project GUTS: Growing Up Thinking Scientifically. Available at https://tinyurl.com/ProjectGUTS-finalreport2011.

Lee, I., Martin, F. Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., and Werner, L. (2011). Computational Thinking for Youth in Practice. ACM Inroads 2 (1): 32-37.

Lee, I., Grover, S., Martin, F., Pillai, S., and Malyn-Smith, J. (2019). Computational Thinking from a Disciplinary Perspective: Integrating Computational Thinking in K-12 Science, Technology, Engineering, and Mathematics Education. J Sci Educ Technol. Dec. 2019. doi:10.1007/s10956-019-09803-w

Lee, I., Hsiao, L. and Anderson, E. (2020). Teachers' knowledge and skills in CT and their enactment of a CT-rich curriculum. In Mouza C., Yadav, A. and Leftwich, A. (Eds.) Preparing Teachers to Teach Computer Science. Information Age Publishers.

Machamer, P., Darden, L., & Craver, C. F. (2000). Thinking about mechanisms. Philosophy of science, 67(1), 1-25. doi:10.1086/392759

Martin, F., Lee, I., Lytle, N., Sentence, S. and Lao, N. (2020). Extending and Evaluating the Use-Modify-Create Progression for Engaging Youth in Computational Thinking. ACM SIGCSE 2020.

National Governors Association Center for Best Practices, Council of Chief State School Officers (2010). Common core state standards for mathematics. National Governors Association Center for Best Practices, Council of Chief State School Officers, Washington, DC.

National Research Council. (2012). A framework for K-12 science education: Practices, crosscutting concepts, and core ideas. Washington, DC: The National Academies Press. https://doi.org/10.17226/13165.

NGSS Lead States (2013). Next generation science standards: for states, by states. The National Academies Press. Washington, DC.

Project GUTS Curriculum. Module on Water Resources (GUTS, 2018). Downloadable at https://TeacherswithGUTS.org/resources.

Russ, R. S., Scherr, R. E., Hammer, D., & Mikeska, J. (2008). Recognizing mechanistic reasoning in student scientific inquiry: A framework for discourse analysis developed from philosophy of science. Science Education, 92(3), 499-525. doi:10.1002/sce.20264

Schanzer, E., Fisler, K.,& Krishnamurthi, S. (2018). Assessing bootstrap: Algebra students on scaffolded and unscaffolded word problems. In Proceedings of the 49th ACM technical symposium on computer science education - SIGCSE'18 (pp. 8–13). Baltimore, Maryland: ACM Press. https://doi.org/10.1145/3159450.3159498.

Schwarz, C., Lee, M., and Rosenberg, J. (2014). Developing mechanistic explanations of phenomena: Case studies of two fifth grade students' epistemologies in practice over time. International Conference of the Learning Sciences, ICLS 2014.

Schwarz, C. V., & White, B. Y. (2005). Metamodeling knowledge: Developing students' understanding of scientific modeling. Cognition and Instruction, 23(2), 165-205.

Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, D., Owens, B. B., Stephenson, C. and Verno, A.. 2011. CSTA K--12 Computer Science Standards: Revised 2011. Technical Report. Association for Computing Machinery, New York, NY, USA.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: a theoretical framework. Education and Information Technologies, 18(2), 351–380.

Swanson, H., Anton, G., Bain, C., Horn, M., & Wilensky, U. (2019). Introducing and Assessing Computational Thinking in the Secondary Science Classroom. In Computational Thinking Education (pp. 99-117). Springer, Singapore.

Tedre, M. and Denning, P. (2016). The long quest for computational thinking. In Proceedings of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling '16). Association for Computing Machinery, New York, NY, USA, 120–129. DOI:https://doi.org/10.1145/2999541.2999542

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.

Wilkerson-Jerde, M., Wagh, A., & Wilensky, U. (2015). Balancing curricular and pedagogical needs in computational construction kits: Lessons from the deltatick project. Science Education, 99(3), 465–499.

Wimsatt, W. C. (1972). Complexity and Organization. PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association, 1972, 67–86. http://www.jstor.org/stable/3698961