CO-DESIGNING TEACHING AUGMENTATION TOOLS TO SUPPORT THE INTEGRATION OF PROBLEM-BASED LEARNING IN K-12 SCIENCE

By

Nicole Marie Hutchins

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

December 17, 2022

Nashville, Tennessee

Approved:

Gautam Biswas, Ph.D.

Shilo Anders, Ph.D.

Catie Chang, Ph.D.

Shuchi Grover, Ph.D.

Maithilee Kunda, Ph.D.

Jessica Watkins, Ph.D.

For Ted.

ACKNOWLEDGMENTS

First, I would like to thank my PhD advisor, Gautam Biswas. Dr. Biswas took a chance on a school teacher from the U.S. Virgin Islands and I am forever grateful for that. I also want to thank my PhD committee, Shilo Anders, Catie Chang, Shuchi Grover, Maithilee Kunda, and Jessica Watson for your support and mentorship throughout this dissertation journey.

Dr. Grover, along with Doris Chin, Rachel Wolf, and Kristen Pilner Blair have provided me with continuous mentorship over the past five years that I am truly grateful for. Your guidance and feedback as inspiring female researchers in our field has motivated me to continue and do better.

Thank you to the incredible, inspiring teachers that I have had the opportunity to work with over the years. I wish I could list you all, but IRB says no.

Thank you to my research mentors Akos Ledeczi, Miklos Maroti, Brian Broll, Gayathri Narasimham, Kevin McElhaney, Jennifer Chiu, Satabdi Basu, Nonye Alozie, Arif Rachmatullah, Luke Conlin, Kelly Mills, and Shannon Campe. I have had the privilege to learn from and work with an amazing group of researchers during my time, and I am forever grateful for your support.

Thank you to my fellow graduate students and lab members Haley Adams, Deirdre Scully, Mona Emara, Ramkumar Rajendran, Allison Moore, Ibrahim Ahmed, Anabil Munshi, Ben Yett, Avisek Naug, Naveed Mohammed, Caitlin Snyder, Marian Rushdy, and Shitanshu Mishra. I have learned from you all.

To my friends and family, this would not have been possible without you.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

The increasing embrace of situative, student-centered perspectives on learning in the context of today's technology-enhanced, K-12 science classrooms, has magnified the need to more deeply understand how students construct knowledge and develop problem-solving skills. As educators and researchers, we face new opportunities as well as new challenges on how we can support students' learning and problem-solving processes in these technology-enhanced environments. Science, Technology, Engineering, and Mathematics (STEM) curricula and accompanying technologies, often developed as part of the wider-spread adoption of the Next Generation Science Standards (NGSS), must support the capture, evaluation, and presentation of student learning and performance from multiple perspectives (e.g., conceptual knowledge growth, applications of practice, and communication skills) and in ways that support *actionable* facilitation by teachers in line with the intended constructivist curriculum design (Baker et al., 2020; Gomoll et al., 2022).

The field of learning analytics has made significant progress in developing and applying cutting edge technologies that support our understanding of student learning behaviors (e.g., machine learning-based predictive analytics; Gašević et al., 2015), provide personalized feedback to students (e.g., intelligent tutoring systems; Graesser et al., 2012), and visualize performance metrics for key stakeholders (e.g., dashboards; Tissenbaum and Slotta, 2012). However, as these artificial intelligence and machine learning models often operate from a purely behavioral perspective, an intuition is that applications of these models represent a return to teacher-centered learning design. Applying these analytics to support constructivist methodologies, such as problem-based learning (PBL; Hmelo-Silver, 2004), in which students learn through the understanding, knowledge construction and problem solving, evaluation, and communicating solutions of real world problems, can create situations in which the feedback generated may (1) provoke "lethal mutations" (Brown and Campione, 1996) in the intended learning design (e.g., by directing students toward single problem-solving pathway) or (2) inadvertently add labor for teachers by inserting new and unfamiliar workflows (e.g., determining how to translate identification of a conceptual knowledge gap into guidance that gives students agency in identifying and understanding the needed concept without directly "filling a gap"). As such, in order to support PBL approaches such as computational modeling and engineering design in K-12 science classrooms, a research priority is to develop methodologies that balance our overriding student-centered orientations in education with the behaviorally-driven response systems that AI methodologies are designed to support, simultaneously preparing teachers for the integration of these learning approaches in their science classrooms. As much as teachers need support from AI technologies, it is very important to remember that teachers need to be an integral part of the iterative instruction and learning loop.

It is within this context that the present dissertation is situated. In particular, this work leverages a participatory design research approach for the co-design of explainable, actionable insights into student learning and problem solving in computational modeling and engineering design environments, such that there is support for evidence-based pedagogical responses supported by an accompanying teacher dashboard, built around a novel methodology for preparing teachers to integrate PBL in their science classrooms.

1.1 Technology-enhanced, Problem-based Learning in K-12 Science Classrooms

The K-12 science classroom is evolving. Coinciding with rapidly-changing technological advancements, state and national standards have made prominent the need to integrate computing and engineering problem solving into science classrooms to better prepare our students for future success (NRC, 2014; NGSS, 2013). Engaging students in these integrated learning experiences promotes interdisciplinary critical thinking and computer-related skill development (Wing, 2006; Grover and Pea, 2013; Weintrop et al., 2016) and immerses students in problem-based, socially-relevant inquiry that can promote interest in relevant STEM domains (NRC, 2012; Hutchins et al., 2020a). This dissertation centers its integrated PBL curriculum on this target educational need.

Initial work on building technology-enhanced learning environments to support this initiative has been successful in engaging students with scientific inquiry and engineering problem-solving opportunities, in addition to integrated learning across multiple domains (Weintrop et al., 2016; McElhaney et al., 2020). In our work, we have not only seen promise in integrating computational and critical thinking strategies across science, computing, and engineering (e.g., Zhang et al., 2017; Hutchins et al., 2019a), we have also demonstrated that students can transfer problem-solving skills to new domains following a computational modeling curriculum (Hutchins et al., 2020a,c).

However, students and their teachers often face significant difficulties with such integrated curricula, which warrants further investigation and development to facilitate instruction and learning. From the student perspective, research has demonstrated students have issues with (1) translating their developing scientific knowledge into computational forms (Sengupta et al., 2013), (2) understanding the mathematical and causal relations between variables (Sengupta and Farris, 2012; Bolger et al., 2012), and (3) applying key computational practices (Basu et al., 2016b). More work is needed to improve instruction and support students as they integrate their developing scientific ideas during computational modeling and engineering design.

From the teacher perspective, little guidance is provided on how teachers may support students in meeting the expectations of integrating across disciplines (e.g., NRC, 2014; NGSS, 2013), especially considering many science teachers do not have computing, pedagogical, and content knowledge for technology-enhanced computational modeling of scientific phenomena and linking science models to engineering problem solving (Bocconi et al., 2016; Cunningham and Carlsen, 2014). There is also limited research that examines how students' STEM knowledge evolves and the difficulties they face in integrating science, computing, and engineering concepts and practices through *multiple linked representations* (Hutchins et al., 2021a). Furthermore, there is a dearth of evidence-based, formative assessment approaches to comprehensively evaluate student learning in these combined domains (Basu et al., 2018). To overcome this, we need to improve methodologies that link learning performance with scientific idea development, model construction, and problem-solving processes (Zhang et al., 2021). Representing and providing teachers with such evidence-based feedback, in ways that they can easily interpret and act on, is critical as research has demonstrated the importance of teacher engagement in students' developing ideas and strategies in supporting integrated STEM learning (e.g., Robertson et al., 2016; Crismond and Adams, 2012).

1.2 Pedagogically-Supportive Learning Analytics

Learning analytics is defined as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of *understanding* and *optimising* learning and the environment in which it occurs" (Siemens and Baker, 2012, p.2, emphasis added). This framework situates learning analytics as a promising approach to support teachers in their classroom instructional and assessment activities.

Advancements in technology-enhanced learning environments have provided a plethora of student log data from a variety of contexts, including programming (Grover et al., 2017), inquiry (Käser and Schwartz, 2020), scientific model building (Leelawong and Biswas, 2008; Basu et al., 2013; Hutchins et al., 2020a), and design (Xing et al., 2021; Vieira et al., 2016) tasks. Learning analytics methods applied to this log data have supported increased *understanding* of how learning occurs through the evaluation of student problem-solving processes, metacognitive strategies, and inquiry or design strategies (Fischer et al., 2020).

Simultaneously, advancements in technology-enhanced learning environments have also spurred the development of student and teacher dashboards to present the resulting learning analysis (e.g., Holstein et al., 2019; Martinez-Maldonado et al., 2016; Prieto et al., 2019; Diana et al., 2017), thereby targeting the *optimization* of learning in these environments. However, there is a dearth of evidence on the impact of these dashboards on pedagogical augmentation and the subsequent impact on optimizing student learning (Wiley et al., 2020). The development of such pedagogically-supportive analytics that improve student learning remains a research priority.

In terms of supporting teachers' *understanding* of learning occurring through technology-enhanced environments, limitations exist in the development of learning analytics measures and the subsequent connecting of these measures to the complex task of providing teachers with actionable insight into the disciplinary-substance of their students' ideas, critical-thinking processes, and knowledge development as they work on their learning and problem-solving tasks (Baker et al., 2020; Wiley et al., 2020). These include:

1. lack of sufficient analytics measures that leverage students' activity logs to characterize their learning of domain-specific concepts and practices, and present them to teachers in a way that they can develop actionable interventions to support students (Bakharia et al., 2016; Hernández-Leo et al., 2019); and

2. lack of teacher involvement in the design and development of learning analytics; complexities of data-driven analytics can make it challenging for teachers to meaningfully contribute to the design and development process (e.g., Martinez-Maldonado et al., 2016).

In order for a teacher to *understand* and *optimize* learning, the teacher must be able to interpret the learning analytics presented, and the learning analytics must provide information to support and justify actionable responses by the teacher (Wiley et al., 2020). Recent efforts have leveraged participatory design approaches to integrate teacher feedback in the design and development of the learning analytics (e.g., Holstein et al., 2019; Prieto et al., 2019) and to support improvements in the interpretation and presentation of the learning analytics to enable actionable responses. These participatory design efforts have:

1. demonstrated improved teacher response to students' developing ideas (Bywater et al., 2019),

2. supported effective pedagogical adjustments that better engage teachers in students' developing ideas (Wiley et al., 2020), and

3. increased teacher agency in the choice of data visualization methods that improve sensemaking and interpretation of student data (Ahn et al., 2021).

Thus far, these participatory efforts have not targeted the development of learning analytics and associated dashboards during open-ended and problem-based learning (Bywater et al., 2021), such as computational modeling in science. As such, more work is needed to understand how to best design and deploy learning analytics to support teachers in interpreting and understanding the learning and knowledge development processes that occur in these contexts.

Finally, given that the majority of science teachers do not have experience in the integration of computing and engineering (see Section 1.1), consideration must be made regarding (1) the application of participatory design methods that engage in the experiences, preferences, and concerns of teachers, and (2) the design, implementation, and visualization of learning analytics that are *interpretable* and *actionable* for teachers with limited computation and engineering design experience.

The empowerment of teachers as drivers of active learning and knowledge construction among their students using advanced technology-enhanced, problem-based STEM+C learning environments is critical. This situation calls for the co-design and development of pedagogical supports that:

1. align with current educational reform efforts (e.g., NGSS, 2013),
2. leverage the unique perspectives and practices of teachers and students (Holstein et al., 2019; Prieto et al., 2019), and
3. visualize student learning and problem-solving processes in a manner that is interpretable and actionable to the teacher (Wiley et al., 2020).

To address these points, this thesis establishes three primary goals:

1. the principled, evidence-centered design of the technology-enhanced, problem-based learning environment and accompanying curriculum and assessments to ensure alignment with national and state standards,
2. applications of novel participatory design techniques to integrate teachers' backgrounds and feedback into the design and development process, and
3. the development of learning analytics and an associated teacher dashboard that help teachers follow students' progress in their model-building and problem-solving tasks and support effective pedagogical decision-making and responses that leverage students' developing scientific ideas and help students' advance in their tasks.

1.3 Scope and Contributions of this Dissertation Research

This dissertation leverages a participatory design research approach for the co-design of a teacher dashboard to support evidence-based pedagogical responses during problem-based computational modeling and engineering design in middle school science. This dissertation research hypothesizes that systematic co-design and visualization of learning analytics targeting student learning and problem solving when building computational models in science and solving engineering problems while using technology-enhanced, problem-based learning environments will lead to instructional and learning benefits for the teachers and their students.

The research presented in this dissertation has evolved in two primary phases of work, each of which has produced a number of research contributions.

Phase 1: Understanding Student Learning and Problem Solving in C2STEM and SPICE. The first phase of research addressed the lack of sufficient assessments and analytics measures to characterize students' learning and problem-solving behaviors in technology-enhanced STEM environments. This phase began with the co-design, development, and evaluation of the Collaborative, Computational STEM (C2STEM) learning environment and accompanying integrated curriculum and assessments (Hutchins et al., 2020a). The initial design and implementation of C2STEM, including the development of a novel domain-specific modeling language (DSML) approach for learning through computational modeling in science (Hutchins et al., 2020b), and the design and execution of fourteen research studies ($n > 1200$ student participants, Vanderbilt IRB approved processes conducted; see Appendix B) were performed collaboratively by a research team that included Vanderbilt University, Stanford University, Salem State University, SRI International, and ETS researchers. Students used C2STEM to learn by building, testing, debugging, and using computational models of science processes in Physics, Marine Biology, and Genetics. Analysis of student work, including computational modeling tasks, log data, summative and formative assessments, and video and discourse data allowed us to deepen our understanding of how students construct their integrated science and computing knowledge (Hutchins et al., 2020a, 2019b), communicate their knowledge and processes to collaboratively problem solve (Emara et al., 2021; Hutchins et al., 2018; Hutchins et al., 2021b; Snyder et al., 2019a, 2022), apply problem-solving strategies to construct and debug their computational models (Hutchins et al., 2019a, 2021b), and transfer problem-solving processes to solve problems in new domains (Hutchins et al., 2020c). Additional findings can be found in the List of Publications in Appendix A.

Continuing this approach, the Science Projects Integrating Computing and Engineering (SPICE) curriculum was developed and implemented as a collaboration with researchers from Vanderbilt, SRI, University of Virginia, Digital Promise, and Washington State University. The research team collaborated with four participating teachers in the design and development of the integrated science, computing, and engineering curriculum and in completing modifications to the C2STEM system. Initial findings have allowed us to deepen our understanding of how computing can serve as a bridge for the integrated learning of science and engineering (Zhang et al., 2022; Basu et al., 2022), the impact of multiple linked representations on student learning (Hutchins et al., 2021a), and the roles problem-solving strategies have on learning in each domain (Zhang et al., 2022). Moreover, results have allowed us to examine methods for automating embedded assessment analysis to support formative feedback (Cochran et al., 2022). This work served as the basis for identifying student learning and behaviors that may be leveraged and acted upon by teachers to engage in their students' learning and problem-solving processes.

Phase 2: Co-Designing a Teacher Dashboard to Support Teacher Noticing and Response During Technology-Enhanced, Problem-Solving in STEM. Leveraging the curriculum, technology, and analysis from Phase 1, Phase 2 targeted the co-design, development, and implementation of the Responsive Instruction for STEM Education (RISE) Dashboard to address the lack of methods for meaningfully integrating teacher insights and preferences in the design and development of educational technology, and in particular, learning analytics and teacher feedback tools (e.g., dashboards). This participatory design approach included three core stages: Needs Analysis and Low-Fidelity Prototyping, High-Fidelity Prototype-Supported Teacher Professional Development, and Planning Period Simulations with RISE. During this time, we partnered with 9 teachers, 3 with SPICE experience, 1 with C2STEM experience, and 5 with no SPICE or C2STEM experience. We adapted established human-computer interaction (HCI) techniques to meaningfully collect teacher insight, needs, concerns, and preferences in the design and implementation of learning analytics and accompanying visualizations through the dashboard. This process allowed us to examine (1) visualization needs for efficient interpretation and reflection, (2) co-construction processes needed for teacher-researcher partners to be on the "same page" of what constitutes actionable insight to support students' problem-based learning, and (3) technology resources to transition from interpretation to pedagogical response construction. Finally, we examined teacher noticing and response of student learning and problem-solving behaviors during this problem-based, technology-enhanced curriculum by implementing a series of simulations that used prior student and class data from previous SPICE implementations and co-designed visualizations and feedback (Hutchins and Biswas, 2022).

In summary, this dissertation contributes to advancing research at the intersection of HCI, AI, and Learning Sciences in the following ways:

- The use of a system of assessments approach combining summative and embedded assessments to track students domain-specific knowledge and skills across the curriculum.
- The application of quantitative analytical and machine learning techniques to analyze and understand student learning and problem-solving process development from the log and otherwise captured data from the technology-enhanced, PBL environment.
- Novel participatory design approaches adapted from established HCI techniques to elicit teacher feedback and preferences on feedback and visualizations needed to conduct evidence-based responses during PBL throughout the design and development of the teacher dashboard.
- A theoretical understanding of different facets involved in designing actionable insight for teachers, along with a novel framework for co-constructing actionable insight needed to support students as they implement technology-enhanced, PBL,
- The use of a novel classroom simulation approach to support and prepare teachers in the implementation of problem-based, STEM+C learning in science.

While this proposed dissertation project primarily focuses on a middle school science classroom using a problem-based computational modeling and engineering design approach, the resulting development strategy and products have broad application across disciplinary domains, instructional contexts, and teacher and student populations.

1.4 Organization of this Dissertation

This dissertation is organized as follows. Chapter 2 provides the **literature review** covering three related areas of (1) technology-supported teacher noticing and response (Section 2.1), (2) integrated STEM and CT learning in K-12 science classrooms (Section 2.2), and (3) learning analytics and pedagogy (Section 2.3). The previous work conducted in these areas motivates the research presented and the contributions made in this dissertation.

Chapters 3, 4, 5, consist of the **three manuscripts** resulting from the work completed during this dissertation. Background from Chapter 2 may be repeated in relevant background sections of each manuscript.

1.4.1 Manuscript One Summary

Title: *Temporal Evolution of Student Learning and Problem-Solving Behaviors During an NGSS-aligned Integrated Science, Computing, and Engineering Curriculum.*

Abstract: Computational modeling offers opportunities for students to explore and develop complex science and engineering concepts that may be difficult to replicate in traditional K-12 classroom environments. However, limited research has studied how students construct their knowledge and develop their problem-solving processes when working with curricula that integrate science and engineering concepts and practices supported by a Computational Thinking (CT) framework. We hypothesize that computation can serve as a bridge, leverage the connections between science and engineering and promote synergistic knowledge construction and learning across domains. This paper examines the knowledge construction processes and associated learning behaviors and strategies that students employed during a three-week, NGSS-aligned integrated middle school curriculum that introduced students to earth science concepts of absorption and runoff, and scientific modeling practices using multiple linked representations. The students then used their constructed computational models to solve a design problem (designing a school yard that minimizes runoff after heavy rainfall while meeting cost and accessibility constraints). Formative assessments are interspersed in the curriculum to support student learning. We apply correlation and Path Analysis to evaluate students' learning trajectories across science, engineering, and computation, and conduct exploratory cluster analysis to explore the interactions between domain-specific learning and problem-solving. We demonstrate the impact of our novel, technology-enhanced curriculum that supports students' progress in synergistic learning of science, computation, and engineering as they progress through the different curricular units.

Status: To be submitted to the Journal of the Learning Sciences.

1.4.2 Manuscript Two Summary

Title: *Co-Designing a Teacher Dashboard to Support Evidence-Based Instruction During Problem-Based Learning in Middle School Science*

Abstract: Keeping the teacher engaged during students' learning and problem solving in technology-enhanced, integrated problem-based learning (PBL) has been shown to support deeper student involvement, and, therefore, better success learning difficult science, computing, and engineering concepts and practices. However, there are identification and scaling challenges in understanding the impact of PBL on students' learning processes, as these processes are captured through mouse clicks, drag and drop actions, and other low-level activities in the computer-based environments. Therefore, students' learning processes and corresponding difficulties are not easily noticed by teachers as students learn from these environments. As a result, teachers find it difficult to set up meaningful interactions with students while also maintaining the focus on student-centered learning. In this paper, we investigate how the creation of classroom instructional-support technology can provide insights to teachers that are actionable and meaningful in PBL classroom contexts. However, open-ended and exploratory approaches that form the basis of PBL, and the accompanying complexities of the technologies we develop to support PBL, make it difficult for teachers to meaningfully contribute to the design and development processes that would be needed to generate the support tools that would help them interpret student learning and generate meaningful support to enhance learning.

This article presents a detailed case study on a multi-step approach to the co-design and development of a teacher dashboard to support and prepare teachers for implementing a technology-enhanced, PBL curriculum in their middle school science classrooms. This work presents a novel, end-to-end demonstration of how to engage teachers in the developing and interpreting of learning analytics and visualization systems that support tracking student performance and learning behaviors in a middle school PBL curriculum, and a co-design approach to developing the tools that support teacher noticing and the co-construction of actionable insight into students' learning and problem solving during PBL. Our approach adapts established participatory design techniques and demonstrates new kinds of prototyping methods to address the unique challenges of co-designing interpretable learning analytics for PBL curriculum applications. We leverage conjecture mapping for the design of our teacher dashboard. Our contributions include the teacher dashboard and descriptions of the co-designed features that teachers found useful for teaching PBL in middle school science, prototyping methods that leverage teachers-researcher partnerships in instructional-support technology design, and a reflection of how these approaches help inform technology refinements and innovation.

Status: Following feedback from my PhD thesis committee, I will submit Manuscript 2 to the Journal of Learning Analytics.

1.4.3 Manuscript Three Summary

Title: *Using Teacher Dashboards to Customize Lesson Plans for a Middle School STEM Curriculum*

Abstract: Prior research has demonstrated the importance of teacher engagement in students' developing ideas and strategies to support their STEM learning. In applications of student-centered learning approaches, such as problem-based learning (PBL), this engagement poses challenges as teachers must interpret and respond to student progress in ways that target learning and problem-solving needs while also maintaining the intent of the learning design (e.g., not always address a specific knowledge gap through direct instruction). Technology-enhanced approaches can mitigate these challenges by visualizing student learning and problem-solving behaviors to support teachers using orchestration technologies such as teacher dashboards. However, little research has targeted (1) dashboard-supported responsive teaching and (2) what features of the dashboard teachers find useful for teaching PBL, especially at the middle school level. This study examined 8 teachers' use of a co-designed teacher dashboard to assess and respond to students' learning and strategies during an integrated, PBL STEM curriculum. Teachers completed a series of 5 "planning period simulations" leveraging the dashboard and think-aloud protocols were implemented, supported by semi-structured interview questions, to enable the teachers to verbalize their thought and evaluation processes. Content analysis was conducted to analyze the transcripts. This study found that expert teachers made consistent links between the integrated domains, including leveraging cross-cutting concepts and linking science and computing practices, and leveraged collaborative pedagogical responses (e.g., pairing students with different problem-solving approaches) at earlier stages in the simulation process. All teachers focused on responses that demonstrated productive practices instead of solely targeting content knowledge deficiencies. Key dashboard features leveraged by teachers included (1) visualizations that grouped students and showed member transitions, (2) reflection tools that supported interpretation and negotiation of potential responses, and (3) highlighting student and class successes. Understanding how teachers use dashboards to support evidence-based teaching practices during technology-enhanced curricula is critical for improving teacher support and preparation.

Status: Following feedback from my PhD committee, I will submit Manuscript 3 to a journal that focuses on teaching and teacher education.

Finally, Chapter 7 discusses the contributions, limitations, and directions for future work to advance the present research.

CHAPTER 2

Literature Review

Engaging in students' developing ideas in a modeling and engineering design curriculum requires that a teacher notice students' developing scientific ideas and problem-solving processes during technology-enhanced learning and respond accordingly. To do so, a teacher must be able to match the necessary information about the learning objectives, including target concepts, practices, and potential misapplications, with student performance, system behaviors, and learning progressions. The goal of this research is to facilitate these teacher moves by providing pedagogically supportive learning analytics that are *interpretable* and *actionable*.

In this literature review, we cover three key areas of research: Responsive Teaching, Integrated STEM+CT Learning, and Learning Analytics. Figure 2.1 highlights the key successes and gaps in the literature that will be discussed throughout this chapter.

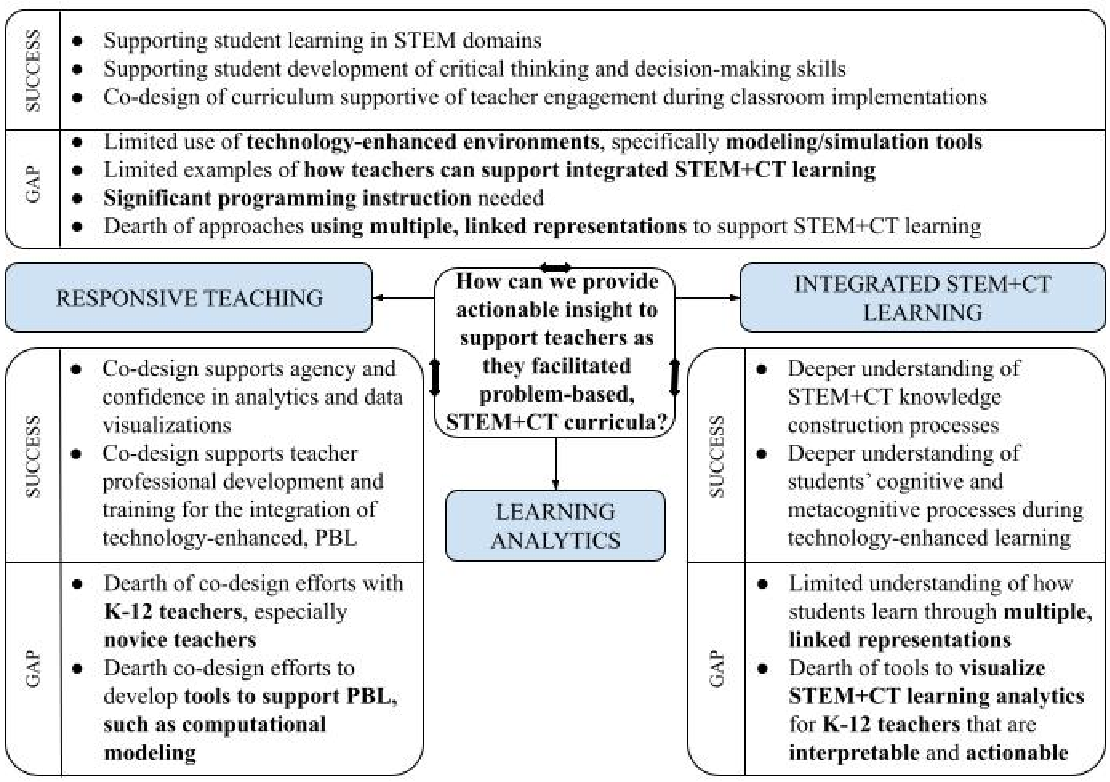


Figure 2.1: Overview of literature review chapter.

Based on this roadmap, we will first provide research on teacher noticing and support in STEM using technology-enhanced learning environments. This is followed by a review of the current state of computational modeling (learning-by-modeling) and engineering design (learning-by-design) in K-12 science education, including recent assessment and learning analytics techniques. We will then provide background on how learning analytics has progressed in supporting teachers in the classroom. The literature review will conclude with a discussion tying together the needs for teacher noticing, 21st century STEM learning, and learning analytics dashboards.

2.1 Supporting Teacher Noticing and Response in Technology-enhanced Classrooms

Science and math education reform has led to the promotion of fluid classroom environments that allow for pedagogical adjustments during instruction (van Es and Sherin, 2002). This pedagogical decision-making paradigm leverages responsive teaching in which the teacher makes in-the-moment pedagogical decisions based on what and how students are thinking, assessed through what students are saying or doing (Bywater et al., 2019; Wendell, 2016; Hammer et al., 2012).

This responsive approach is in contrast to traditional methods, in which lesson plans are predetermined and direct students' "flow of thought" (Hammer et al., 2012, p.54). This predetermined, traditional approach limits student opportunities to develop and assess their own ideas, which is needed for inquiry learning (Jiménez-Aleixandre et al., 2000) and open-ended learning approaches that include learning-by-modeling (Wilkerson-Jerde et al., 2015) and learning-by-design (Bywater et al., 2021; Watkins et al., 2018), such as that targeted in this proposed research.

Attending and responding to the disciplinary substance of student ideas is considered a core teaching practice in science, math, and engineering (NRC, 2007; Levin et al., 2009; Lampert et al., 2013; Coffey et al., 2011; Johnson et al., 2017). Responding to student ideas as they unfold in class has proven to:

1. help students engage in science practices (Schwarz et al., 2017),
2. focus student attention on the disciplinary substance of their thought (Robertson et al., 2016; Warren and Rosebery, 1995), and
3. improve students' conceptual understandings (e.g., Pierson, 2008; Empson and Jacobs, 2008).

This process is akin to formative feedback, providing students information to support adjustments in their thinking, guide them towards the desired learning goals, and improve knowledge development (Shute, 2008; Bransford et al., 2000).

However, Van Es and Sherin note that successful applications of responsive teaching requires teachers to develop new ways to engage in and interpret classroom interactions (2002). The complex, challenging practice of responding to student ideas requires that teachers consider and evaluate copious amounts of classroom information (e.g., student discourse, performance) as well as the intrinsic and extrinsic constraints of the classroom environment (e.g., learning standards and objectives, time, assessment needs), and make in-the-moment decisions on what and how to engage in their students' ideas (Bywater et al., 2019; Sherin, 2002; van Es and Sherin, 2002).

The complexity of this practice can be exacerbated during problem-based computational modeling and engineering design, as student actions are difficult to view and reason about. In our research, teachers have discussed that a main concern about such integration is that they are no longer able to view and evaluate student problem-solving actions as they can during a physical lab. These experiences motivate a deeper understanding of what it means to notice student thinking during technology-enhanced learning and how to systematically design systems and feedback to support teachers in this process.

In this section, we will discuss what it means to notice and interpret students' developing ideas as they unfold in class and considerations that should be made to support teacher noticing and response in technology-enhanced learning environments.

2.1.1 Teacher Noticing

The promotion of responsive teaching has invigorated a growing amount of literature to better understand teachers' noticing and responses to the disciplinary substance of student thinking (Walkoe et al., 2017; Sherin et al., 2011), and how to adequately support teachers in this process (Baker et al., 2020; Bywater et al., 2019).

Noticing, or identifying and interpreting the substance of students' developing ideas, is a critical, initial component of responsive teaching (Jacobs et al., 2010; Hammer et al., 2012). This includes "what teachers identify as important during classroom learning; what connections teachers make between specific classroom events and broader pedagogical ideas; and what contextual information can be used to reason about the specific classroom events" (Bywater et al., 2019, p. 17). Effective application of this process allows for teachers to decide how to best respond to their students' ideas (Jacobs et al., 2010). Moreover, the continuing attention to and evaluation of student thinking during class has been shown to positively impact student learning and the teacher's pedagogical decision making (Cowie and Bell, 1999; Hammer, 1997).

Van Es and Sherin discussed three key elements of teacher noticing:

1. identifying what is important about the classroom interaction (i.e., determining what information deserves further attention),

2. leveraging one's understanding of the context and classroom environment to reason about the interaction (i.e., using knowledge of the subject matter, the students, and the classroom context to evaluate the

evidence of the information presented), and

3. connecting specifics of the classroom interaction to broader teaching and learning ideas represented (i.e., deducing how a specific interaction is an example of a more general principle of teaching and learning) (2002).

Combined, these features require teachers to integrate multiple sources of information (e.g., subject-matter knowledge, classroom context, their understanding of how students think about the topic) in order to reason about the classroom interaction. For instance, in mathematics, it has been shown that a key factor in increasing a teacher's ability to notice developing mathematics ideas is the development of a teacher's own Mathematics Knowledge for Teaching (MKT; Ball et al., 2005) in order to (1) increase their understanding of the diversity in student approaches and (2) understand how to scaffold students towards the lesson goals (Ball et al., 2009).

Effective noticing can be challenging for teachers (Sherin, 2002). As mentioned previously, teachers need to grapple with multiple competing goals and constraints as well as information sources (Sherin, 2002). In addition, research in math education demonstrates that teachers struggle to attend to the diversity of students' developing ideas in the classroom (e.g., Sherin et al., 2011).

These difficulties have also been seen in science instruction (Barnhart and van Es, 2015) and more recently during applications of engineering design (Watkins et al., 2018). For instance, effective teacher noticing and guidance in learning-by-design environments requires that teachers understand and notice the different strategies employed by students (Crismond and Adams, 2012). However, in engineering design, students are likely to have unique solutions and their paths for reaching their solutions may differ significantly, creating challenges for teachers to effectively notice and respond (Wang et al., 2011; NRC, 2014).

Finally, professional development efforts have been implemented to promote and support effective noticing and response to student ideas in STEM. These methods include:

1. the implementation of video analysis sessions with teachers (Johnson and Mawyer, 2019; Hammer and van Zee, 2006; Sherin and Han, 2004),

2. applications of interpretive frameworks of students' ideas (Furtak and Heredia, 2014), and

3. the development of pedagogical tools and strategies (Windschitl et al., 2012)

Furthermore, collaborative teacher inquiry into students' work has allowed teachers to work together to notice and interpret student work and discuss pedagogical responses, supporting their skills in responsive teaching (Little et al., 2003). These approaches provide a baseline for preparing and training teachers for teacher noticing during a our curriculum.

2.1.2 Technology-Supported Teacher Noticing

Research on teaching with technology has focused primarily on teachers' use of technology, including their competency with and beliefs about how and when to use technology in their classrooms (e.g., Mishra and Koehler, 2006). Recent work has explored curriculum and assessment design and modifications with technology-enhanced learning environments (Kali et al., 2015). However, limited research has explored teacher noticing of students' developing ideas while students use technology and there has been limited impact of teacher noticing on the design and implementation of K-12 classroom technology (Walkoe et al., 2017).

Technology-enhanced learning environments provide new opportunities for students to engage in authentic science practices (e.g., NGSS, 2013) and generate explanations of their developing STEM knowledge (Bywater et al., 2019; Slotta and Linn, 2009). Students' interactions with the technology can afford a unique perspective into the progression of student knowledge and thinking as they engage with computing tools (Noss and Hoyles, 1996) and the type of thinking the students express using these tools are often those promoted by the recent state and national standards (Walkoe et al., 2017). For instance, we have leveraged student activity data during modeling to evaluate student debugging and data analysis strategies during computational modeling (Hutchins et al., 2019a; Emara et al., 2020). This will be discussed in more detail in the Section 2.2.2.

Although these environments have the potential to support responsive teaching (Bywater et al., 2019), teacher noticing difficulties may be increased when students use problem-based compuational modeling environments due to (1) teachers' limited background in computing, programming, and teaching using technology (Bocconi et al., 2016), (2) the decreased visibility of student thinking, as it is now applied through mouse clicks and other user-interface interactions and, therefore, not easily or readily apparent to the teacher (an important feature of lesson design to support teacher noticing; e.g., National Council of Teachers of Mathematics, 2014), and (3) software constraints or user-interface difficulties that may impact teachers' abilities to adequately respond to student thinking or issues (Walkoe et al., 2017).

As an example, a teacher in one of our studies expressed the desire for the system to provide students feedback on potential user-interface difficulties, while letting the teacher focus her attention on student processes and potential domain-specific knowledge misunderstandings. For instance during physics computational modeling, students may inadvertently choose a "set x position" programming block instead of a "change x position" block because the blocks are close to each other in the list of available blocks and the student does not understand the difference between setting (i.e., initializing) a variable value versus changing (i.e., updating) the variable value. If the student is under the impression they selected the correct block and debugging processes do not work, it may be time consuming for a teacher to read through the students' code line-by-line to check.

Finally, while these environments support key processes highlighted in state and national standards, these strategies are often not engaged in by teachers during instruction (Walkoe et al., 2017). As such, more research and development is needed to leverage action data to provide teachers the necessary feedback to employ the benefits of effective teacher noticing and response.

In this proposed research, we are interested in increasing our understanding (1) how to design problem-based computational modeling and engineering design learning environments in a manner that supports responsive teaching and (2) how teachers can then use learning analytics generated by analyzing student work in the environment to notice and respond to the disciplinary substance of student thinking and problem-solving processes during open-ended learning, specifically during learning-by-modeling and learning-by-design.

These open-ended tasks provide a unique opportunity to examine the effectiveness of learning analytics to provide teachers with information on student idea development and problem-solving processes in a manner that is *interpretable* (teachers are able to identify the important disciplinary substance in student action sequences) and *actionable* (teachers have opportunities to respond or provide formative feedback to students to guide them towards the learning goals). In the next section, I will discuss these curricular approaches in more detail.

2.2 K-12 STEM Classrooms

The Framework for K-12 Science Education and the Next Generation Science Standards (NGSS) calls for the integration of science and engineering in K-12 classrooms have highlighted a need to provide authentic learning experiences that better prepare students to succeed in the 21st century. The open-ended nature of these activities, such as computational modeling and engineering design tasks, provides a unique opportunity for students to explore and represent their developing scientific ideas. In these open-ended learning environments, students are given specific problem-solving tasks, but they are free to choose their approach to learning and problem solving. In this section, we will discuss the importance of integrating computation and engineering into K-12 science classrooms. We will provide background on learning-by-modeling and learning-by-design in K-12 science classrooms, the benefits and difficulties for students and teachers, and progress on assessments and analytics that are supportive of feedback on student knowledge construction, ideas, and problem-solving processes.

2.2.1 Integrating Science, Computing, and Engineering

Wing (2006) spurred researchers, educators, and policymakers to introduce computational thinking (CT) and computer science (CS) as "a universally applicable attitude and skill set" oriented towards designing and finding solutions to problems using computational mechanisms (Wing (2006), p. 33). These skills include (among others) logical and algorithmic thinking, abstraction, problem decomposition, pattern recognition and generalization, and debugging (systematic error detection and resolution) (Hutchins et al., 2020a). The 2012 Science Framework (NRC, 2012) also acknowledged the multiple connections among STEM and CT domains— "more and more frequently, scientists work in interdisciplinary teams that blur traditional boundaries" (p.31)—, and "consider connections among science, technology, engineering, and mathematics" (p.32). For example, the Science Framework identifies Using Mathematics and Computational Thinking as one of eight scientific and engineering practices that K-12 students should learn. Mandates for an education that prepares learners for life and work—and specifically STEM and CT work—for the 21st century, as well as progressive standards for STEM subjects reflect this integrated STEM perspective (Grover and Pea, 2018; NGSS, 2013). The leveraging of these key STEM and CT integration benefits has been actualized through the use of learning-by-modeling and learning-by-design pedagogical approaches (e.g., Hambrusch et al., 2009; Weintrop et al., 2016; Hutchins et al., 2020a; Chiu et al., 2019).

With respect to learning-by-modeling, integrating CT and scientific modeling can be synergistic (Hutchins et al., 2020a; Snyder et al., 2019b), i.e., supportive of each other along multiple dimensions, by:

1. lowering the learning threshold for science concepts by reorganizing them around intuitive computational representations that introduce discrete and qualitative forms of fundamental laws, which are simpler to understand than equation-based continuous forms (Redish and Wilson, 1993; Sherin, 2001);
2. studying a phenomenon as a discrete time process, where behavior advances in a step-by-step fashion is easier for students to comprehend when compared to continuous dynamics (diSessa, 2001; Hutchins et al., 2020a; Sherin, 2001);
3. representing programming and computational modeling as core scientific practices, such as modeling, verification, and explanation (Soloway, 1993);
4. contextualizing computational constructs in order to make it easier to learn programming (Papert, 1991).

For example, in terms of computational representations of science phenomena, having to specify how to split things up (e.g., separate upward and downward motion for a projectile fired upward, and make decisions, such as, when the projectile hits the ground traveling downward, should it stop, and its velocity be set to 0, instead of continuing to allow it to decrease?) helps make assumptions more explicit, and student conceptions more visible.

Finally, visualizations afforded by simulating computational models (e.g., animations and graphs) make it easier for learners to judge legitimacy (Sherin, 2001). These benefits reflect the framing of proficiency in both science and CT (by the NGSS and K-12 CS Framework, respectively) as the integration of knowledge and practice.

Learning-by-design curricula have also emphasized preparing students for the 21st century workforce, engaging students in scientific investigation and engineering design activities that improve their knowledge, reasoning, and problem-solving skills (NRC, 2000; NAE and NASEM, 2019). In the K-12 setting, students can engage in integrated science and engineering activities in meaningful ways, that include question posing, design testing, and solution generation (Hirsch et al., 2007). Curricular implementations of engineering design have proven to support students conceptual science and engineering learning (e.g., McElhaney et al., 2020; Mehalik et al., 2008).

Moreover, recent reports also indicate that the learning of engineering and CT concepts and practices can be synergistic, empowering learning in each domain (Ehsan et al., 2020). Engineering and CT can compliment each other in problem solving and system design, and the conceptual underpinnings of both engineering and CT may make engineering a productive discipline for extending CT learning and applications (e.g., Zhang, 2020; Wing, 2006; Shute et al., 2017).

In the following two sections, we will describe advances in learning-by-modeling and learning-by-design curricula integral for this proposed research.

2.2.2 Learning-by-Modeling

The learning-by-modeling framework, illustrated in Figure 2.2 (adapted from the Common Core Mathematics Standards (CCSSO, 2011)), highlights the role that different sub-processes may play in acquiring, interpreting, and refining one's knowledge when performing modeling tasks. The sub-processes illustrated match the NGSS on "Developing and Using Models" (NGSS, 2013) and define the key processes that a comprehensive learning-by-modeling environment must support. For the purpose of this research, we target the development of simulation models in STEM domains. In particular, our work focuses on developing comprehensive, simulation models, that capture the emergent behavior of relevant scientific phenomena expressed using computational constructs, and then using of those models to create engineering designs.

Simulation models, and specifically agent-based modeling, have received significant attention as a means of supporting STEM learning (Weintrop et al., 2016; Hutchins et al., 2020a). Simulation models adopt a multi-representational approach, where students develop computational models using programming con-

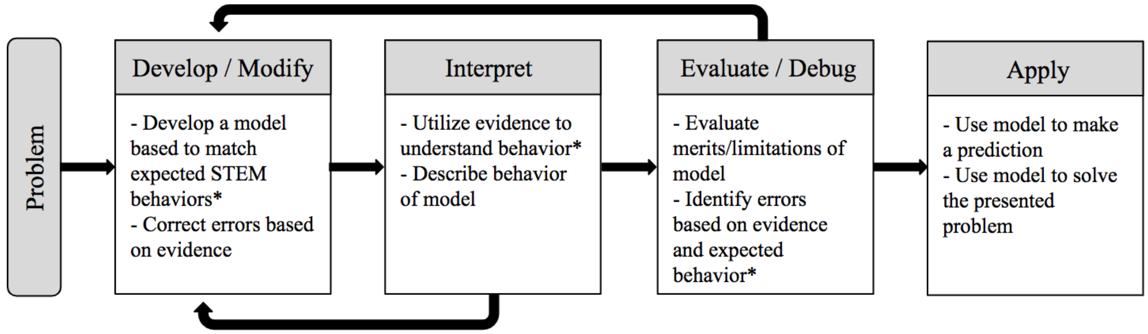


Figure 2.2: Processes and subprocesses integral for learning-by-modeling (Hutchins et al., 2020b).

structs, and visualize their behavior using animations, and other representational schemes, such as plots and charts. Agent-based modeling can be contrasted from constraint systems (e.g., Betty's Brain; Leelawong and Biswas, 2008), which use causal relations to model system behaviors, and system-dynamics models (e.g., Dragoon; VanLehn et al., 2017), which use simplifications of differential equations to represent dynamic system behaviors.

Environments such as CTSiM, ViMap, and CT-STEM (Basu et al., 2013; Sengupta et al., 2015; Jona et al., 2014) have supported research in this area, demonstrating the synergistic learning benefits resulting from this pedagogical approach. All of these environments extend NetLogo, a multi-agent programming language for building models that simulate the dynamic behaviors of complex, natural and social phenomena (Wilensky and Resnick, 1999). NetLogo provides an authoring environment for an agent modeling language that allows students to create their own models (or modify existing models that are available in a large, accompanying model library that comes with Netlogo). This enables learners to simulate and "play" with their models, exploring their behavior under various conditions. CTSiM (Basu et al., 2013), ViMap (Sengupta et al., 2015) and CT-STEM (Arastoopour Irgens et al., 2020; Swanson et al., 2019) provide a block-structured visual programming environment as an abstraction layer over NetLogo. This allows students to focus on the domain modeling tasks, without being overwhelmed by the syntax of the NetLogo programming language. Classroom studies conducted with these systems have produced successful results (e.g., Basu et al., 2016a; Weintrop et al., 2016; Hutchins et al., 2020a), supporting student learning in science and CT. Other work in learning-by-modeling includes Starlogo (Colella et al., 2001), and AgentSheets for scalable game design (Repenning et al., 2010).

2.2.2.1 Student and teacher difficulties with learning-by-modeling

As mentioned previously, the fast-paced nature of technological advancement has accelerated our need to better prepare students to utilize computational tools as vehicles for problem solving and professional advancement (Dede, 2010; Redish and Wilson, 2011; NRC, 2010). Whereas this conforms with the NGSS call for engaging students in authentic modeling practices in science, integrating computational modeling into the K-12 science curriculum has created difficulties in student learning. These difficulties include:

- Translating learnt domain knowledge to computational forms for model building (Sengupta et al., 2013; Basu et al., 2016b)
- Integrating key aspects of programming and CT (e.g., programming language syntax, identifying appropriate abstractions and developing iterative structures to model the dynamics of the scientific processes) (Grover and Basu, 2017; Hutchins et al., 2020a)
- Relating the behavior of individual entities to aggregated or emergent system behaviors (Chi, 2008; Wilensky and Resnick, 1999)
- Understanding the mathematical relations between variables and interpreting graphs in relation in the context of generated simulation behaviors (Sengupta and Farris, 2012; Araujo et al., 2008)
- Debugging the behaviors (results) generated by the abstract representations and interpreting them in terms of scientific principles and theories (Basu et al., 2016b).

These difficulties can be mapped on to the subprocess illustrated in Figure 2.2, and they need to be addressed in the context of the sub-processes in the figure to make the learning-by-modeling approach a productive experience for novice learners.

Additional concerns about computational modeling in science arise from a teaching and classroom perspective. These difficulties center on (1) limited teacher background on integrating computational thinking and computing in STEM and (2) difficulties and confidence with computational tools needed for the integration.

From the teacher background perspective, many teachers lack the experience and education with CT and computing needed (Yadav et al., 2016; Cuny, 2012; Peel et al., 2020). Research targeting STEM and CT integration have found that teachers hold misunderstandings about CT (Sands et al., 2018) and integration is impacted by low self-confidence and self-efficacy for teaching CT (Wu et al., 2018). Sands et al. (2018) recommend leveraging teachers' backgrounds by explicitly highlighting how CT helps learn disciplinary content.

From the system perspective, difficulties include the development of a shared understanding of the modeling language used to support the construction of the science models, and understanding the model behaviors when the model is executed (VanLehn, 2013). We hypothesize that this may be exacerbated when students create their own modeling structures (e.g., creation of custom blocks in Scratch or Snap!), especially if students have STEM and CT domain misunderstandings prior to their model development activities (Sengupta et al., 2013).

Finally, there are increased training requirements for teachers and students (e.g. class time spent on learning the features of the modeling environment) to establish a sufficient understanding of the computational constructs needed to build meaningful science models, and communicate results. This is especially true when text-based programming languages are used (e.g., Hashem and Mioduser, 2011; Sherin et al., 1993). Recently, researchers supporting environments such as CTSTEM have leveraged curriculum co-design approaches with participating teachers to tackle this issue (Wu et al., 2020), serving as motivation for this dissertation approach.

2.2.2.2 Learning-by-Modeling Assessments

Assessment is one of the most salient drivers of education at scale (Cuban, 1984). Without meticulous attention to assessments the dissemination of learning-by-modeling applications in K-12 schools has little hope (i.e., Grover et al., 2014; Basu et al., 2021). This is due to (1) the need to better understand how students learn in a curriculum that integrates science and CT (i.e., to improve pedagogical content knowledge, discussed in Section 2.1.1) and (2) the role assessments play in supporting improved curricular efforts in the integration of CT in STEM (e.g., modifying curriculum tools and scaffolds to better target identified misunderstandings of all learners).

In order to teach learning-by-modeling and CT effectively, teachers must have a solid understanding of what students know and can apply, as well as how their students develop CT skills over time (Bienkowski et al., 2015). This requirement is exacerbated when CT is introduced in other domains, as the educator must be able to differentiate between learning gains in each domain and to identify potential misunderstandings in domain concepts or practices in either domain to support classroom discussions and student feedback.

Given the recent impetus for increasing CS and CT educational opportunities through CSForAll, significant efforts have focused on assessment development and applications. Research on integrated STEM and CT assessments have mainly applied three formats: project-based, interview-based, and multiple-choice and short-response assessments.

Project-Based Assessments. Open-ended, free choice final projects are often used to assess student learning. This has been encouraged by the recent K-12 CS framework as a form of authentic assessment (Parker and DeLyser, 2017). While common in block-based programming environments, including Scratch, these computational artifacts can represent imperfect and incomplete measures of student learning, especially when used as the only form of assessment (Brennan and Resnick, 2012; Grover and Basu, 2017).

Project-based assessments, similar to those implemented in Alice (Werner et al., 2012), require subjective grading efforts that are often time consuming (Grover et al., 2014). The assessments typically focus on final code submission, eliminating potentially key information about student learning-by-modeling processes or the implementation of CT practices over time. In addition, from the standpoint of a classroom teacher, the total amount of potential CT information collected for each student may not be sufficient for the time it takes to grade each individual assignment. And finally, the dependency on a particular software can be problematic, as Webb noted that these options often require that students have familiarity with the software in use (2010). As such, these approaches are often not generalizable across software or domains.

Interview-Based, Questionnaires, and Surveys. The use of qualitative, interview-based strategies for assessing CT skills and understandings is common. These approaches (e.g., Brennan and Resnick, 2012; Werner et al., 2012) have proven beneficial in:

1. identifying potential gaps in CT understanding not captured through frequency analysis of project data,
2. improving understanding of students' CT processes through evaluations of students' ability to understand and explain someone else's code, and
3. allowing for the capture of data regarding the transfer of knowledge or metacognitive abilities required for learning with understanding.

For instance, in the Brennan and Resnick paper (2012), students were asked to describe how portions of a particular program they developed work. This allowed researchers to evaluate students' ability to teach an outside party about how the code works. In some cases, this approach also allowed researchers to identify conceptual misunderstandings (for instance, when students were describing a code replicated from a publicly available project).

However, the applicability of this methodology in a classroom study is limited. This is primarily because the application of this assessment is time consuming. In a classroom setting, the need to meet individually with students to assess understanding is difficult because of the lack of time and resources. Moreover, evaluation of interview data is often subjective (Grover et al., 2014), limiting the generalizability of this form of assessment in K-12 classrooms. Further investigation on how to target these conceptual gaps and methods for capturing metacognitive and transfer skills is needed.

Multiple Choice and Short Answer Assessments A recent trend in CT assessments is the development of objective, multiple-choice and short-answer assessments targeting defined learning objectives. The applicability of this form of CT assessment is promising as it has proven to be a tool for building "a cumulative knowledge base of learning science for CT" (Grover et al., 2014, p.62). These assessments, often administered in a summative format, utilize questions that test student understandings of concepts and require students to apply relevant CT practices (e.g. debugging a pictured code segment in order to complete a required task).

However, as a standalone form of assessment, these assessments may not evaluate the CT skills and understandings targeted for a particular curriculum in a comprehensive manner. In particular, these forms of assessment often to not make the student thinking involved in the response process visible to teachers, limiting teachers' understandings of how an answer was derived, a key design feature for teacher noticing (see Section 2.1.1).

Systems of Assessment One approach for targeting the limitations of individual assessment types is the application of a "systems of assessment" approach (Conley and Darling-Hammond, 2013). This technique targets a deeper understanding of student knowledge construction by systematically combining multiple forms of assessments, including summative and formative assessments, and has proven to be effective in evaluating learning gains and improving our understanding of how CT is learned over time (Grover et al., 2015).

In addition, for the purpose of providing teachers with interpretable and actionable feedback during an integrated science, CT, and engineering curriculum, it will be important to evaluate multiple applications of student knowledge in all domains (Hutchins et al., 2021a). As such, a more comprehensive approach with multiple forms of assessment is needed to target deeper learning in STEM and CT.

2.2.2.3 Learning-by-Modeling Analytics Measures

Advancements in learning environments and learning analytics have supported the development of techniques to evaluate the processes or strategies students apply during modeling tasks (Zhang et al., 2021). Early efforts in the analysis of student log data from these systems focused on exploratory approaches to identify patterns of block usage in constructing models (Winne and Baker, 2013). For instance, Brennan and Resnick's use of log data consisted of frequency analysis (Brennan and Resnick, 2012), a common rubric methodology (Grover et al., 2018a). However, this approach was shown to not provide enough information regarding student conceptual understanding or process abilities (Brennan and Resnick, 2012).

Research in the development of adaptive student feedback during computer science education has led to advancements in our understanding, identification, and evaluation of problem-solving processes and knowledge representation during programming. For instance, Piech et al. utilized Code.org projects (an environment built using Blockly) to design partial solution feedback (2015), while Blikstein et al. used log data to identify program states and assess the likelihood of reaching a solution state or facing a "sink" state in which a student was likely to get stuck in (2014). Grover et al.'s hypothesis-driven learning analytics framework

(2017) introduced a blended hypothesis- and discovery-driven approach incorporating multiple data sources to interpret user actions related to computer science and CT learning. These approaches towards increasing our understanding of student processes and difficulties motivate the log-based learning analytics utilized in this proposed dissertation research.

Recently, significant work has been done targeting feedback development for students using the Snap! programming environment (the environment used in this proposed dissertation research) during their computer science coursework. This includes recent contributions from the HINTS Lab, led by Dr. Thomas Price. This work includes the development of a dynamic testing framework that enables teachers to design test cases for the automatic assessment of student work (Wang et al., 2021) and the development of a code classification model to support the semi-automated discovery of problem-specific misconceptions students demonstrate while programming (Shi et al., 2021). However, it is important to emphasize that these approaches are specific to computer science contexts (e.g., the teachers building the test cases are well-versed in computer science constructs and practices) and the feedback generated targets students (as opposed to this proposed research where teachers will receive the feedback so they may support their students). Thus, effort needs to be made to (1) apply these analysis to represent both domain and computer science (specifically, CT) knowledge applications or misunderstandings and (2) support science teachers in tasks, such as the development of system tools (e.g., test cases) and understanding the representations of student work generated by the system.

There have been improvements in log-based learning analytics specific to STEM modeling. For instance, Basu, Biswas and Kinnebrew (2017) describe students' modeling progress by calculating the distance to an expert model at each model revision. Others have implemented clustering methods to evaluate students' learning based on action data (Segedy et al., 2015b,a; Zhang et al., 2017). This includes our use of clustering analysis methods to evaluate differences between high and low performing students during a C2STEM computational modeling unit in physics (Hutchins et al., 2019a. More recently, we have established a generalizable framework for the identification of productive and unproductive strategies in open-ended learning environments (Zhang et al., 2021) leveraging a coherence analysis and task model approach and comparing strategy applications of high and low performing groups using differential sequence mining (Kinnebrew et al., 2013) determined by their summative performance.

These approaches allow us to identify the difficulties students face, described in the Section 2.2.2.1 through trace action data as model construction occurs. This lessens the objectivity concerns present in interview and think-aloud assessment approaches. For instance, in Zhang et al. (2021), differences in debugging approaches were identified between high and low performing groups, indicating the importance of productive debugging processes on learning gains.

However, limited research has targeted context-preserving strategy analysis that deepens our understanding of students' domain-level deficiencies that impact their model-building and debugging processes. Furthermore, these efforts have not included teacher feedback for the design and development of learning analytics visualizations that help teachers identify and understand these strategies and act on the results (as discussed in Section 1.2). This dissertation research targets these limitations.

2.2.3 Learning-by-Design

The prominent position of engineering in the K-12 Science Education Framework, NGSS, and the proposed structure of science standards that distinguishes between practices, disciplinary core ideas, and crosscutting concepts has increased interest and research in the design and development of engineering curricula that can be integrated into existing K-12 science classrooms (NRC, 2014; NGSS, 2013).

The Framework describes (1) the disciplinary core ideas of engineering (e.g., defining and delimiting an engineering problem, developing possible solutions, optimizing the design solution) (2) the key connections between science and engineering, and (3) essential practices for K-12 students. Bolstered by the need to develop students' innovative thinking and creative problem-solving skills for a rapidly changing technological workforce (see Section 2.2.1), these efforts engage students in science and engineering practices such as innovating, investigating, evaluating and testing, and reasoning with designs (Cunningham and Kelly, 2017).

Fundamental to engineering is design (Ferris, 2012; Cunningham et al., 2007). Engineering design involves goal-directed problem solving (Archer 1965) that invokes cognitive processes such as (1) understanding and defining the problem, (2) learning new concepts necessary for solving problems, (3) generating possible solutions, (4) optimizing solutions through testing and refinement to accomplish problem-solving goals (Crismond and Adams, 2012; English and King, 2017; Mehalik et al., 2008; Lucas and Hanson, 2016; NRC, 2010). Designers must leverage declarative science and math knowledge to inform design prototypes as well as procedural knowledge to effectively solve the target problem (Bucciarelli, 2003). Therefore, engineering design requires not only applying scientific knowledge to solve problems (e.g., de Figueiredo, 2008), but also the systematic evaluation of criteria and constraints to deliver a solution based on the social dimension of the problem (Ferris, 2012; Cunningham and Kelly, 2017).

Extending this work, Cunningham and Kelly (2017) described four categories of engineering actions, including "engineering in social contexts, uses of data and evidence to make decisions, tools and strategies for problem-solving, and finding solutions through creativity and innovation" (p. 491) with associated practices in each category (e.g., consider problems in context, envision multiple solutions, make evidence-based decisions, assess implications of solutions, work well in teams). These efforts help provide a clearer view for students and teachers of how to leverage key practices in solving engineering design problems (Mangiante and Gabriele-Black, 2020).

Leveraging this background, Figure 2.3 presents our learning-by-design framework. Similar to the learning-by-modeling framework, this figure highlights the role that different sub-processes play in the creation, testing, and reasoning with engineering designs as well as what a learning-by-design learning environment needs to support. Although engineering and engineering design have not traditionally been part of the K-12 curriculum (Cunningham et al., 2007), research efforts have targeted this dissemination.

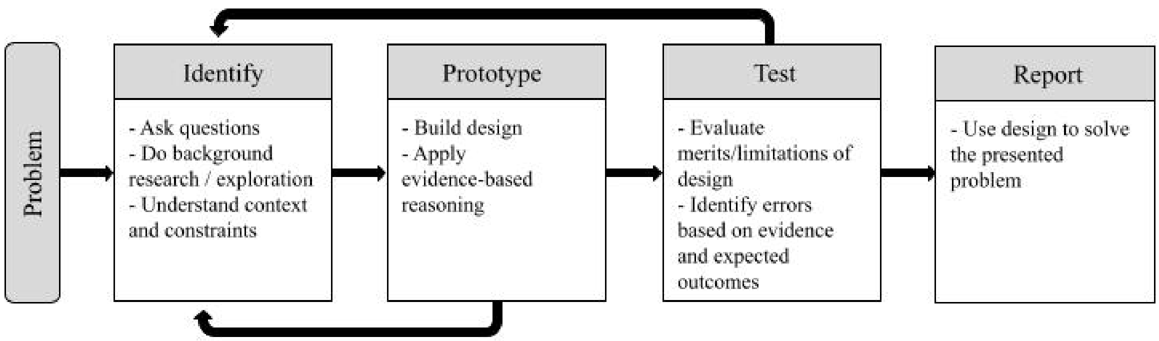


Figure 2.3: Processes and subprocesses integral for learning-by-design.

The Learning by Design (LbD) curriculum involved weeks-long design challenges and explored students' scientific reasoning (Kolodner et al., 1998, 2003). LbD emphasized the need to support learners' development of critical thinking and decision-making skills in the modern world (Kolodner et al., 2003). Wendell and Rogers (2013) developed the Science Through LEGO Engineering curriculum that was based on the LbD framework. In this curriculum, elementary students' designed and tested a musical instrument, a model house, a people mover, and an animal model.

Similar to these efforts, research evaluating the Engineering Is Elementary program also evaluated scientific knowledge development during an engineering curriculum (e.g., Lachapelle et al., 2015). In addition, earlier studies by Schauble et al. (e.g., 1991) evaluated how different modalities of thinking during experimentation influenced middle school students' learning of physics.

In recent work, Ehsan, Rehmat, and Cardella targeted an evaluation of the synergies between CT and engineering by investigating childrens' CT applications during an engineering design activity (2020). Through the systematic, qualitative analysis of students' processes (video recorded), the researchers aligned key engineering design actions with relevant CT competencies. For instance, problem scoping (e.g., understanding the problem's constraints and criteria) was linked to the CT practices of decomposition and abstraction) while design evaluation (e.g., evaluating prototype created) was linked to troubleshooting and debugging in CT. These linkages will serve as motivation for the selection of problem-solving strategies targeted for feedback to teachers.

Other approaches have targeted scientific sense-making during design. These include Roth's (1997) evaluation of elementary school students' reasoning when creating lifting machines and Penner et al.'s (1997) work evaluating students' reasoning about force and motion during a task called Designing Human Elbows, a biomechanics design project. Additional work includes the evaluation of students' conceptual knowledge about engineering (e.g., Streveler et al., 2008) and causal relationships in simple mechanical devices (e.g., Lehrer and Schauble, 1998; Bolger et al., 2012).

2.2.3.1 Student and teacher difficulties with learning-by-design

Learning-by-design is a complex task that involves simultaneous consideration of the design problem (including criteria and constraints) and potential solutions (de Figueiredo, 2008). The intertwining of multiple practices as they develop their science, math, and engineering conceptual knowledge can pose challenges for students. These include:

- Unsystematic prototyping, such as the use of trial and error patterns as opposed to more systematic design strategies (e.g., Ahmed et al., 2003; McElhaney et al., 2020)
- Applying science knowledge to make evidence-based design changes (McElhaney et al., 2020)
- Understanding cause and effect amongst design components or applying mechanistic reasoning to describe design components (Bolger et al., 2012)

These difficulties can be mapped on to the sub-process illustrated in Figure 2.3, and they need to be addressed in the context of the sub-processes in the figure to make the learning-by-design approach a productive experience for novice learners.

We hypothesize that addressing similar difficulties from the learning-by-modeling figure (e.g., translating scientific knowledge into computational form) may support the addressing of some of these difficulties. For instance, supporting student processes in translating science conceptual knowledge (i.e., understanding of water runoff) to computational form (creating the conditions for when there is runoff based on absorption limits of surface materials and total rainfall) requires students to reason about the impact of variable changes on other variables. Students may need similar support to improve their understanding of cause and effects of making adjustments in their design prototypes.

Furthermore, the push to integrate engineering in K-12 science curricula places new demands on science teachers (NRC, 2010; NGSS, 2013). Engineering design tasks involve applications of both conceptual engineering design knowledge and epistemic practices that may be difficult for teachers to notice and respond to (Mangiante and Gabriele-Black, 2020). Not only does this integration mean adding new content to an already full curricula (NRC, 2010), few teachers have the background or experience in engineering design

(Yaşar et al., 2006; Watkins et al., 2018; Cunningham, 2008), and teachers face challenges including misconceptions about what engineers do and general fears about teaching engineering in K-12 (Cunningham, 2008; Yaşar et al., 2006). However, little work has targeted responsive teaching specific to engineering design (Watkins et al., 2018) and there is a lack of professional development opportunities in engineering education (Mangiante and Gabriele-Black, 2020).

Finally, the majority of studies reviewed are implemented as paper or physical lab challenges. In this research, we leverage a simulation environment and online tools to support students in making links between science concepts and engineering analysis, while also facilitating the generation and testing of design solutions. We hypothesize that training and understanding of the computational tools, as seen in the learning-by-modeling section (Section 2.2.2), will pose additional challenges for K-12 teachers.

2.2.3.2 Learning-by-Design Assessments

There is a dearth of research on formative assessments in engineering education research, especially at the K-12 level (Wendell, 2016). Instead, research in evaluating engineering design has focused primarily on the disciplinary targets of instruction and has evaluated students using summative assessments, project-based rubrics, and in-depth case studies .

Summative assessment approaches have targeted evaluations of students' science and engineering content knowledge and practices (Wendell and Rogers, 2013; Streveler et al., 2008; Lehrer and Schauble, 1998; Lachapelle et al., 2015), and critical thinking skills (e.g., Penner et al., 1997). Recently, work by Zhang et al. (2020) distinguished learning evaluations in science, CT, and engineering in order to evaluate the impact domain learning on the other integrated domains over time.

Several assessment approaches have leveraged think-aloud protocols to assess engineering design. For example, Atman et al. (2008) asked participants to verbalize their design process as they designed a playground. Similar to Brennan and Resnick (2012), researchers have also implemented evaluation approaches in which students critique others' design processes (e.g., Hsu et al., 2014).

Other methods for visualizing student thinking processes include student written responses to prompts (Hirsch et al., 2012), the development of concept maps (Sims-Knight et al., 2004), and applications of formative surveys to support the identification of design challenges students face over time (Purzer et al., 2011).

However, similar to those seen in Section 2.2.2.2, these manual approaches are time-consuming and labor intensive to grade, and, therefore not scalable or supportive of on-demand teacher feedback. Results from these approaches often indicate discrepancies between what students say and the processes applied (indentified through log data) (Atman et al., 2008). Finally, findings from these manual approaches have shown that students' ability to describe the design process does not imply they know how to apply this knowledge to solve the design problem (Vieira et al., 2016).

Given the importance of formative assessment in the dissemination and integration of domains such as computation and engineering in K-12 science classroom, efforts targeting formative assessment development are needed.

2.2.3.3 Learning-by-Design Analytics Measures

Recent work has leveraged log data to evaluate engineering testing strategies. This includes characterizing differences in students' design processes using Time Series Analysis (Xie et al., 2014), analytics methods to evaluate systematic and unsystematic experimentation processes (Vieira et al., 2016), and Bayesian Network Models to automatically evaluate students' engineering design performance (Xing et al., 2021). These approaches have demonstrated how student engagement as well as differences in design processes can be evaluated using log data.

Recently, Zhang et al. (2019) evaluated changes in elementary and middle school students' design solutions based on predefined criteria, evaluating the number of tests, the number of satisfying designs, the best score achieved, and the submitted score, and correlated the students' behaviors with their performance on an evidence-based, integrated science and engineering pre-post assessment. In addition, Bywater et al. (2021) developed a new sequence segmentation method known as the Differential Segmentation of Categorical Sequences (DiSCS) algorithm to identify meaningful periods of design activity as students implement engineering design tasks in open-ended learning environments.

To our knowledge, research that leverages log data and learning analytics to evaluate engineering design practices at the K-12 level is still in its infancy. In addition, and similar to the learning analytics presented in the learning-by-modeling section (Section 2.2.2.3), these approaches lack contextual information about design testing strategies that may be beneficial for teacher interpretation and response during class (described in Section 1.2). As such, this proposed research targets contributions in engineering design analytics and visualizations to address these issues.

In order to integrate learning-by-modeling and learning-by-design into K-12 STEM classrooms, careful consideration must be made in designing curriculum and systems that facilitate active learning of STEM concepts and practices while also providing multiple sources of student knowledge applications for a more comprehensive understanding of students' learning processes and needs. This may require:

- evidence-centered design, development and integration of curricula and associated products (tasks, assessments, environment tools, etc)
- establishing a tight coupling and shared semantics between the STEM modeling language and the CT

constructs needed to build the models, and

- Systematic, context-preserving data processing of system usage to capture dual disciplinary substance of student processes during learning-by-modeling.

In order to effectively engage teachers with students' developing scientific ideas and knowledge, it is not enough to provide performance feedback. Given the difficulties faced by teachers in integrating modeling in STEM classrooms, more work is needed (1) to increase teacher knowledge and understanding about how students learn through learning-by-modeling and learning-by-design and (2) to co-design learning analytics visualizations that inform teachers about student learning behaviors, successes and difficulties, in a manner that can be leveraged by the teacher for effective pedagogical adjustments. The following section provides background on learning analytics research targeting educator support.

2.3 Learning Analytics and Pedagogy

Learning analytics is a promising approach for supporting teachers' noticing of and response to students' developing scientific ideas and problem-solving processes as students use technology in the classroom (Wiley et al., 2020; Bywater et al., 2019). As such, developing learning analytics measures that support effective teacher noticing and response remains a research priority.

In this section, we describe the progress and limitations of three key research areas in learning analytics and teacher decision-making. We also present related teacher dashboards that target providing teachers with feedback on student thinking and problem-solving processes.

2.3.1 Predicting Performance Using Digital Trace Data

The advancement and proliferation of learning management systems and MOOCs has led to an increase in digital traces of student learning (Fischer et al., 2020). Coupled with advances in data science and artificial intelligence, initial research in learning analytics used these digital traces (e.g., clickstreams, time on tasks, types of actions, etc) to create student behavior profiles in these environments (Gašević et al., 2015).

In most cases, the initial target users for these learning analytics methods were school administrators as these tools were used to predict student success utilizing the software to make institutional decisions, rather than in-the-moment pedagogical support (Dawson et al., 2014; Means et al., 2011). One reason for this was data literacy concerns. School administrators needed less training than teachers on the provided analytics (Vatrapu et al., 2011; Means et al., 2011).

In order to target dashboard data literary concerns for teachers, researchers began developing teacher-focused dashboards. However, these efforts proved inefficient in supporting teachers as drivers of student learning because the data used was not reflective of the disciplinary substance of the learning process and student thinking (Verbert et al., 2013; Schwendimann et al., 2017). For instance, clickstream data does not provide the necessary content to understand student ideas or domain misunderstandings needed to explain an observed sequence of actions a student performs. It is this insight that teachers need to best support students in their conceptual understanding and class performance (Baker et al., 2020).

Simultaneous, learning analytics researchers were expanding efforts to leverage available data to dive deeper into student learning processes and knowledge development using technology-enhanced learning environments. For instance, Segedy et al. (2015c) applied coherence analysis to better understand student's self-regulated learning behaviors while building causal models of science phenomena. Others applied machine learning and data analysis techniques to better understand students' abilities and where students' may need further support (Gobert et al., 2013). However, these approaches lacked participatory efforts by teachers to better leverage these learning analytics for their in-the-moment classroom needs.

2.3.2 Co-designing for Orchestration and Scripted Analytics

Stemming from the lack of teacher-focused efforts is the advancement of learning analytics based on participatory design. This work mainly began from the increased use of intelligent tutoring systems and centered on the creation of teacher dashboards that included feedback requested by participating teachers.

To do so, teachers would provide researchers with their learning objectives and assessments goals and, in turn, the researchers would design, develop, and implement the relevant learning analytics feedback and visualizations needed to support each teachers' pedagogical decisions (Tissenbaum et al., 2012; Echeverria et al., 2018; Rodríguez-Triana et al., 2018). For instance, Matuk and Linn (2015) implemented participatory design with teachers in order to evaluate how teachers used student performance data to inform the adaptation of curriculum.

However, many of these studies co-designed with teachers that had substantial experience with the learning environment and teaching with technology (Echeverria et al., 2018; Holstein et al., 2017; Matuk and Linn, 2015). This may limit the generalizability of the approach and the learning analytics used for pedagogical decision making (especially for less experienced teachers), and often hides the fact that the learning analytics still leveraged clickstream and time data as the source of information (as discussed in Section 2.3.1).

An additional approach to leveraging learning analytics for in-the-moment classroom adjustments is scripted analytics. In scripted analytics, pedagogical scripts are provided to instructors that outline the path students need to follow to successfully meet the curriculum's learning goals and objectives (Fischer et al., 2013). Dashboards associated with this approach provide teachers feedback on student trajectories, and deviations that need correction (Tissenbaum and Slotta, 2012; Rodríguez-Triana et al., 2018).

We include efforts targeting the creation and implementation of orchestration graphs, which help teachers conduct sequences of classroom activities at key social levels (individual, group, and class) (Dillenbourg, 2015; Haklev et al., 2017).

Feedback generated for teachers using scripted analytics approaches are primarily aimed at getting students "back on track" (Haklev et al., 2017) and not providing feedback for teachers to support student ideation and problem solving, which is needed for open-ended learning such as learning-by-modeling and learning-by-design.

2.3.3 Learning Analytics to Support Curriculum Design

The push to leverage learning analytics to support curriculum design and adjustments is a significant inspiration for this proposed research. Specifically, this work is driven by key limitations in existing learning analytics research that include:

1. lack of adaptivity in teacher control and agency to support the individual needs, preferences, and values of teachers (Shibani et al., 2019),

2. insufficient grounding in learning science theory to support collection and analysis of consequential student and classroom data generated by the curriculum activities (Reimann, 2016),

3. mis-alignment with curriculum design theory to inform and support actionable pedagogical responses to student queries and difficulties (Mangaroska and Giannakos, 2018), and

4. inability to integrate the teacher's pedagogical interventions with students' performance and learning behavior data so as to wholistically inform the curriculum design and the pedagogical processes adopted by the teachers (Dyckhoff et al., 2012).

Efforts to target these concerns resulted in the creation of learning analytics for curriculum design frameworks. While initial frameworks were high-level and difficult to operationalize (Corrin et al., 2016), frameworks such as Learning Analytics Implementation Design (LAID) (Wise and Vytasek, 2017) and Orchestrating Learning Analytics (OrLA) (Prieto et al., 2019) have been developed to target these operational issues.

LAID ensures systematic LA design methods that incorporate the principles of coordination (including alignment with learning objectives and recognition of classroom constraints), comparison (of learning performance based on fixed standards and relative class performance, grounded by learning sciences theory), and customization (recognizing differences in values, preferences, and constraints of individual teachers).

OrLA emphasizes the need for inter-stakeholder communication during the design and development process. The framework offers support and guidance on collaborative discourse and decision making for the adoption and implementation of learning analytics in the classroom.

Finally, researchers have targeted bringing context to data interpretation, which has proven to be more meaningful to teachers (e.g., Bakharia et al., 2016; Hernández-Leo et al., 2019) and recent efforts in multimodal learning analytics (MMLA) have shown promise in leveraging multiple data sources to better inform teachers about key learning and critical thinking processes (Mangaroska et al., 2020).

Addressing the issues itemized above remains a research priority, especially at the K-12 level. For instance, current research on using learning analytics to support curriculum design work has predominantly been applied in university settings. Therefore, additional efforts are needed to operationalize these approaches at the K-12 level.

2.3.4 Teacher Dashboards: Visualizing Student Learning

Coinciding with the progress in learning analytics discussed thus far, teacher dashboard development has also increased. As discussed earlier, the purpose of the teacher dashboard is to evaluate student performance and example efforts include the research discussed in Section 2.3.2 (e.g., Matuk and Linn, 2015; Echeverria et al., 2018).

Additional efforts include the design and development of dashboards and ambient displays to visualize student progress and difficulties during activities (Slotta et al., 2013; Alavi and Dillenbourg, 2012) to help teachers determine where to direct their attention. However, as mentioned above, these efforts mainly aim to get students "back on track" (Haklev et al., 2017) and do not focus on the disciplinary substance of student learning and critical thinking processes.

For my research, I target teacher engagement with and in student thinking and problem-solving processes that can be amplified and made available through teacher feedback using learning analytics tools. Example approaches that target similar goals include The Teacher Responding Tool (Bywater et al., 2019), Lumilo (Holstein et al., 2019), and The Teacher Action Planner (Gerard et al., 2020).

The Teacher Responding Tool supports responsive teaching by scaffolding the response process, providing automated, student-specific recommendations on students mathematical ideas by leveraging a natural language processing tool. A control experiment demonstrated the benefits of the scaffolding tool to support teacher noticing of the disciplinary substance of student mathematical ideas and improving teacher responses as compared to the unscaffolded group. However, this work was not designed for in-the-moment feedback during class.

Lumilo extended the idea that intelligent tutoring systems (ITSs) could be more effective in helping students learn if they work together with human teachers (Xhakaj et al., 2017). Holstein et al. (2019) co-designed and implemented a real-time teacher awareness tool that provided teachers with feedback on student learning situations that were identified as better-suited for the teacher to handle. A 3-condition experiment demonstrated the effectiveness of Lumilo in helping narrow learning outcome gaps and improving student learning. As discussed above, though, this work did leverage experienced teachers and utilized digital trace and performance data that does not reflect students' developing ideas.

Finally, the Teacher Action Planner targets responsive teaching by providing teachers with evidence of student ideas during class. Similar to the Teacher Responding Tool, the Teacher Action Planner leverages natural language processing technology to evaluate students' written responses to web-based inquiry tasks and provides recommendations for instructional customizations. While this research included experienced teachers and focused solely on written responses, as opposed to model and design construction (targeted in my research), results demonstrate the promise of leveraging teacher dashboards to support evidence-based pedagogical modifications based on teachers' noticing of students developing scientific ideas.

2.4 Bringing it All Together: Motivation for My Research

Given the proliferation of technology-enhanced learning tools in today's classrooms as well as the need to best prepare our students for the 21st century workforce through computing education integrated with STEM, efforts to better support teachers in engaging in and responding to students developing disciplinary ideas and critical-thinking processes while using these tools are needed. As reviewed in this section, a number of limitations currently exist towards the actualization of this effort:

- Problem-based computational modeling and engineering design learning environments and accompanying curriculum are not designed in a manner that considers the needs, preferences, and concerns of teachers in noticing and responding to students' developing scientific ideas (Walkoe et al., 2017). This include methods to provide teachers' feedback and support that leverage and consider their background and experience limitations (e.g., lack of computing experience) (Sands et al., 2018; Walkoe et al., 2017),

- Dearth of comprehensive curriculum and assessment approaches for the integration of learning-by-modeling and learning-by-design in K-12 classrooms (Wendell, 2016; Hutchins et al., 2020a) as well as the time and labor constraints inherent in the manual evaluation of assessments, and

- Learning analytics and accompanying teacher dashboards have not typically targeted the evaluation and representation of students' developing scientific ideas and problem-solving strategies in a manner that can be leveraged by classroom teachers to support their instructional activities and support student learning (Baker et al., 2020; Wiley et al., 2020; Bywater et al., 2019).

When teachers engage with student ideas, students' knowledge and skills increase (Robertson et al., 2016). Technology-enhanced, problem-based learning environments, including learning-by-modeling and learning-by-design curricula, offer a unique opportunity for students to develop and implement their scientific and engineering ideas, which can be captured as students work with computational tools in the environments. Moreover, these environments promote students engagement in the practices and skills needed for success in our technology-enhanced workforce (Redish and Wilson, 2011; Grover et al., 2018b).

However, to our knowledge, no such tools exist that incorporate teacher needs, preferences and concerns regarding the integration of open-ended learning in their classroom through responsive teaching in the design of the curriculum and environment. In addition, no co-designed teacher dashboards exist that identify and provide teachers feedback on student thinking and problem-solving processes during open-ended learning, such as learning-by-modeling and learning-by-design. In the following Chapters, I will present the three manuscripts developed for this dissertation targeting these gaps.

CHAPTER 3

Manuscript One: Temporal Evolution of Student Learning and Problem Solving Behaviors During an NGSS-aligned Integrated Science, Computing, and Engineering Curriculum

3.1 Introduction

The growth and proliferation of computational technologies warrants increased application of technology-enhanced, learning and problem-solving opportunities in our K-12 science, technology, engineering, and mathematics (STEM) curricula. As scientists and engineers leverage computational processes and devices in their exploration, inquiry, modeling, and problem-solving tasks, it is important for us to introduce relevant computing methods in K-12 science and engineering curricula (Hambrusch et al., 2009; Weintrop et al., 2016; Hutchins et al., 2020a; Kolodner et al., 2003; Wendell, 2016; Schauble et al., 1991). Appropriate use of computing technologies should bolster students' learning and skills, and engage students in gaining a deeper understanding of STEM concepts by supporting inquiry and problem solving to them actively participate and excel in our technology-oriented workforce.

Problem-based learning and inquiry have been motivated and supported by the *Framework for K-12 Science Education* (NRC, 2012) and the Next Generation Science Standards (NGSS; NGSS, 2013), which articulate a vision for integrating science and engineering, and including computational thinking (CT) as a key science and engineering practice. This integrated science and engineering framework highlights the importance of providing "*a context in which students can test their own developing scientific knowledge and apply it to practice problems"* (NRC, 2012, p. 12, emphasis added). However, limited resources exist for curriculum development and assessment in such integrated curricula (e.g., NRC, 2014; McElhaney et al., 2020), and our understanding of how students learn and problem solve in such integrated curricula is still in its infancy (Zhang et al., 2021; Bywater et al., 2019).

Given this background, this paper examines a novel approach for integrating science and engineering curricula, leveraging computing as a bridge to develop an understanding of systems using science inquiry and exploration, and then applying this learned knowledge to solve engineering problems. In our approach, we leverage *learning-by-modeling* to develop a deep understanding of scientific concepts and practices (c.f., Sengupta et al., 2013; Wen et al., 2020), and then using the developed computational models to solve engineering design problems (c.f., Ehsan et al., 2020). In more detail, our approach is grounded in the idea that a sequence of connected representations from conceptual to computational modeling of target science phenomenon to application of the computational model for engineering design helps students develop their knowledge construction processes (c.f., Frederiksen et al., 1999). In addition, such approaches are particularly useful for studying phenomena, such as water runoff that are hard to study systematically in physical environments. Therefore, one of our primary goals in this paper is to study how this approach contributes to our understanding of how students learn in integrated science and engineering curricula. We will do this by studying students' learning and problem solving leveraging a sequence of linked assessments, model-building tasks, and an engineering design problem.

More specifically, our SPICE problem-based learning (PBL; Hmelo-Silver, 2004) curriculum challenges students to redesign their schoolyard to minimize water runoff after heavy rainfall, while adhering to accessibility and cost constraints. The curriculum is composed of a sequence of fifteen lessons L1-L15, and interspersed with formative assessments, F1-F6 as shown in Figure 5.1. The lessons take students through a sequence of multiple connected representations, that include (1) *conceptual modeling* to learn the basic science concepts and their relations to real-world phenomena; (2) *Translation into a more formal representation using rules*; (3) learning *computational thinking (CT) constructs and practices* using a set of unplugged activities; (4) *computational modeling* to develop a simulation model of the scientific phenomena; and (5) *engineering design* solutions that are developed from a computational model to solve a playground design challenge. To study student learning, we applied an evidence-centered design (ECD) approach to curriculum and assessment development, including pre-post tests and formative assessments that covered the science, CT and engineering domains.

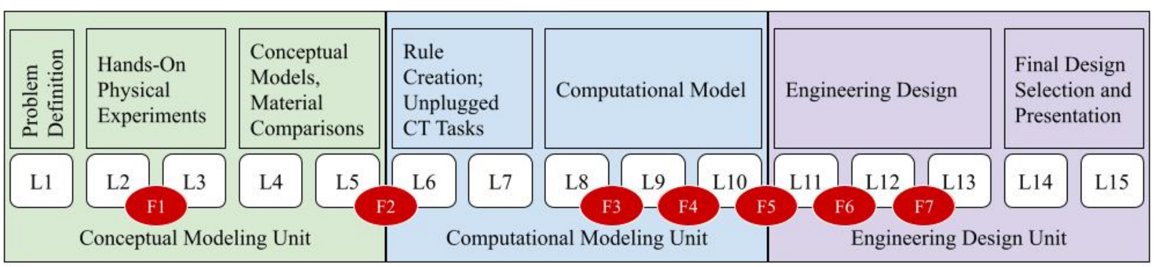


Figure 3.1: SPICE curricular sequence (L items are lessons and F items in red are formative assessments).

It is important to understand how and what students learn in these integrated learning environments before they can be disseminated to the larger community. In this paper, we report the results of a study conducted in a sixth grade science classroom in Southeastern United States. By analyzing students' learning performance and learning behaviors across the different Lesson units of curriculum, we answer the following research questions:

- (RQ1) *How effective is our integrated SPICE learning-by-modeling and problem-solving curriculum, with its sequencing of science, computing, and engineering design support student learning across the* *three disciplines?*

- (RQ2) *What is the role of computational thinking in facilitating learning of science and engineering concepts and practices?*
- (RQ3) *What kind of effective and ineffective strategies do students employ in their model-building tasks and how do these strategies correlate with their science, CT, and engineering learning?*

To answer RQ1 we provide statistical analysis of student results in our science, CT, and engineering assessments. For RQ2, we apply correlation analysis to evaluate the relationship between CT curriculum tasks on performance in science and engineering tasks and post-test performance and further this analysis by conducting Path Analysis to model those relations. Finally, to answer RQ3 we conduct an exploratory clustering analysis, grouping students based on common computational modeling and engineering design behaviors and evaluating the impact of those behavior combinations on students' learning in science, CT, and engineering.

This paper is organized as follows. Following a review of the literature on integrating science, CT, and engineering in K-12 classrooms and the identification of research gaps targeted by our research, we provide a detailed description of the SPICE curriculum and the technology-enhanced learning environment. This includes the design perspectives leveraged to create SPICE and an overview of the tasks, assessments, and technology. We then cover our methods for this paper, describing the study, participants, data collection processes, and the data analysis procedures implemented to answer our research questions. Our results and discussion section is organized by research question and is followed by conclusions, limitations, and future directions for this research.

3.2 Literature Review

With rapidly-changing technological advancements impacting our workplaces and every day lives, state and national standards have made prominent the need to integrate computing and engineering problem solving into K-12 science classrooms to better prepare our students for future success (NRC, 2014; NGSS, 2013). Engaging students in these integrated learning experiences promotes interdisciplinary critical thinking and skill development (Wing, 2006; Grover and Pea, 2013; Weintrop et al., 2016), while immersing students in open-ended, socially-relevant inquiry across STEM domains (NRC, 2012; Hutchins et al., 2020a). This paper leverages past approaches that have been successful in integrating science and computing (e.g., Weintrop et al., 2016; Hutchins et al., 2020a), science and engineering (e.g., Cunningham and Kelly, 2017; Kolodner et al., 2003), and, more recently, computing and engineering (e.g., Ehsan et al., 2020). In this section, we provide an overview of the successes and the gaps in the STEM integration literature, and summarize our findings in Figure 3.2.

3.2.1 Integrating Science and Computing

Past research has demonstrated the synergistic relation between computing and science along multiple dimensions (Grover and Pea, 2018), such as:

1. *Lowering the learning threshold for science concepts* by reorganizing them around intuitive computational representations that introduce fundamental laws using discrete and qualitative representations, which are simpler to understand than quantitative equation-based representations (Redish and Wilson, 1993; Sherin, 2001);

2. *Studying phenomena as discrete time processes*, where dynamic system behavior advances in a step-by-step fashion, is easier for students to comprehend as compared to differential equation representations of continuous dynamics (diSessa, 2001; Hutchins et al., 2020a; Sherin, 2001);

3. *Representing computational modeling as a core scientific practice*, that includes model construction, debugging, verification, and explanation (Soloway, 1993);

4. *Contextualizing Program construction* using domain-specific forms (e.g., Domain Specific Modeling Languages (DSMLs − (Hutchins et al., 2020b) to create synergistic relations between programming and building science models (Papert, 1991).

Moreover, visualizations afforded by simulating computational models in science (e.g., animations and graphs) make it easier for learners to interpret and verify the correctness of their models (Sherin, 2001).

Specific to the curricular approach in this research are agent-based, learning-by-modeling environments. Agent-based models capture the emergent behavior of relevant scientific phenomena expressed using computational constructs. These technology-enhanced environments and tools such as Netlogo and NetTango (Weintrop et al., 2016; Martin et al., 2020), COSCI (Chang et al., 2020), Glowscript VPython (Weller et al., 2021), CTSiM (Basu et al., 2013), ViMAP (Sengupta et al., 2015), and C2STEM (Hutchins et al., 2020a) leverage agent-based, computational models programmed by students to simulate the dynamic behaviors of complex, natural and social phenomena. Students are able to represent their developing scientific knowledge in computational form and then evaluate their code through visual assessments of animated agents' behaviors. This process leverages contextualized representations of computational constructs (e.g., conditional logic) to better expose students to programming (Papert, 1991). As such, these environments have received significant attention as a means for supporting STEM learning adopting a multi-representational approach.

Advances in learning environments and learning analytics have supported the development of techniques to evaluate the processes or strategies students apply during modeling tasks (Zhang et al., 2021). Early efforts in the analysis of student log data from these systems focused on exploratory approaches to identify patterns of block usage in constructing models (Winne and Baker, 2013). For instance, Brennan and Resnick's use of log data consisted of frequency analysis (Brennan and Resnick, 2012), a common rubric methodology (Grover et al., 2018a). However, this approach was shown to not provide enough information regarding student conceptual understanding or process abilities (Brennan and Resnick, 2012).

There have been improvements in log-based learning analytics specific to STEM modeling. For instance, Basu, Biswas and Kinnebrew (2017) describe students' modeling progress by calculating the distance to an expert model at each model revision. Others have implemented clustering methods to evaluate students' learning based on action data (Segedy et al., 2015b,a; Zhang et al., 2017). These approaches allow us to identify the difficulties students face through trace action data as model construction occurs. This lessens the objectivity concerns present in interview and think-aloud assessment approaches. For instance, in Zhang et al. (2021) and Grover et al. (2016), differences in debugging approaches were identified between high and low performing groups, indicating the importance of productive debugging processes on learning gains. However, to our knowledge, these approaches are still limited in number and no approach has extended this behavior evaluation's impact on problem-solving behaviors in other domains.

While the benefits detailed in prior research on science and CT reflect the framing of proficiency in both domains (by the NGSS and K-12 CS Framework, respectively) as the integration of knowledge and practice, limitations persist. These include (1) the need for extensive programming instruction which presents challenges for teachers and students with limited experience (Hashem and Mioduser, 2011), (2) dearth of comprehensive curriculum and assessment approaches that allow for temporal evaluation of domain-specific learning (Hutchins et al., 2020a), and (3) lack of analytics approaches to evaluate the impact of students' open-ended problem-solving behaviors on their domain-specific learning (Zhang et al., 2021).

3.2.2 Integrating Science and Engineering

The prominent position of engineering in the K-12 Science Education Framework, NGSS, and the proposed structure of science standards that distinguishes between practices, disciplinary core ideas, and crosscutting concepts has increased interest and research in the design and development of engineering curricula that can be integrated into existing K-12 science classrooms (NRC, 2014; NGSS, 2013). In the K-12 setting, students can engage in integrated science and engineering activities in meaningful ways, that include question posing, innovating, design testing, and solution generation (Hirsch et al., 2007; Cunningham and Kelly, 2017). Curricular implementations of engineering design have proven to support students conceptual science and engineering learning (e.g., McElhaney et al., 2020; Mehalik et al., 2008).

For example, the Learning by Design (LbD) curriculum involved weeks-long design challenges and explored students' scientific reasoning (Kolodner et al., 1998, 2003). LbD emphasized the need to support learners' development of critical thinking and decision-making skills in the modern world (Kolodner et al., 2003). Wendell and Rogers (2013) developed the Science Through LEGO Engineering curriculum that was based on the LbD framework. In this curriculum, elementary students' designed and tested a musical instrument, a model house, a people mover, and an animal model.

Similar to these efforts, research evaluating the Engineering Is Elementary program also evaluated scientific knowledge development during an engineering curriculum (e.g., Lachapelle et al., 2015). In addition, earlier studies by Schauble et al. (e.g., 1991) evaluated how different modalities of thinking during experimentation influenced middle school students' learning of physics.

However, the intertwining of multiple practices as students develop their science and engineering conceptual knowledge can pose challenges for students and gaps exist in terms of the integration of such curricula. For students, these challenges include:

- Unsystematic prototyping, such as the use of trial and error patterns as opposed to more systematic design strategies (e.g., Ahmed et al., 2003; McElhaney et al., 2020)
- Applying science knowledge to make evidence-based design changes (de Figueiredo, 2008; McElhaney et al., 2020)
- Understanding cause and effect amongst design components or applying mechanistic reasoning to describe design components (Bolger et al., 2012)

Similar to the integration of science and computing, there is a dearth of assessment approaches for evaluating student learning and problem-solving behaviors during science and engineering integration (Wendell, 2016). In addition, to our knowledge, research that leverages log data and learning analytics to evaluate engineering design practices at the K-12 level is still in its infancy with initial research characterizing differences in students' design processes (e.g., (Xie et al., 2014; Vieira et al., 2016; Bywater et al., 2021) and automatically evaluating students' engineering design performance (Xing et al., 2021). Finally, to our knowledge, current frameworks do not leverage modeling practices, in particular at the K-12 level, to support the connection between the two disciplines.

3.2.3 Integrating CT and Engineering

The final integration direction involves recent reports that indicate the synergistic nature of learning engineering and CT concepts and practices (e.g., Ehsan et al., 2020). These approaches currently include informal learning scenarios (Ehsan et al., 2020) and makerspaces (Yin et al., 2020). Engineering and CT can compliment each other in problem solving and system design, and the conceptual underpinnings of both engineering and CT may make engineering a productive discipline for extending CT learning and applications (e.g., Zhang, 2020; Wing, 2006; Shute et al., 2017) while also supporting the framing engineering as a thought process rather than "building."

Due to the limited research in this area, the integration of computing and engineering faces similar challenges as previously stated, including the death of assessment and analytics, the focus of implementations on older age groups (e.g., high school and university), and the inexperience of teachers and students in computing and engineering. Moreover, to our knowledge, limited work has targeted the use of computational modeling in the integration of computing and engineering. This approach may provide support in enabling students to design and develop engineering solutions that cannot be modeled physically (for instance, solutions to water runoff concerns).

3.2.4 Integrating Science, Computing, and Engineering Using Technology-Enhanced Environments

In addition to leveraging prior work in curriculum and assessment development for integrating STEM domains, this work is framed in the context of (1) problem-based learning (PBL) in open-ended learning environments (OELEs) and (2) learning through multiple linked representations.

OELEs provide students the opportunity to practice problem-solving skills in real-world contexts (Land, 2000a), and allow students to have a choice in how they pursue their learning and problem-solving tasks (Hannafin et al., 1999, 2014). Learners can leverage resources provided by the environment to acquire, understand, and apply knowledge needed to solve or complete a problem (Land, 2000a). In addition, OELEs provide students with tools or features to test and revise their evolving solutions. Example OELEs include inquiry environments (e.g., Ecolab (Luckin and du Boulay, 2016)), study tools (e.g., nStudy (Winne and Hadwin, 2013) and MetaTutor (Azevedo et al., 2010)), game-based environments (e.g., Crystal Island (Taub et al., 2019) and Tuglet (Käser and Schwartz, 2020)), constraint systems (e.g., Betty's Brain (Leelawong and Biswas, 2008) and DynaLearn (Bredeweg et al., 2013)) and, specific to our work, agent-based, learning-by-modeling environments (discussed above).

Finally, while technology-enhanced environments can scaffold students in computational modeling and engineering design processes that engage inquiry and problem solving (e.g., Jonassen et al., 2005; Keating et al., 2002), these evolving experiences need to be anchored to strong underlying conceptual models of the phenomena being investigated in order to leverage the unique affordances of each representation (Ainsworth, 2006). For instance, in this study, each of the evolving modeling representations makes explicit the conservation relationship among rainfall, absorbed water, and water runoff at different levels of abstraction and generality. Together, these representations provide a more complete depiction of phenomena and support students in deriving linkages between model representations (Frederiksen et al., 1999). Limited research describes how students make transitions and connections among model representations and identifies the instructional supports they require.

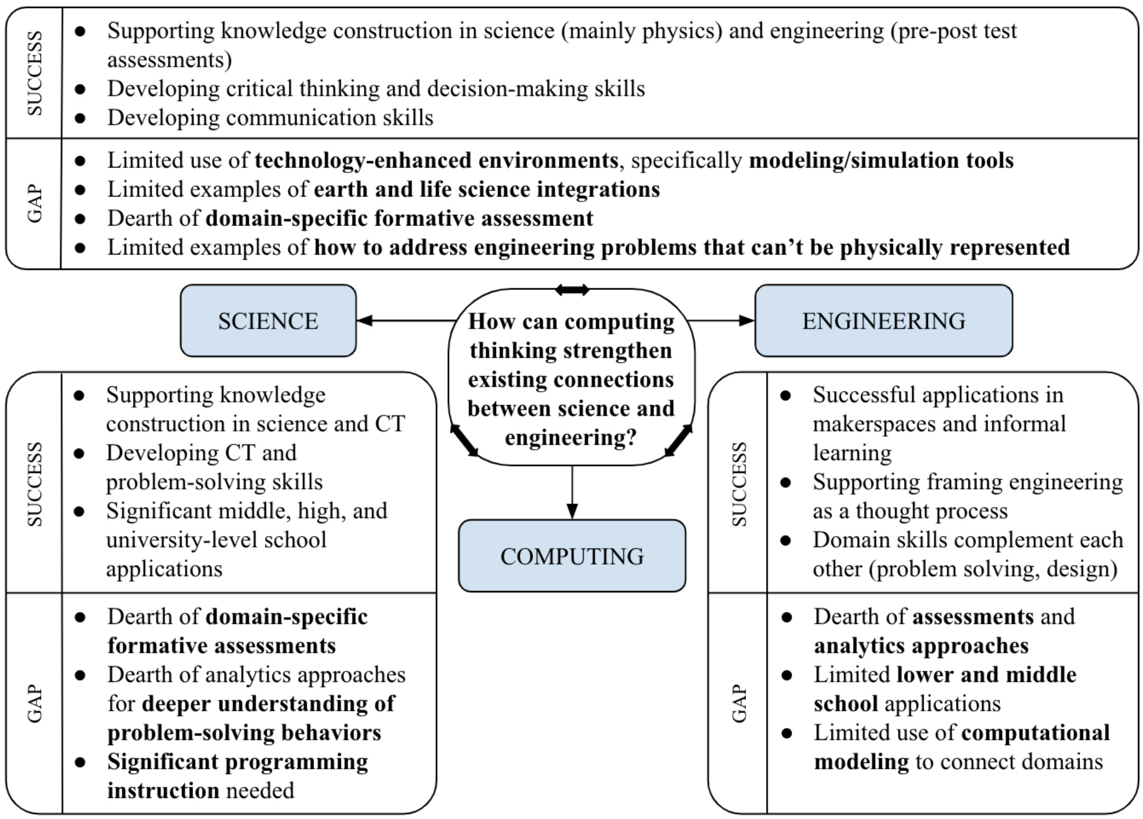


Figure 3.2: Understanding successes and gaps in the STEM integration literature.

3.2.5 Problem Statement

From the literature and illustrated in Figure 3.2, we ground our approach in key research findings, including the synergistic nature of the science, computing, and engineering domains when integrated and the affordances provided to students in terms of the development of critical-thinking and problem-solving skills (Hutchins et al., 2020a). In addition, we leverage opportunities discovered through applications of computational modeling in science in simulating complex science topics and supporting knowledge construction and problem-solving skill development in the coupled domains.

However, our review highlights key curriculum and assessment gaps that we target in this research. On one hand, although computational modeling has been suggested to be beneficial in science and computing integration, limited research has leveraged this approach to (1) support the addressing of engineering problems that can't be physically represented, (2) increase our understanding of problem-solving behaviors by leveraging trace data from OELEs, and (3) understand how computing can strengthen existing connections between science and engineering. Secondly, a dearth of assessment approaches in each combination of domains warrants further examination in order to more deeply understand how students construct their integrated knowledge over time. To do so, we systematically designed, developed, and implemented the SPICE curriculum and learning environment to target our research questions.

3.3 SPICE Curriculum and Learning Environment

SPICE supports teachers in the implementation of the SPICE Challenge (Chiu et al., 2019; McElhaney et al., 2019). The SPICE is a three-week, NGSS-aligned unit that challenges students to redesign their schoolyard using different surface materials to minimize the amount of water runoff after a storm, while adhering to a series of design constraints. These include the overall cost and accessibility, while providing for different functionalities for the schoolyard. The curriculum consists of three core units: physical experiments and conceptual modeling, computational modeling of the water runoff phenomenon, and engineering design, in which students use their computational models to redesign their schoolyard. This learning context is authentic and relevant to students facing similar problems (limited usability and pollution) in their own schools, therefore, the SPICE is potentially engaging and personally meaningful to the learners (Hutchins et al., 2021a; McElhaney et al., 2020).

SPICE targets NGSS performance expectations for upper elementary and middle school Earth science and engineering design curricula, emphasizing the movement of surface water in a system after heavy rainfall and the human impact of this runoff on the environment. In this section, we will detail the SPICE design process, curriculum, and assessments.

3.3.1 Design Perspectives

Our SPICE design processes to support the leveraging of computational modeling as a bridge to support science and engineering learning are guided by the following design perspectives.

- **Provide for learning from multiple, linked representations.**

  Technology-enhanced environments can scaffold students in complex learning processes that adopt exploration, inquiry, and systematic problem-solving processes (e.g., Jonassen et al., 2005; Sengupta et al., 2013; Weintrop et al., 2016). In order to support student learning-by-design, their design experiences need to be anchored to strong underlying scientific models of the phenomena being investigated in order to leverage the unique affordances of each representations. In designing the SPICE system, we have focused on multiple linked representations (Ainsworth, 2006; Basu et al., 2016a) that help the students progress through a conceptual understanding of the science phenomena (water runoff after a rainfall explained as a conservation principle), convert this understanding into computational models for computing the runoff in different materials after a rainfall, and then use the computational models to support the engineering design task. For example, a key mechanism used in this work to support link derivations are Domain Specific Modeling Languages (DSMLs) for computational model building (Hutchins et al., 2020b; Martin et al., 2020), which provide students programming blocks in the science domain to support the translation of science into computational form, and to use code developed to support model and design evaluations. Together, the representations provide a more complete depiction of phenomena and support students in deriving linkages between their science knowledge and their engineering design representations (Frederiksen et al., 1999).

- **Promote technology-enhanced problem solving.**

  The majority of studies reviewed in Section 3.2 are implemented as paper or physical lab challenges. In this research, we leverage a modeling and simulation environment that includes online design solution generation and design evaluation tools to help students understand the links between science concepts (e.g., absorption and runoff) and generating engineering solutions (e.g., what materials to choose for designing the playground) in a technology-enhanced learning environment (Kim and Hannafin, 2011). In addition, we provide tools for testing and evaluating design solutions (e.g., what are the costs and runoff for the current playground design solution). The environment, the tools, and their interfaces help students develop problem-solving skills, a key goal of the NGSS (NGSS, 2013).

- **Situate learning in open-ended, real-world problem contexts.**

  The NGSS and other science frameworks have highlighted the importance of providing authentic, real-world learning experiences that engage students in the skills needed to better prepare students for the needs of the 21st century workforce NGSS (2013). We leverage problem-based learning Hmelo-Silver (2004) in open-ended learning environments (OELEs) (Biswas et al., 2016; Hannafin et al., 1999). OELEs provide students the opportunity to practice problem-solving skills in real-world contexts (Land, 2000b), and allow students to have a choice in how they pursue their learning and problem-solving tasks (Hannafin et al., 2014). Learners can leverage resources provided by the environment to acquire, understand, and apply knowledge needed to solve or complete a problem (Land, 2000b).

- **Link student performance in the environment with the science and engineering learning objectives.**

  We employ evidence-centered design (Mislevy and Haertel, 2006) as an overarching framework to systematically integrate science and engineering disciplines and to align curricular activities and assessment tasks to the science and engineering concepts and practices. ECD facilitates coherence in system and assessment design by explicitly linking claims about student learning, evidence from student work products, and the instructional and assessment tasks that elicit the desired evidence. We leverage this framework to establish the features of the learning environment, the interactive tasks students perform in this environment, and the instructional and assessment tasks that are built into our environment.

3.3.2 Curriculum Integrating Computing and Engineering into K-12 Science

Figure 3.3 illustrates our designed learning trajectory and criteria from conceptual to computational model to finding engineering design solutions.

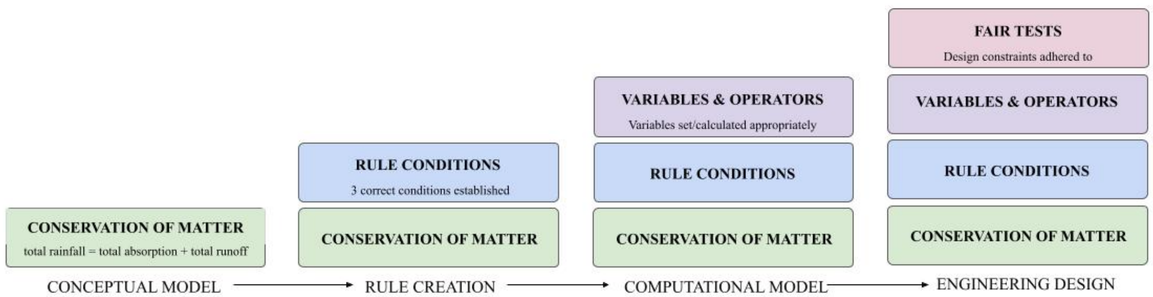


Figure 3.3: Learning trajectory from conceptual to computational model tasks (extended from (Authors, nd).

Initially, students were expected to understand the conservation of matter principle (representing a scientific principle or law) by conducting real physical experiments and drawing conclusions from the observed experimental results. In this domain, students are presented with scenarios where there is rainfall, some of the rainfall is absorbed into the ground materials and the remaining amount is runoff. This translates to "Total rainfall = total absorption + total runoff". In the SPICE curriculum, students are introduced to the science concepts of matter conservation and the absorption characteristics of different surface materials. Students begin with engaging in a series of hands-on activities involving experimenting with physical materials commonly available in schoolyards and playgrounds and then contrasting the absorption capabilities of these different surface materials. After acquiring a basic understanding of the runoff scenario, students develop conceptual, pictorial representations that express the amount of water runoff as the difference (if any) between the total rainfall and water absorbed by surface materials. For instance, to complete Figure 3.4(a), students are tasked with predicting the amount of absorption and runoff for 3 inches of rainfall and a 1-inch absorption limit of the surface material. As a second step in the model evolution process, the students then create a more precise conceptual model on paper, where they create rules to describe the three different runoff conditions.

Each subsequent modeling form required application of additional CT concepts to specify the model in

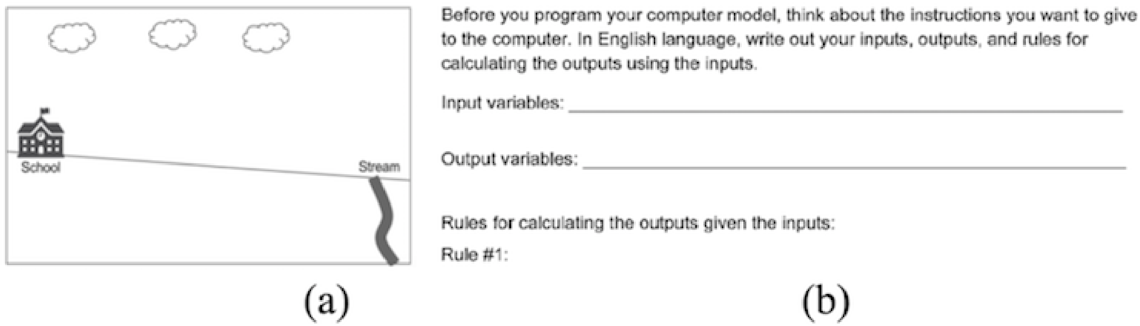


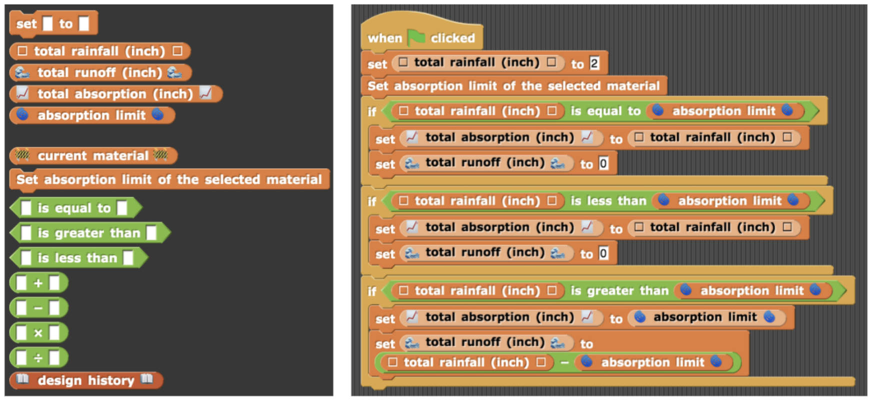
Figure 3.4: Task examples for paper-and-pencil conceptual modeling (a) and rule creation (b).

a more general form (see Figure 3.3). To support this, we implemented an intermediate paper-and-pencil Rule Creation task (Figure 3.4b) to elicit an additional representation of the science phenomenon enabling students to express the relation between the science concepts: total rainfall, total absorption, absorption limit, and total runoff. Students are tasked with expressing three scenarios (i.e., when rainfall is greater, less than, and equal to the surface absorption limit) as semi-structured rules. These relations take into account the conservation laws while using conditional logic expressions to specify when different situations apply (e.g., no runoff versus a certain amount of runoff). Students then transfer their rules into a computational model using the given DSML blocks to create the model components (i.e., the three rules).

Translating the rules to the computational modeling activity requires additional knowledge of variables and mathematical and relational operators. To support this, we first implemented a series of unplugged activities to engage students in variable, equation, and conditional logic representation. This was followed by the construction of their computational models. using DSML constructs (Figure 3.5a.) that facilitate the translation of the runoff rules into the computational model (Authors, nd). The DSML blocks help students assign variables to specific values, and translate their runoff rules to "if" constructs (e.g., "if total rainfall is greater than the absorption limit", then "set total runoff to [total rainfall — absorption limit]"). Students also needed to assign the value of total rainfall and the absorption limit before the conditional block statements.

In addition to testing various total rainfall amounts throughout the computational modeling process, a visual interface (Figure 3.6) allows students to populate individual playground squares with 1 of 6 available schoolyard material (defined in Table 3.1) in order to test their models on different material options. The system calculates the total runoff and cost based on their input and selected material and provides the results in the visual interface (as seen in the left of Figure 3.6). By the end of the specified time (determined by the classroom teacher), students who have not achieved the correct model are supported in adjusting their models for the engineering design task.

Finally, students use their computational models to solve the engineering design problem - redesigning



(a) (b)

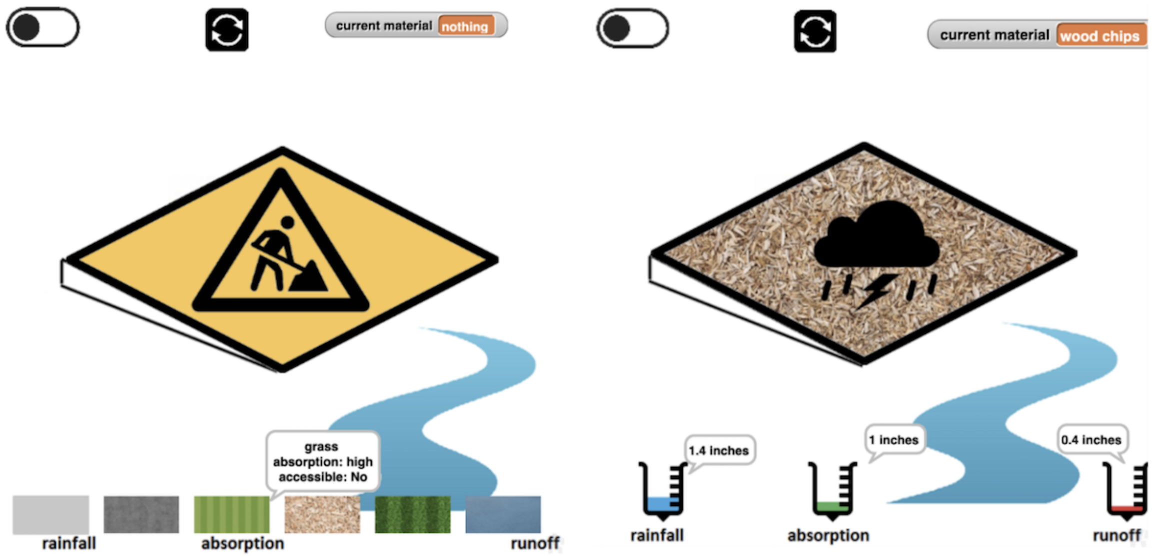
Figure 3.5: Earth Science DSML (a) and example computational model (b).

Table 3.1: Schoolyard material information.

Material	Absorption Limit	Cost	Accessible
Concrete	0.1 inches	$37,500	Yes
Permeable Concrete	1.3 inches	$93,750	Yes
Natural Grass	1.2 inches	$18,750	No
Wood Chips	1.0 inches	$37,500	No
Artificial Turf	0.6 inches	$112,500	Yes
Poured Rubber	1.2 inches	$187,500	Yes

their schoolyard to minimize the amount of water runoff after a storm while adhering to a series of design constraints in the overall cost, accessibility, and different utilities of the schoolyard. To do so, students should apply fair tests as they explore the design space and come to an optimum solution for their playground design. We detail each activity, below. An extended version of the runoff model is implemented under the hood to support the 4x4 schoolyard design interface. Figure 3.7 (on the right) illustrates the visual interface for the engineering design task, with a photo of the school terrain this visual interface was modeled after on the left. Each square in the 4x4 model has an area of 37,500 sq.ft. In this task, students implement a search process to find the optimum solution to minimize runoff, while meeting cost (a design must have a total cost less than $750,000) and accessibility (there must be at least 6 squares with accessible materials). To do so, students can click on squares (for instance, the yellow square shown in Figure 3.7) and select their desired material (from the materials found in Table 3.1). Students document chosen design solutions throughout this process.

In designing each of the model building activities, we maintained coherence across the three representations, and gradually introduced students to CT concepts and practices with support of activities such as the Rule Creation task. This approach provides a framework for evaluating students' modeling artifacts across



(a) (b)

Figure 3.6: The computational modeling visual interface for selecting material (a) and the resulting calculations (b).

different representations and understanding how these representations support students' learning trajectories.

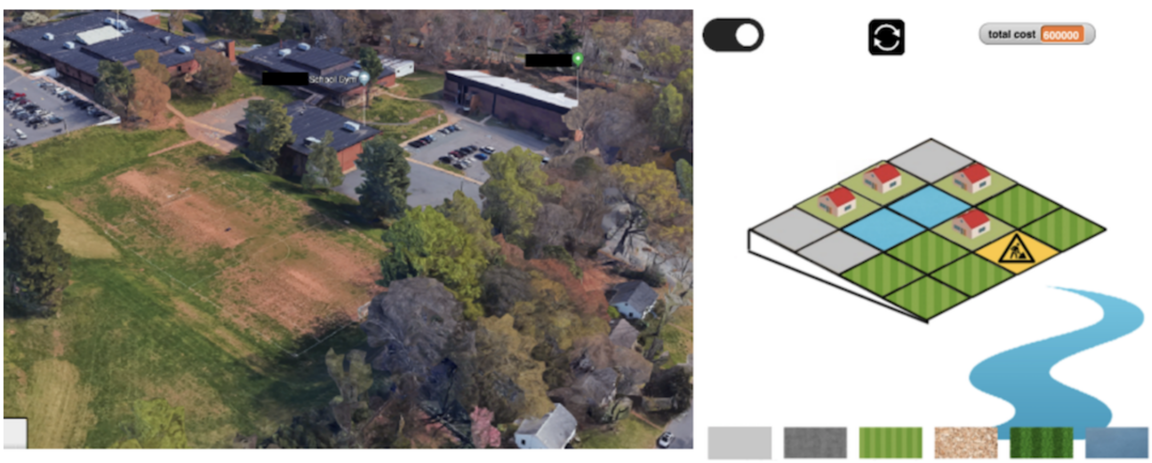


Figure 3.7: SPICE engineering design task.

3.3.3 Assessments

Leveraging our ECD process, we developed a system of summative and formative assessments to evaluate student learning in identified target science, CT, and engineering concepts and practices over the course of the 3-week curriculum. This includes a summative assessment that is split into the three domain sections and is implemented prior to and directly following the curriculum. Our science and engineering pre-post assessment aligns with a number of NGSS Performance Expectations (PEs). The CT assessment tasks were aligned with the concepts and practices that students perform as part of their science modeling activities (e.g., variables, operations, conditionals, program development). The rubrics used for coding and scoring these assessments were updated from our previous work (Authors, nd). Similarly, our formative assessments were also developed during the ECD process and target key concepts and practices pertinent to the curricular unit each assessment was scheduled around (e.g., conceptual modeling, computational modeling, engineering design).

All formative and summative assessment items were labeled to identify the key concepts and practices targeted in each domain and discussed in the previous section. For instance, Figure 3.8 illustrates a tree-like model that traces applications of key domain-specific concepts and practices (this is a simplified example) in the key tasks that compose the Computational Modeling unit in SPICE (the computational model, written responses in the packet, and a formative assessment). This framework allows us to (1) ensure that all target domain concepts targeted in the curriculum were assessed through multiple representations and (2) track student performance in each concept or practice over time based on the labeled submissions.

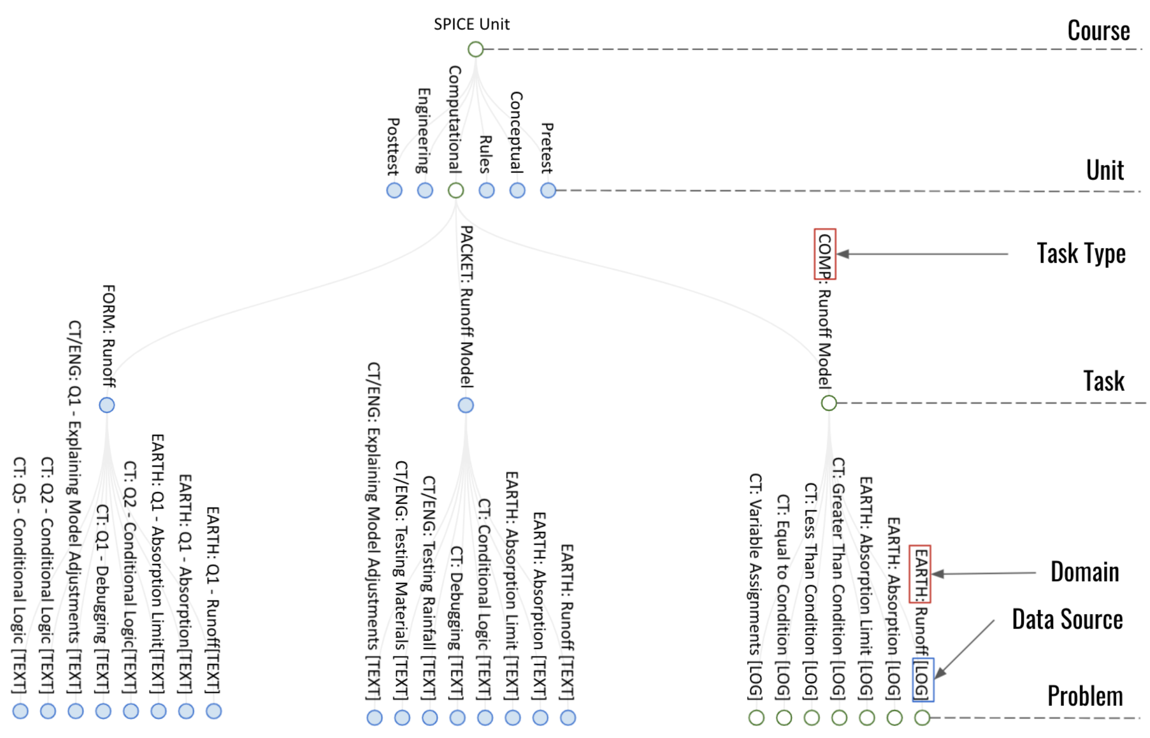


Figure 3.8: Mapping applications of science, CT, and engineering concepts and practices throughout the SPICE curriculum.

3.4 Methods

3.4.1 Study and Participants

This exploratory analysis leverages student data collected from a classroom study with 99 students in a 6th-grade classroom in the southeastern United States. The study, led by two experienced science teachers, was run in the Fall of 2019. Three Vanderbilt University researchers provided additional support in the classroom. The two teachers participated in four days of professional development conducted by the research team during the summer session before the study. Elementary programming classes are part of the middle school curriculum in this school, and all participating students had varying amounts of prior programming experience with Scratch (https://scratch.mit.edu/about).

The SPICE curriculum was covered by the teachers with intervening student work on the system for 45 minutes per day, three days a week in their regular science classes, and 75 minutes, twice a week that included additional personalized-learning time. The curriculum was covered in 15 school days, with NGSS-aligned science and engineering and CT pre-post assessments administered during two additional 45-minute class sessions. Procedures of this study were approved by the ABC Institutional Review Board.

3.4.2 Data Collection

Data collection occurred in three phases: paper notebooks and assessments, computational modeling and design submissions, and logged actions.

3.4.2.1 Student Notebooks and Assessments

Students completed all written response activities, including the physical experiments, conceptual models, unplugged activities, short-answer questions during computational modeling and engineering, and their formative and summative assessments on paper-and-pencil packets. Students used the same non-identifiable username on each submission as well as to log into the SPICE OELE. These paper-based tasks were coded and then graded using rubrics to be discussed in Section 3.4.3.1.

3.4.2.2 Supporting Automatic Analysis of Computational Models Using Abstract Syntax Trees

We have developed analytical tools to represent students computational models as abstract syntax trees (ASTs) (Bille, 2005) to compare the tree edit distances (TED) to correct implementations (cf., Bille, 2005). ASTs provide a flexible, compact, and extendable representation, where semantic information can be embedded into the tree structure to enable more in-depth analysis of a program (Rabinovich et al., 2017). ASTs, often used in compilers to represent a computer program's structure (Grosch and Emmelmann, 1990), have been used for code evaluation (Baxter et al., 1998) and to examine students' code construction processes

(Neamtiu et al., 2005). Our research lab has leveraged the latter approach to evaluate students model construction processes in CTSiM (Basu et al., 2014) and final code performance in SPICE (Zhang, 2020).

This approach is optimized by the use of DSMLs, as the modeling language is the same for all students and not prone to differences resultant when students are required to construct and name custom blocks (Hutchins et al., 2020b). Moreover, the development of the DSML representations is aligned with the learning design and objectives of the curriculum to support linkages between the computational model build and concepts and practices targeted through assessment.

We apply semantics-preserving transformations to standardize the ASTs (e.g, equalizing semantically isomorphic expressions total rainfall $<$ absorption limit and absorption limit$>$total rainfall) (Xu and Chee, 2003). Therefore, syntactically different but semantically isomorphic code representations will have an equal TED result. Figure 3.9 shows a snippet of a standardized AST reconstructed from the log data. This snippet contains the subtree specific to the "total rainfall is equal to absorption limit" rule (or condition).

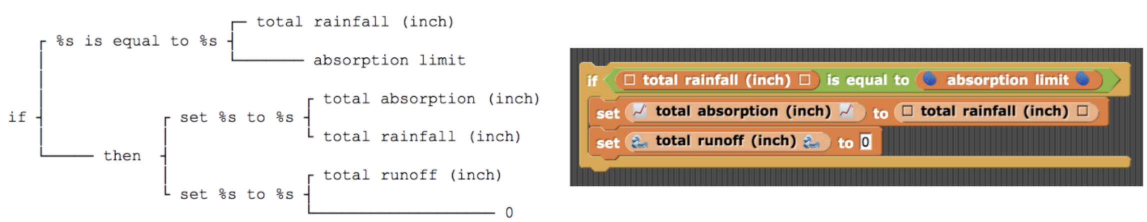


Figure 3.9: Abstract syntax tree representation of equal-to subtree in SPICE computational model

3.4.2.3 Modeling and Engineering Activity Data Processing

Student actions in the SPICE environment were recorded in two parts: (1) logged timestamped student actions and (2) interaction with the visual interface and associated simulation composition (e.g., the design history and system calculated total runoff). The latter were recorded using the Cloud Variables RPC (Broll, 2018).

Student computational modeling actions are recorded in log files with timestamps. Interpreting actions is more productive if we can associate them with the specific goals students may have when performing a set of actions. For analysis, student actions are represented at a level of abstraction, so that action patterns can be interpreted as students' model-building processes with semantic and contextual interpretations Werner et al. (2013). In order to represent students' actions at a level of abstraction that makes it easier to interpret their model construction and debugging behaviors, following previous work (e.g., Basu et al., 2017), we created a task model that is illustrated in Figure 3.10. We extract and interpret students' action sequences during computational modeling that are linked to code assessment and code construction.

Finally, Figure 3.11 shows example actions logged with the Cloud Variables RPC, including RunSim-

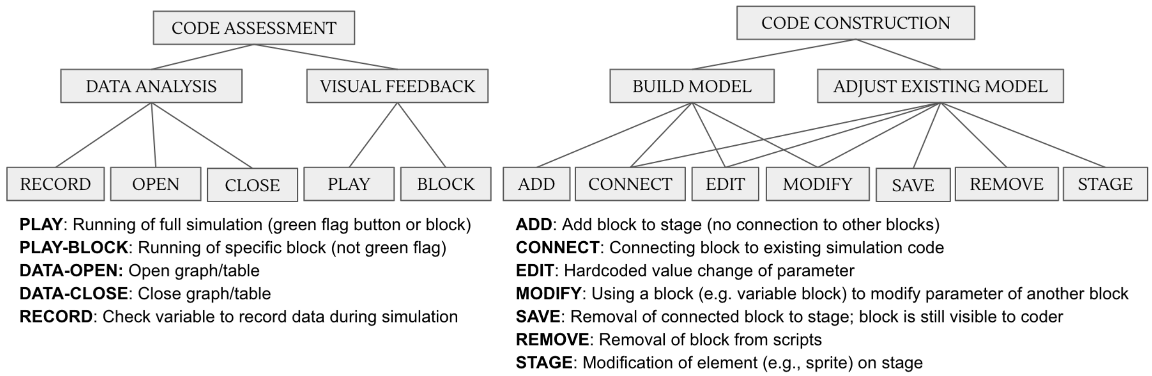


Figure 3.10: C2STEM computational modeling task model

ulation and ChangeMaterial. For the ChangeMaterial logged action, we evaluated when the student was changing the material of a square in single mode (one square representation seen in Figure 3.6) or in the 4x4 mode (seen in Figure 3.7). For the single mode, we are able to monitor how many different materials each student tested during the computational modeling phase. In the 4x4 mode, we track the total cost and the total runoff limit of their built design, the total number of squares not in the default state, and the overall status (e.g., how many blocks of each material are present). When a student clicks the green flag to test their design, the RunSimulation log records the total rainfall, total absorption, and total runoff as well as the construction status at the time of the RunSimulation.

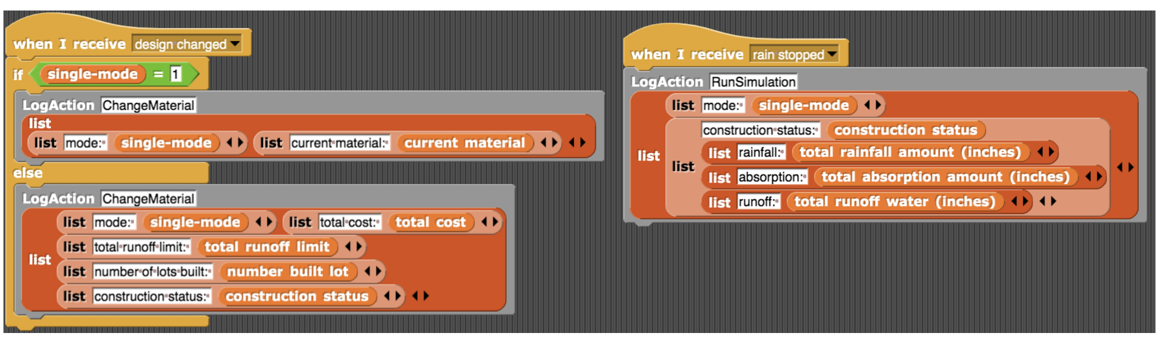


Figure 3.11: SPICE engineering design logs (Zhang, 2020)

3.4.3 Analysis Techniques

To answer our research questions, we (1) leveraged our ECD-developed rubrics to score student tasks and assessments and applied statistical analysis to explore the impact of the intervention on students integrated learning, (2) applied correlation analysis and *Path Analysis* to examine the impact of computing on the learning of science and engineering, and (3) implemented an exploratory clustering algorithm to describe aggregate characteristics of students' behaviors as they relate to learning in each domain.

3.4.3.1 Scoring Student Packets and Assessments

Students completed a paper-and-pencil pre-post assessment that was split into a science and engineering component and a CT component (McElhaney et al., 2020). Science and engineering tasks were aligned with a number of Next Generation Science Standards (NGSS) Performance Expectations in Earth science and engineering (NGSS Lead States, 2013). Of the five science and engineering tasks, three asked students to apply an engineering practice to the solution of a water runoff problem, one asked students to develop a model of water runoff in a context different from the SPICE, and one asked students to apply an engineering practice to the solution of a problem unrelated to water runoff. For example, in one task, students compare two street designs based on criteria related to water runoff performance, usage, and cost. The science and engineering task rubrics measured the extent to which students could apply the focal science and/or engineering practice to the water runoff context and/or other engineering design criteria. The CT assessment tasks were aligned with the CT concepts and practices addressed in the runoff CM. Students could score a maximum of 23 points on science and engineering and 13 points on CT. The rubrics used for coding and scoring the tasks were updated from our previous work. Two researchers received 5 hours of training on the rubrics, and following an interactive grading procedure established inter-rater reliability (Cohen's κ at 0.8 level on all items). All differences in the coding were discussed and resolved before the remaining submissions were graded by a single researcher.

Students' final runoff compurational models were logged and scored using a rubric targeting conservation of matter rules and conditional statements for the different rules. The rubric rewarded four main criteria of the compurational models: (1) assigned appropriate values to the total rainfall and absorption limit variables, (2) included code for three conditional statements that compared the total rainfall and absorption limit variables, (3) updated the absorption and runoff variables correctly for each of the three conditions, and (4) code is generalizable (e.g., assigned variables to expressions rather than to hardcoded numeric values).

Students completed a paper-and-pencil workbook over the course of the unit. We scored and analyzed student responses to specific workbook activities such as the conceptual modeling tasks, dice game activities, test case identification tasks, and a CM debugging task. A series of four student conceptual models and corresponding written explanations were scored based on their adherence to the conservation of matter principle and articulation of the causal relations governing the flow of rainwater. We also scored student performance on the dice games that introduced students to CT skills. Each of the four games were scored based on whether students could correctly evaluate conditional statements and interpret variable assignment statements . We report dice game scores only for students who documented outcomes from all four games (n = 62). We scored students' choice of test cases for testing the three rules in the runoff CM for a material of their choice based on whether the test cases address each of the model's three rainfall conditions. We scored a debugging task where the variable assignment for the absorption and runoff variables were flipped for a particular condition.

Finally, we scored student responses to formative assessments based on time of implementation. Conceptual modeling formative assessments targeted students' ability to interpret and evaluate provided conceptual models based on their developing science knowledge. Computational modeling tasks assessed students' ability to predict the output of given computational models, identify errors, and fix buggy computational models. Finally, engineering design assessments targeted students' ability to implement fair tests and evaluate design solutions.

3.4.3.2 Evaluating Student Performance Over Time

Our initial exploration targets the identification of correlations between curricular tasks and learning performance in science, computing, and engineering. We implement Spearman's correlation on the normalized student results (scored using the ECD rubrics, described above) on key CT tasks.

To extend this analysis, we apply *Path Analysis* to study the relationship between science, engineering, and CT learning in SPICE. We hypothesized that students' knowledge gains, the behaviors they developed, and their performance in the tasks they worked on would influence their learning, behaviors, and performance in subsequent tasks and assessments in the SPICE curriculum. Path Analysis can be seen as a variation of Structural Equation Modeling (Kline, 2015) without the latent variables. In many ways, the methods used for path analysis can be related to work in deriving Bayesian causal models from data (Ellis and Wong, 2008; Hagger and Hamilton, 2018). We hypothesized the causal paths shown in Figure 3.12 among the different tasks students performed in the SPICE. In particular, we used Path Analysis to study the relationship between science, engineering, and CT learning in the SPICE, with a particular emphasis on the CT tasks. We hypothesized that students' knowledge, behaviors, and performances could influence their subsequent learning behaviors, performances, and summative test scores in the SPICE curriculum. Conceptually, we hypothesized causal paths as shown in Figure 3.12. Each arrow in the diagram indicates a direct effect on the endogenous variable from the exogenous variable. The horizontal positions of the variables also correspond to their temporal order in the SPICE curriculum (pre-tests $\prec$ unplugged CT activities $\preceq$ computational modeling $\prec$ post-tests).

3.4.3.3 Grouping students based on computational modeling and engineering design behaviors

The open-ended nature of computational modeling and engineering design paves the way for students to adopt different learning and problem-solving strategies, thereby demonstrating a variety of learning behaviors. Due to these varieties, our approach targets forming aggregate characterizations of student behaviors during the

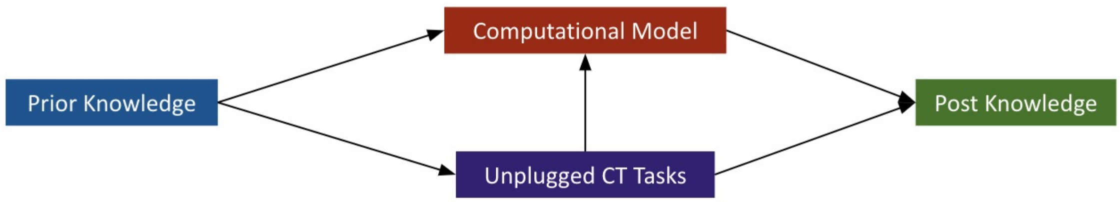


Figure 3.12: A hypothesized path model of the direct effects on the different categories of learning behavior and performance variables

Table 3.2: Selected features (marked with an *) and descriptions.

Metric	Description
Computational Model Edits*	Percentage of model construction actions
Computational Model Testing*	Percentage of single-mode model testing (with material selection)
Computational Model Edit Size	Average number of consecutive model edits between play actions
Unique Rainfall Tests*	Percentage of unique values of rainfall tested
Avg Material Tests*	Average tests per unique material
Materials Tested*	Percentage of possible materials tested
Engineering Design Edits	Percentage of engineering design edit actions
Total Complete Design Tests*	Proportion of engineering design tests per engineering design actions
Engineering Design Edit Size	Average number of consecutive design changes between design tests
Satisfying Tests*	Percentage of satisfying design tests out of all tests conducted
Euclidean Total*	The extent to which a learner explored the eng. design experiment space
Lowest Runoff Score	The lowest calculated runoff value for all satisfactory design tests

computational modeling and engineering design units, rather than evaluating individual behaviors. Features selected for this exploratory clustering analysis are detailed in Table 3.2. Initial features were selected based on past analyses (e.g., Zhang et al., 2017, 2022). We apply a correlation-based filter method comparing the normalized mean values for features to students science, CT, and engineering learning gains to evaluate each feature's significance in the learning process. Features that were not highly correlated were removed (for instance, the Average Design Edit Size, see Table 3.2) and features that were highly correlated, but were also highly correlated with other features were filtered based on researcher input. After this pre-processing step, we ran the K-means clustering algorithm on the selected feature set to group students by common behavior characteristics across the two units. We used silhouette analysis to identify the optimal cluster size. Finally, we evaluate resulting groups based on their summative and formative task performances in each domain, discussing the potential impact learning behaviors of each cluster had on domain-specific learning.

3.5 Results

3.5.1 RQ1: How effective is our integrated SPICE learning by modeling and problem solving curriculum, with its sequencing of science, computing, and engineering design support student learning across the three disciplines?

Students' pre-post test scores were compared to determine their learning gains in science, engineering, and CT. We conducted Kolmogorov-Smirnov tests on the assessment scores and found that the data was normally distributed. Therefore, we used the paired t-test to measure the statistical significance of the difference in the pre-post scores. As shown in Table 3.3, all differences are statistically significant with moderate (≥ 0.5) to large (≥ 0.8) effect sizes.

Table 3.3: Students' (n=99) learning gains and effect sizes (Zhang et al., 2022)

	Total Points	Pretest(*SD*)	Posttest(*SD*)	p-value	Cohen's d
Science	7	4.56(1.03)	5.13(1.04)	<0.001	0.54
Engineering	16	8.73(2.62)	10.50 (2.67)	<0.0001	0.67
Computing	13	6.23(2.60)	8.41(2.69)	<0.0001	0.83
Overall	36	19.52(4.47)	24.03(4.39)	<0.0001	1.02

Table 3.4 provides a break-down of scores for high- and low-performing students based on the median scores of the pretest and the posttest (Zhang, 2020). The results show that although 12 low performing students moved into the high-performing category, the majority of the low performers remained low performers. As such, further work needs to be conducted to improve support for low performers. We discuss this further in the Conclusions.

Table 3.4: Confusion matrix of qualitative change in summative assessment performance.

		Posttest	
		High Performers	Low Performers
Pretest	High Performers	36	12
	Low Performers	12	39

3.5.2 RQ2: What is the role of computational thinking in facilitating learning of science and engineering concepts and practices?

To evaluate this question, we first conducted correlation analysis using Spearman's correlation investigating the impact of key curriculum tasks. The calculated Spearman's ρ values (Kokoska and Zwillinger, 2000) are reported in Table 3.5. Statistically significant correlations are marked in bold font. The first three columns are key CT tasks that may impact student learning over SPICE, including the pre-CT test, the unplugged Dice

Game and Debugging tasks (described in the SPICE section), and Rule Creation Task (Figure 3.4b). The remaining columns include the pre-Science and Engineering test, the Conceptual Model (Figure 3.4a), the Computational Model (Figure 3.6), and Engineering Design Max Score (Figure 3.7). Finally, the bottom-two rows include the the Science and Engineering posttest and the CT posttest, to correlate performance in each column with overall learning performance.

A key results from this analysis is the correlation between the conceptual model, the rule creation task, and the debugging task on science and engineering post-test performance. For CT, the rules, dice game, test cases, and the computational model score were highly correlated with post-test CT performance. It is important to note the significant correlation result between pre and post test performance, which is associated with the identification of needed support for lower performing students in the previous section.

Table 3.5: Spearman's ρ's of the CT scores (statistically significant correlation coefficients, ** $p < 0.01$, * $p < 0.05$)

	Pre-CT	Rules	Dice	Debug	Pre-Sci+Eng	Concept.	Comp.	Eng. Score
Rules	0.184							
Dice	**0.20***	**0.36****						
Debug	**0.26****	**0.35****	**0.31****					
Pre-Sci+Eng	**0.39****	**0.30****	0.18	**0.23***				
Conceptual	0.19	**0.32****	0.14	**0.28****	**0.21***			
Computational	0.20	**0.32****	0.06	0.09	0.13	0.05		
Eng Score	0.14	-0.05	-0.16	-0.11	0.10	-0.07	-0.07	
Post-Sci+Eng	0.07	**0.47****	0.18	**0.26****	**0.41****	**0.39****	0.04	-0.10
Post-CT	**0.62****	**0.31****	**0.20****	**0.23****	**0.43****	0.20	**0.24****	0.08

We hypothesize that the rule creation task and the debugging task prove to be key CT applications to support (1) multiple representations of the key science phenomenon linking each domain (e.g., representing and exploring difficult science concepts in computational form) and (2) applications of key problem-solving skills that require applications of developing science knowledge to complete and that systematic testing and evaluation practices. For the rules, students needed to apply conditional logic to what they learned in the conceptual modeling section. For the debugging task, students needed to leverage their science knowledge to identify errors in a provided computational model and correct those errors.

Based on these results, we then implemented Path Analysis considering our hypothesized path model illustrated in Figure 3.12 using the IBM®SPSS®Amos software. We combined the pre- and posttest performance to evaluate the impact of the CT tasks on overall student learning. We modeled a total of 10 direct effects from the 5 variables in the path diagram. These included the pretest, posttest, and computational model score as well as the Rules Creation Task and Debugging, identified as highly correlated to student learning above. Results are shown in Figure 3.13. We calculated the model-fitting statistics as compared to a baseline model (Schreiber et al., 2006): $\chi^2 = 75.472$ (DF=10, p-value < 0.000). The root mean square error of approximation (RMSEA) was < 0.000 (< 0.06 threshold) and the comparative fit index (CFI) was 1.000 (> 0.9 threshold). In Figure 3.13, statistically significant correlation coefficients are labeled as ** $p < 0.001$ and * $p < 0.01$.

These results extend our previous findings, above. Of note is the Rule Creation Task and its impact on the computational model score and overall student learning. We hypothesize that the computational model score may not be indicative of student learning, as student may have reached a threshold following the Rules Creation Task and its support of the translation into computational form. It is important to discuss the high significance of the pretest performance on posttest performance. This relates to our findings for RQ1 and further emphasizes the need for additional work to better support and engage students that begin with lower prior knowledge scores.

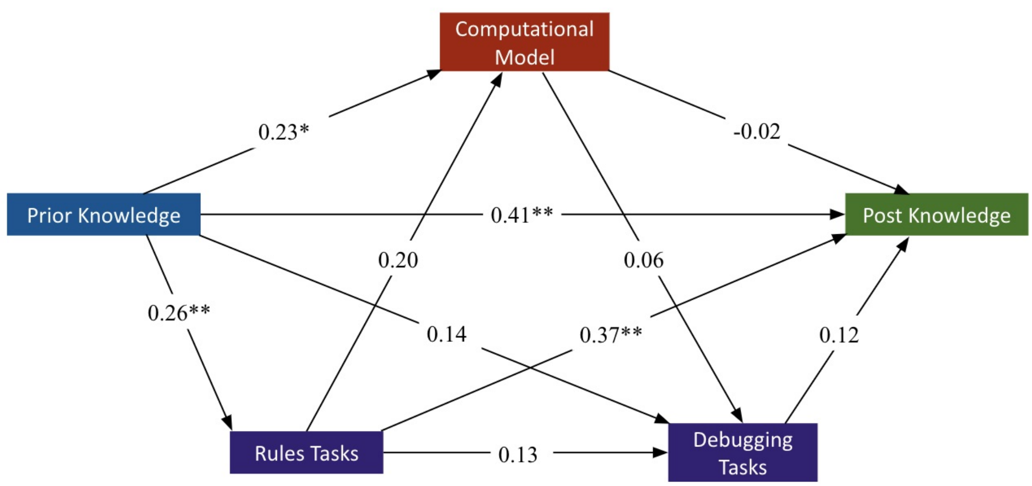


Figure 3.13: Discovered causal paths.

To further our analysis, particularly the impact of model representations and applications of key CT skills and processes seen in these results (such as debugging), we applied cluster analysis to evaluate student group behaviors as they transitions from computational modeling to engineering design and how those behaviors may have impacted learning.

3.5.3 RQ3: What kind of effective and ineffective strategies do students employ in their model-building tasks and how do these strategies correlate with their science, CT, and engineering learning?

We applied the K-means clustering algorithm on the subset of metrics picked by feature selection. Using silhouette analysis, we identified the optimal cluster size of 3. The Euclidean metric was used as the distance measure, and 1000 random restarts were performed to mitigate the effects of initial cluster center selection. Table 3.6 summarizes the mean values (and standard deviations) of the metrics for all of the derived clusters. Table 3.7 provides the mean values (and standard deviations) of each clusters pre-post performance and learning gains in each domain. Finally, Table 3.8 provides each cluster's results on key curriculum tasks over the course of the integrated curriculum.

Table 3.6: Cluster means (and standard deviations) of computational modeling and engineering design behaviors.

Metric	Play-it-Safers (n=34)	Explorers (n=53)	CT Strategists (n=12)
Computational Model Edits %	69.65(8.10)	56.08(7.84)	54.45(12.89)
Computational Model Testing %	17.00(5.67)	33.24(9.60)	30.56(11.33)
Unique Rainfall Tests %	23.54(8.25)	24.13(7.03)	46.26(20.83)
Avg Material Tests	2.73(1.36)	2.03(0.92)	4.14(2.30)
Materials Tested %	37.75(18.94)	74.84(21.59)	31.94(11.14)
Total Complete Design Tests %	7.26(4.57)	9.72(5.46)	18.38(14.80)
Satisfying Tests %	73.88(24.94)	55.64(19.37)	56.34(25.65)
Euclidean Total	8.26(7.36)	33.07(23.96)	20.47(20.89)

Table 3.7: Means (and standard deviations) of learning performance by cluster.

Cluster	Play-it-Safers	Explorers	CT Strategists
SCI Pretest (7)	4.26(0.93)	4.74(1.04)	4.67(1.15)
SCI Posttest (7)	4.94(1.07)	5.25(1.05)	5.17(0.94)
SCI Learning Gains	-0.06(0.82)	0.17(0.56)	0.10(0.61)
ENG Pretest (16)	8.22(2.56)	9.06(2.36)	8.71(3.75)
ENG Posttest (16)	9.81(2.18)	11.08(2.46)	9.83(4.12)
ENG Learning Gains	0.35(0.35)	0.17(0.66)	0.16(0.55)
CT Pretest (13)	6.01(2.66)	6.25(2.56)	6.75(2.73)
CT Posttest (13)	8.50(2.92)	8.34(2.60)	8.46(2.59)
CT Learning Gains	0.18(0.50)	0.40(0.38)	0.00(1.34)

The first cluster is defined by their high percentage of construction and material change actions (high percentage of computational model edits and low percentage of complete design tests to engineering design actions), their low testing variety (low unique rainfall, materials tested, and euclidean distance), and their

Table 3.8: Means (and standard deviations) of performance in formative tasks (max scores in parentheses).

Cluster	Concept. (4)	Rules (10)	Dice Game (8)	Comp. (15)	Eng Max Score (4.25)
Play-it-Safers	3.09(0.62)	6.61(3.09)	6.03(2.28)	13.47(3.05)	3.79(0.40)
Explorers	3.10(0.80)	7.02(2.87)	6.43(1.83)	13.51(2.18)	4.00(0.21)
CT Strategists	3.50(0.52)	8.18(2.52)	7.08(1.00)	14.42(2.02)	3.91(0.27)

high number of satisfactory engineering design tests conducted. Overall, this group tended toward a more depth-first construction and design approach. During tested, they not only tested a low number of materials, but the average test per material was also low, indicating they did not leverage the computational modeling testing tools to explore material types. However, the group implemented a high number of single mode tests (27.76(19.66)), indicating they may have utilized the same rainfall values to support model construction or debugging. This limited exploration is also seen in engineering, with this group instead demonstrating a low number of design tests, but of those, a high number of satisfactory designs.

In terms of domain-specific performance, this group started with the lowest average pre-test performance in each domain. These difficulties translated into their low conceptual, rules, dice game, and computational model performance. By the post test, the group demonstrated the high highest learning gains in engineering 0.35(0.35) and the highest post-test performance in CT (learning gains averaged 0.19(0.50)). However, they averaged negative learning gains in science (-0.06(0.82)).

Overall, we hypothesize that improvements in this group's testing variety may support leveraging CT as a bridge, leveraging the science and engineering practice of planning and carrying out investigations from NGSS (NGSS, 2013) to link material testing outcomes to what they completed during their science experimentation and conceptual modeling and systematically planning engineering design changes based on what they investigated. Positive reinforcement for this group could include their use of CT skills (another NGSS science and engineering practice), including implementing consistent, repeated testing (e.g., re-trying rainfall values to test conditional statements, or the rules, against their expectations) and design prototypes (thinking through each design tests resulting in a high number of satisfactory tests).

The second cluster is defined by their high testing variety. This group demonstrated (1) the highest percentage of Code Assessment actions (e.g., Computational Model Testing), (2) the highest percentage of materials tested during computational modeling, and (3) the highest euclidean total during engineering design testing. Added to this is the fact that this group completed the highest number of single mode tests (37.98(20.37)) and engineering design tests (37.06 (18.70)). Two factors seemed to impact this successful application of this approach: the low average tests per material (it was below 3 indicating that each of the three rules was not checked per material) and the low testing percentage during engineering design (indicating the group may have implemented major design changes each time and not fair tests). Moreover, this group had the lowest percentage of satisfactory engineering design tests.

In terms of domain-specific performance, this group began the curriculum as the highest science and engineering performers on the pre-test, remaining the highest with slight average learning gains in each domain (science, 0.17(0.56); engineering, 0.17(0.66)). The group demonstrated the highest average learning gains in CT (0.40(0.37)), but in had the lowest post-test average in the domain. Their formative tasks were akin to cluster one.

Overall, we hypothesize this group may also benefit from scaffolding or instructional support on productive testing strategies. Distinct from the first cluster, this group may need support on the science practices of obtaining, evaluating, and communicating information and developing and using models. For instance, their unsystematic approach to testing may be supported by tasks that force them to evaluate and reflect on each test based on their developing science knowledge, and explain why they are changing a rainfall value or material for subsequent tests. Positive reinforcement for this group includes the variety in material testing, which could have been resultant from their stronger prior science and engineering knowledge. This approach may have supported the contextualization of different CT concepts, such as conditional logic, which has been shown as a benefit of scientific computational modeling (e.g., (Sengupta et al., 2013)).

The final cluster is defined by their unique computational modeling testing approach. This group indicated the highest percentage of code assessment actions and unique rainfall tests. In addition, of the materials tested (albeit, a low number of materials tested) the group had the highest number of different tests per material (above 3, indicating they potentially checked each rule for each material). Their engineering design behavior differed from the other two groups in that they demonstrated the highest percentage of tests per actions. The group also implemented a low average number of satisfying tests, while falling in between in terms of the solution space explored. Interestingly, this group completed the highest total number of engineering tests (41.67 (34.40)).

In terms of domain-specific performance, this group began as the strongest CT group on the pre-test, which seemed to support their performance in the Rules and Dice Game unplugged activities as well as achieving an almost perfect computational model score. This intervention also supported learning gains in science (0.10(0.61)) and engineering (0.16(0.55), the lowest overall learning gains in this domain), performing as the middle group in each domain. It is interesting to note that the group performed the best on the conceptual model activity, perhaps supported by their strong pre-science and CT tests.

Overall, we hypothesize this group could be supported by (1) the science and engineering practice of planning and carrying out investigations (NGSS, 2013) and (2) increased variety in material testing to support the linkage between science, computing, and engineering. The productive testing that occurred to construct and debug the computational model did not translate to successful engineering design tests. This group seemed to lean on their strong CT background to construct a correct computational model, but may not have leveraged all computational tools to link the domains. Positive reinforcement for this group may include that strong computational testing approach (part of the practice of developing and using models), and in particular, strategically testing each material multiple times (perhaps to check their rule implementations).

The three groups demonstrate successes in and opportunities for how model-building and problem-solving behaviors can support students' learning in science, CT, and engineering. On the one hand, rainfall testing strategies seem to support successful CT applications, and in particular, checking if each conditional block behaves correctly. And on the other, material testing in the computing phase may be beneficial in supporting the linkage between computing and science and engineering, connecting computational output to prior science experimenting, and preparing students to think critically about engineering design changes. It is important to note that no group excelled in every domain and top performers in each domain had room for growth. We believe this indicates that our curriculum provides a unique opportunity to engage with all students.

3.6 Conclusions and Future Implications

This research provides one of the first instances of an NGSS-aligned curricula that provides an integrated framework for science, CT, and engineering. Using the SPICE learning environment, students constructed multiple, linked representations to model the water runoff phenomenon and developed key problem-solving skills to construct computational models and use them to solve a complex engineering design problem. Our findings indicate that this integrated curricular approach supports learning in each domain and the significant correlations between science, engineering, and CT learning indicates the synergistic nature of that integrated learning. In addition, this research examines how computing can strengthen the existing connections between science and engineering. We have demonstrated the role of computing in (1) providing a resource for addressing complex science and engineering problems that can't be physically represented and (2) supporting the development of testing and problem-solving skills.

To answer research question one, we demonstrated science, CT, and engineering learning over time through a system of assessments, including the domain-specific pre-post tests, formative assessments, and curriculum tasks, scored using ECD-developed rubrics. The application of these assessments targets the dearth of domain-specific formative assessments in each domain as well as an innovative curriculum approach for integrating both computing and engineering at the lower and middle school grade levels.

To answer research question two, results from Path Analysis demonstrate that students' learning in SPICE is strongly associated with applications of computing skills. In particular, unplugged CT activities, such as the Rule Creation and Debugging tasks demonstrated significant impact in student learning in science and engineering. We hypothesize this is supported by students' learning through multiple, linked representations. Such learning approaches can provide students a more complete depiction of a difficult science phenomena and support students in deriving linkages between the representations (Ainsworth, 2006; Frederiksen et al., 1999). These tasks supported the translation of science knowledge into computational form (in this case, creating pseudo-code and using science knowledge to debug a code-snippet provided on paper). Coupled with the benefits of productive computational modeling behaviors (such as robust testing of rainfall and materials), we believe these findings relate to the leveraging of computational models as tools to simulate phenomena that are difficult to replicate physically, as students first understood the underlying science represented in the computational model and were then able to try a variety of materials (a gap indicated in the literature review).

To answer research question three, results from clustering demonstrate the impact of students' problem-solving behaviors on science, CT, and engineering learning. Depth-first approaches to computing and engineering seem to have a negative impact on science learning. This can be seen in Cluster One, and related to prior research (e.g., Zhang et al., 2021). Seemingly opposite, while high exploration may support learning in CT through the contextualization of difficult CT concepts, the approach seems to limit engineering learning as students do not systematically leverage the science and engineering practice of planning and carrying out investigations (as seen in Cluster Two). Finally, we see that translating productive computing skills to engineering design can be a complex task in need of additional instructional or scaffolding (for instance, with Cluster Three). These results support deeper investigations into using analytics approaches to evaluate learning and problem-solving behaviors using OELEs.

We recognize limitations in our research. In the future we aim to increase our student population to determine whether our statistical analyses and clustering results hold. Moreover, we aim to diversify our student populations and evaluate the impact of this "real world" approach on more students and determine if further customizations are needed to better engage students in this difficult curriculum. The feature selection process was also vulnerable to bias and we aim to use more robust selection processes, such as a Sparse Clustering method Witten and Tibshirani (2010). Finally, all groups identified through clustering indicated room for growth. Based on our performance and behavior results, we will target the development and analysis of instructional responses and formative feedback to help students develop good testing skills, targeting the key science and engineering practices discussed in our results.

CHAPTER 4

Manuscript Two: Co-Designing a Teacher Dashboard to Support Evidence-Based Instruction During Problem-Based Learning in Middle School Science

The modern classroom is increasingly changing. On the one hand, social and technological advancements have ushered in increased calls for designing inclusive educational opportunities that engage students in real-world, student-centered, inquiry-based Science, Technology, Engineering, and Mathematics (STEM) learning (NGSS, 2013; NRC, 2014). These approaches immerse students in open-ended, constructivist learning opportunities such as problem-based learning (PBL; Hmelo-Silver, 2004), that support technology-enhanced problem solving and knowledge construction (Freeman et al., 2014; Hutchins et al., 2020a). Simultaneously, advancements in artificial intelligence(AI)-based learning analytics approaches have increased our knowledge on student learning pathways, targeting students' individual needs and successes to allow for curricular adaptations (Graesser et al., 2012; Khribi et al., 2015). The combination of these directions poses a challenge - how might we leverage advancements in our understanding of students' technology-enhanced learning and problem solving to better support classroom teachers in implementing PBL in their K-12 classrooms?

Problem-based learning that incorporates real-world, student-centered, inquiry-based STEM learning has proven effective in engaging K-12 students in learning difficult science, computing, and engineering concepts and practices (Hmelo-Silver, 2004; Hsu et al., 2018; Zhang et al., 2022). However, limitations exist in the implementation of these approaches in K-12 classrooms. From the student perspective, students that do not have experience in these approaches struggle on tasks that require both knowledge and skill development in combination with developing self-regulation and maintaining motivation (Hmelo-Silver and Barrows, 2015). From the perspective of integrating science with domains such as computing and engineering, students struggle translating their developing science knowledge into computational form and/or to support engineering design (Sengupta et al., 2013; McElhaney et al., 2020) and applying systematic testing, debugging, and evaluation strategies to improve their solutions (e.g., computer models, engineering designs; Basu et al., 2016b; Zhang et al., 2021).

From the teacher perspective teachers must cultivate nuanced pedagogical practices designed to facilitate student-centered learning and provide in-time, evidence-based student support in line with the learning design (Gomoll et al., 2022). The difficulties of this process are exacerbated in technology-enhanced PBL as:

- student thinking is actualized through user-face interactions (Walkoe et al., 2017), and students often do not ask for or know they need help (Aleven et al., 2015),

- student-centered PBL approaches are linked to open-ended learning (c.f., Zhang et al., 2021; Biswas et al., 2005), and this openness implies a variety of approaches that students can take in their learning and problem-solving tasks, of which the teacher must be prepared to respond to when necessary, and

- little guidance is provided on how teachers may support students in meeting the expectations of integrating across disciplines to support PBL (e.g., NRC, 2014; NGSS, 2013), especially considering many science teachers do not have computing, pedagogical, and content knowledge for such learning designs (Bocconi et al., 2016; Cunningham and Carlsen, 2014).

As such, supporting teachers in the facilitation of PBL approaches in their classrooms is a research priority.

As computer-based environments are being developed to implement PBL curricula, we have been able to more deeply understand student-centered learning processes and the learning and problem-solving strategies they employ through advances in artificial intelligence and machine learning algorithms (Zhang et al., 2021). Specific to our work integrating problem-based learning opportunities integrating science, computing, and engineering, these advances leverage machine learning to identify productive and unproductive debugging strategies during computational modeling (Grover et al., 2016; Emara et al., 2021; Zhang et al., 2021) and systematic prototyping practices in engineering design (Vieira et al., 2016; Xing et al., 2021), evaluate student written responses through natural language processing (Bywater et al., 2019; Cochran et al., 2022), and group students based on learning and problem-solving behaviors to explore the nuances in students' strategies and how they impact learning through clustering analysis (Zhang et al., 2017, Manuscript One of this Dissertation). While these approaches have advanced our understanding of how students' learn and problem solve, more research must target the complex task of translating what we know as scientists and researchers into a language that classroom teachers can interpret and convert to actionable information (Wiley et al., 2020). While it is important to generate analytics that characterize different approaches students can take (e.g., by clustering or mining), it is equally important to help teachers customize the characterizations, so that they can convert them into actionable information - for class instruction, or to aid students individually or in small groups.

Integrating teacher insight into the design and development of instructional-support technology through co-design can help ensure alignment of that technology with teachers' needs, preferences, constraints, and goals of their practice (Matuk et al., 2016; Prieto et al., 2019). However, these approaches typically leverage insight from teachers experienced with the technology and for application of non-student-centered curriculum approaches (e.g., teacher integration of intelligent tutoring systems in their classroom; c.f., Holstein et al., 2019), and are designed to help students get "back on track" (Haklev et al., 2017) as opposed to supporting engagement in the disciplinary substance of students' ideas and problem-solving processes. Simultaneously, advances in learning analytics and data visualizations have demonstrated the positive impact of dashboards in supporting instructor facilitation of PBL (e.g., Chen et al., 2021); however, these efforts are still in infancy and co-design work has rarely focused on the development of instructional-support technology for the facilitation of K-12 PBL (Hutchins and Biswas, 2022). Integrating teacher insight into the creation of such technology for PBL applications is critical for (1) coming to a shared understanding about how responsive teaching practices are generated in face-to-face, PBL implementations (Gomoll et al., 2022), and in particular technology-enhanced approaches, and (2) increasing our understanding of the sensemaking processes inherent in moving from understanding and interpreting learning analytics feedback and developing evidence-based responses (Campos et al., 2021).

This work presents a novel, end-to-end demonstration of how to engage teachers in the development of learning analytics and visualization systems in the form of dashboards that support PBL curriculum approaches in K-12. In the process, we adapt established participatory design techniques and demonstrate new kinds of prototyping methods to address the unique challenges of co-designing teacher-support technology for PBL curriculum applications, specifically technology-enhanced, problem-based curricula integrating science, computing, and engineering. For example, this work addresses co-design issues, such as the need for stakeholder training and support for meaningful contributions (Cook-Sather, 2014) by integrating prototyping sessions with real student data into teacher professional development on the PBL curriculum to connect teachers' experience with and understanding of factors impacting the curriculum design in their classroom to prior student experience illustrated through data visualization.

We leverage conjecture mapping (Sandoval, 2014) to demonstrate how teachers adapt the design, and provide design narratives (Hoadley, 2002) to detail the co-construction, evaluation, and implementation of design decisions made, culminating in the implementation of the Responsive Instruction for STEM Education (RISE) teacher dashboard (Hutchins and Biswas, 2022). Finally, we share reflections on changes made to our dashboard tools, and how these co-design approaches helped inform technology refinements and innovation to support the teaching of PBL in science classrooms. In doing so, this research aims to contribute precedent knowledge (Oxman, 1994) of PBL instructor-supporting technology design recommendations and useful co-design procedures.

The organization of the paper is as follows. We first provide background on responsive teaching to facilitate technology-enhanced, PBL in K-12 science classrooms and how co-design has been used to support technological innovations, in particular, instructor support technology (Section 4.1). This includes descriptions of limitations in the existing literature that our co-design methods target. We then present our methods (Section 4.2), including the instructional context targeted by our instructor-support technology, design perspectives that frame our design process, background on our participants, a high-level overview of our co-design methods, and data collection and analysis procedures. This is followed by our design narratives (Section 4.3), detailing each co-design method and design decisions made, and overarching teacher feedback on designing instructor-support technology for PBL classroom implementations. Next, we provide illustrative examples of RISE's use through teaching simulation activities to demonstrate the usage of co-design modifications by teachers (Section 4.4). Finally we close with a reflections of our process and recommendations for future co-design efforts targeting PBL curriculum applications (Section 4.5), and concluding remarks (Section 4.6).

4.1 Literature Review

The goal of our technology development is to facilitate responsive teaching of PBL by providing co-designed instructor-support technology with learning analytics and visualizations that are *interpretable* and *actionable*. This section examines prior work in technology-supported responsive teaching and the co-design of such technology.

4.1.1 Responsive Teaching for Technology-Enhanced, Problem-Based Learning in Science

Research on teaching with technology has focused primarily on teachers' use of technology, including their competency with and beliefs about how and when to use technology in their classrooms (e.g., Mishra and Koehler, 2006). Recent work has explored curriculum and assessment design and modifications with technology-enhanced learning environments (Kali et al., 2015). However, limited research has explored technology-supported responsive teaching, or noticing, interpreting, and responding to the disciplinary substance of student thinking (Walkoe et al., 2017; Sherin et al., 2011), while students use technology and there has been limited impact of incorporating teacher noticing with the design and implementation of K-12 classroom technology (Walkoe et al., 2017).

Technology-enhanced learning environments provide new opportunities for students to engage in authentic science practices (e.g., NGSS, 2013) and generate explanations of their developing STEM knowledge (Bywater et al., 2019; Slotta and Linn, 2009). Students' interactions with the technology can afford a unique perspective into the progression of student knowledge and thinking as they engage with computing tools (Noss and Hoyles, 1996) and the type of thinking the students express using these tools are often those promoted by the recent state and national standards (Walkoe et al., 2017). For instance, we have leveraged student activity data during computational modeling to evaluate student debugging and data analysis strategies during computational modeling in science (Hutchins et al., 2019a; **?**).

Although these environments have the potential to support responsive teaching (Bywater et al., 2019), teacher noticing difficulties may be increased when students use technology-enhanced learning environments due to:

1. teachers' limited background in computing and teaching using technology (Bocconi et al., 2016),

2. the decreased visibility of student thinking, as it is now applied through mouse clicks and other user-interface interactions and, therefore, not easily or readily apparent to the teacher (an important feature of lesson design to support teacher noticing; e.g., National Council of Teachers of Mathematics, 2014), and

3. software constraints or user-interface difficulties that may impact teachers' abilities to adequately respond to student thinking or issues (Walkoe et al., 2017).

Finally, while these environments support key processes highlighted in state and national standards, these strategies are often not engaged in by teachers during instruction (Walkoe et al., 2017). This is particularly challenging for teaching through student-centered learning approaches such as PBL, as teachers must interpret and respond to student progress, represented through data visualizations on a dashboard, in ways that target learning and problem-solving needs while also maintaining the intent of the learning design (e.g., not always address a specific knowledge gap through direct instruction) (Chen et al., 2021). As such, more research and development is needed to leverage action data to provide teachers interpretable, actionable feedback to employ the benefits of effective teacher noticing and response (Wiley et al., 2020).

4.1.2 Co-Designing Instructional-Support Tools

Research in the learning sciences has continuously demonstrated the importance of integrating teacher insight into the design and development of curriculum materials, assessments, and instructional strategies (Reiser et al., 2000; Shrader et al., 2001; DiSalvo and DiSalvo, 2014; Könings et al., 2014). For this research, we focus on co-design, coined by Penuel et al. (2007) to describe the collaboration between researchers and teachers for the systematic design and construction of technology-enhanced educational innovations.

In co-design, researchers and teachers share a mutually beneficial partnership throughout the design process in which their roles are clearly defined and activities, generally situated outside of traditional work environments, combine individual expertise to support the building of technology that addresses shared educational goals. Co-design shares assumptions and philosophies with participatory design, in which stakeholders are actively involved in the design process from start to end (Muller et al., 1992), value-sensitive design, in which the adoption of the technology is dependent on the degree that the design reflects the users values and needs (Friedman et al., 2002), and scenario-based designs, which focus on the context of the technology implementation (Carrol, 1999), amongst others.

Co-design approaches have demonstrated significant benefits. These include: (1) supporting teacher and student learning (Penuel et al., 2007), (2) aligning educational goals and instructional strategies across multiple stakeholder perspectives (Barab and Luehmann, 2003), (3) creating unexpected innovations (Holstein et al., 2019), (4) empowering participants by giving them a voice in shaping technology that impacts their practice (DiSalvo et al., 2017), and (5) helping ensure sustainability by keeping materials relevant and usable (Barab and Squire, 2004; Blumenfeld et al., 2000). Moreover, researchers are provided enriched opportunities to learn from teacher experience to more clearly understand teaching activities, processes, and goals that can serve as the basis for defining technology requirements (Matuk et al., 2016), which is important as we develop technologies to facilitate the complex task of engaging in students' problem-based learning (Chen et al., 2021). In this research, we follows key co-design recommendations from the literature to leverage these benefits. This includes clearly defining the roles of researcher and teacher partners (Roschelle and Penuel, 2006) while also allowing for an adaptable co-design process that supports teacher engagement, ownership, and value in the design and outcomes (Cober et al., 2015).

However, challenges have limited the application of this approach, particularly in the creation of learning analytics and accompanying data visualizations (Holstein et al., 2019). Teachers ideas about the role of technology in their teaching is often unclear or inconsistent (Kirschner and van Merriënboer, 2013) and teachers may hold different perspectives on student learning (Penuel et al., 2007), which may be exacerbated by teachers' limited implementation of PBL in their classrooms (discussed above). Logistics issues include time constraints of teachers (Penuel et al., 2007) and the need for additional training or support to ensure meaningful contributions (Cook-Sather, 2014). Moreover, in past approaches, teachers were brought in to the design process at a late-stage, after educational goals impacting the technology requirements are already established (Rodríguez-Triana et al., 2018). Limited guidance is provided on end-to-end co-design, with the exception of Holstein et al. 2019 and Prieto et al. 2019. However, these works do not cover student-centered approaches such as problem-based learning curricular implementations and limited co-design research exists that supports responsive teaching in science (Matuk et al., 2016).

The meaningful contribution of teachers in shaping learning analytics technologies, including instructor-support technology, is a central open challenge in human-computer interaction research (Baumer, 2017). In order to bring PBL approaches into K-12 classrooms, we need to better support teachers in noticing and responding to the substance of students developing science, CT, and engineering ideas and the complexities in the variety problem-solving pathways students choose as they take control of their learning. Recent research has demonstrated the effectiveness of integrating teacher insight into the creation of educational technologies; however, co-design research supporting the creation of teacher-support technologies for PBL implementations is scarce. This research targets this deficiency in the literature by providing detail on the multi-step co-design process culminating in the creation of the Responsive Instruction for STEM Education.

4.2 Methods

4.2.1 Learning Context

This research centers on the application of the Science Projects Integrating Computing and Engineering (SPICE) curriculum, developed as a multi-year research project supported by groups from Vanderbilt University, SRI International, Digital Promise, University of Virginia, and Washington State University. SPICE is co-designed and developed through an interative, design-based process, systematically refining each component of SPICE based on research studies in Tennessee (2019) and in Virginia (2018 and 2020).

SPICE supports teachers in the implementation of the Water Runoff Challenge (Zhang et al., 2020; Chiu et al., 2019; McElhaney et al., 2020; Hutchins et al., 2021a). The Water Runoff Challenge (WRC) is a three-week, NGSS-aligned unit that challenges students to redesign their schoolyard using different surface materials to minimize the amount of water runoff after a storm, while adhering to a series of design constraints. These include the overall cost and accessibility, while providing for different functionalities for the schoolyard (Chiu et al., 2019). The problem-based learning curriculum consists of five core units, illustrated in Figure 5.1. These units include: physical experiments, conceptual modeling, paper-based computational thinking tasks, computational modeling of the water runoff phenomenon, and engineering design, in which students use their computational models to redesign their schoolyard. This learning context is authentic and relevant to students facing similar problems (limited usability and pollution) in their own schools, therefore, the WRC is potentially engaging and personally meaningful to the learners (McElhaney et al., 2020).

The WRC targets NGSS performance expectations for upper elementary and middle school Earth science and engineering design curricula, emphasizing the movement of surface water in a system after heavy rainfall and the human impact of this runoff on the environment. For detailed information about our curriculum, assessments, and learning environment, please see (Manuscript One of this Dissertation).

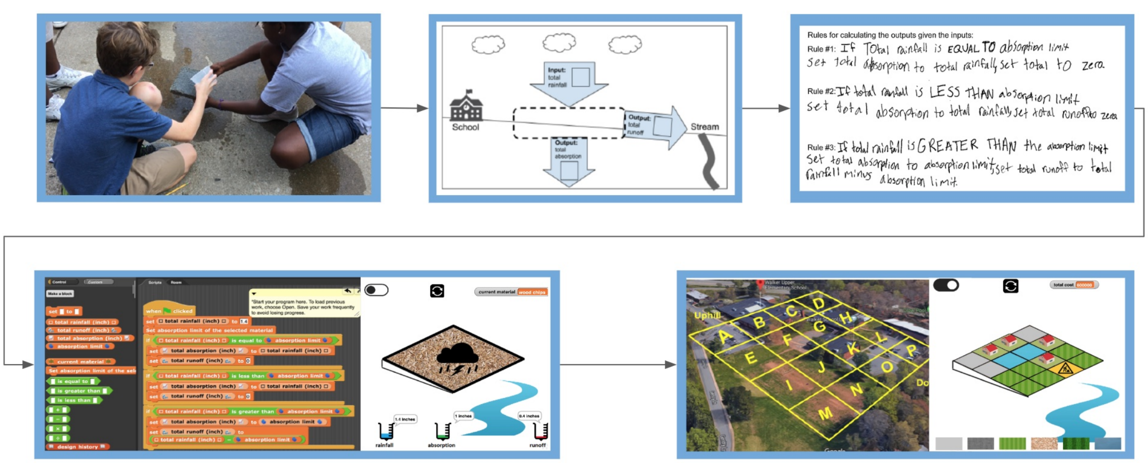


Figure 4.1: SPICE Curriculum Progression

4.2.2 Participants

Nine middle school STEM teachers (6 female, 3 male) participated in three design sessions. Due to varying availability, three teachers participated in all three sessions, one participated in two sessions, and the remaining five participated in one session. Each teacher consented to take part in the research. The teachers were from varying urban and rural locations, including Tennessee, Illinois, Virginia, New York, Wyoming, and the US Virgin Islands. Three teachers had prior SPICE implementation experience, one teacher had prior experience with C2STEM (the core learning environment; Hutchins et al., 2020a), and five teachers had no prior experience.

4.2.3 Design Perspectives

The complexities of problem-based learning approaches require designers of supporting technologies come to an understanding of teaching practices in the classroom (e.g., how teachers engage and support their students during problem-based learning designs) and out of the classroom (e.g., how teachers prepare to teach problem-based learning and the impact of grading and other evaluation processes have on evidence-based pedagogical adjustments) (Matuk et al., 2016). In addition, designers must develop a shared understanding with teachers on the impact technology has on those processes. Based on this background, and that provided in Section 4.1, we developed and implemented a codesign process to align our design requirements and tools with the values, needs, and concerns of the participating teachers. This process included coming to a shared understanding of what it means to support students' during student-centered, problem-based learning in SPICE and how teacher-support tools could enable teachers to enact that support.

Design narratives have been used in the learning sciences to systematically describe the methods, processes, and decisions made in the design of educational technology (Hoadley, 2002). The use of design narratives targets the documentation of design examples relevant to the co-design and development of educational technology, in particular that supportive of PBL curricula implementations. We also clarify that our approach is not intended to communicate empirical evidence or validation. We anticipate this work will support future research and practice for related design problems.

We adopt conjecture mapping as the approach to illustrate the design narratives (Sandoval, 2014; c.f., Lawrence et al., 2022). Conjecture maps have been used *a posteriori* to contextualize the relationship between theory, features of the learning design (embodiment), actions generated by those features (mediated processes) and outcomes (e.g., student learning). This process emphasizes the role of emergent behaviors leveraging the designed tools on learning outcomes, as opposed to the tools themselves (Sandoval, 2014) and this allows us to illustrate the impact of teacher insights during codesign on the design of key technology features. For example, insights from the codesign sessions target a deeper understanding of how teachers utilize the teacher-support technology in the context of a problem-based learning curriculum (design conjectures) to determine evidence-based responses that engage in student learning and problem-solving processes during SPICE, and how those responses support student learning in science, computing, and engineering (theoretical conjectures). Conjecture mapping involves (1) the development of a high-level conjecture, or a theoretical idea about the learning and context targeted, (2) defining the features of the learning design, including tools, tasks, participants, etc. (embodiment), (3) describing the mediated processes generated by the embodinement features (also known as design conjectures), and (4) describing the hypothesized interactions between the mediated processes and learning outcomes (theoretical conjectures).

Figure 4.2 shows our team's initial conjecture map (prior to the codesign sessions). The high-level conjecture framing this research is that evidence-based pedagogical responses targeting learning and problem-solving strategies, supported by a co-designed teacher dashboard, can lead to better learning for students during problem-based learning in middle school science. This research examines the co-design methodologies and technological tools that support the embodiment of this conjecture, the mediating processes supported, and a reflection of changes to our tools and design and theoretical conjectures throughout the co-design process.

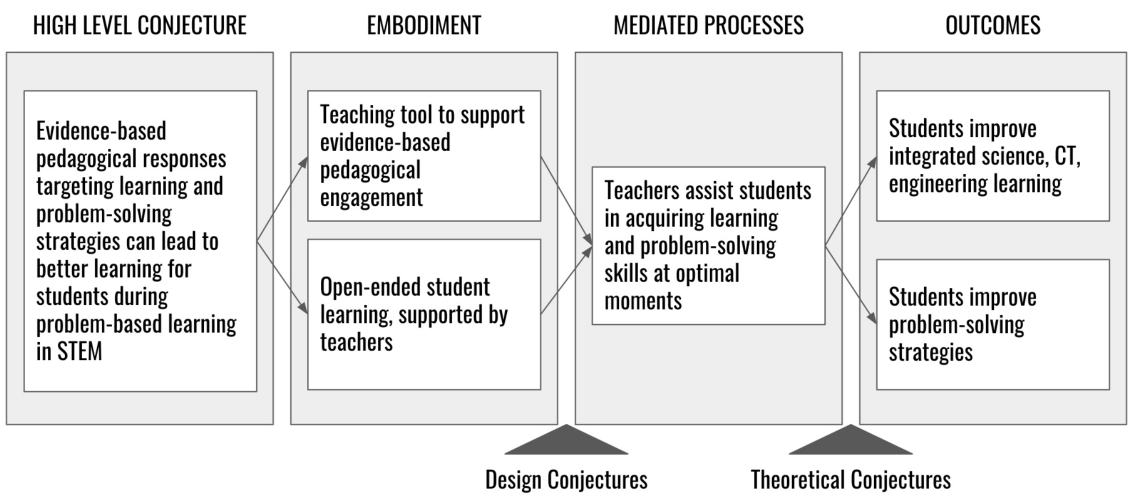


Figure 4.2: Conjecture map before the design process.

4.2.4 Overview of HCI Methods

At a high-level, our design process followed the LATUX workflow for designing and deploying learning analytics tools (Martinez-Maldonado et al., 2016). However, due to COVID, we have not yet completed a classroom implementation, but aim to do so in the near future. Our approach consisted of three core design phases: Needs Analysis, High-Fidelity, Prototype-Supported Professional Development, and Planning Period Simulations. As we will describe in the design narratives, choices of method used for each phase of the multi-step design process were made adaptively, based on (1) uncertainties present in learning analytics and accompanying visualizations, (2) teacher experience and background and methods needed to support meaningful contributions, and (3) logistic concerns (e.g., time constraints, COVID protocols). A brief overview of the design narratives describing the HCI methods used, the purpose of each case, and key insights from the implementation, is presented below and detailed further in Section 4.3.

Table 4.1: Summary of Co-Design Techniques Used

Phase	Use	Key Insights
Needs Analysis	For a deeper, detailed understanding of teachers' needs, concerns, and values using data visualizations to support PBL instruction For presenting new tools in which the details for visualization had not been concretely defined For understanding the impact of student learning and performance over multiple, linked representations	Supported teachers in imagining how the tools would support their pedagogical decision-making, and changes that may be needed Supported researchers in explaining complex learning analytics results to teachers Example student artifacts linked learning analytics to contrasting student cases to support discussions on noticing and interpreting results Linked results from multiple lessons and domains to support discussions on the impact of pedagogical decisions on future tasks
High-Fidelity, Prototype-Supported Professional Development	For refining and extending features of the co-designed dashboard in the context of teacher PBL training	Invited comparisons to teachers' prior implementation of PBL curricula and/or experience in implementing the SPICE tasks to reason about student problem solving, learning, and how to respond Supported the articulation of specific (versus abstract) design decisions based on interactions with the prototype's data visualizations
Planning Period Simulations	For identifying expert and novice PBL teachers' responsive teaching processes and comparing against design decisions	Provided contrasting cases of teachers' responsive teaching with RISE for researchers to evaluate technology effectiveness in addressing design decisions

In each of these approaches, we discuss how they were implemented, how different representations were used to open discussions, and how we grounded design decisions in evidence from the approach.

4.2.5 Data Collection

To support a thick description (Hoadley, 2002) of our design prototypes, tools, processes, and decisions, we used a range of data sources, mostly collected through video recordings as described in Table 4.2. All sessions were conducted virtually and recorded using a video conferencing platform. In total, we had approximately 35 hours of video data, which we transcribed using an online transcription service. After each design phase, members of the research team met to synthesize insights and make decisions about the next prototype phase.

Table 4.2: Data Collection by Design Session

Phase	*n*	Video	Observations	Prototypes	Meetings
Needs Analysis	5	2 4-hour videos, 2 1-hour videos	14 pages of notes	3 Journey Maps (Miro.com); 30 Google Slides	10 pages of notes; 20 slides
Hi-Fi Prototype	4	4 2-hour videos, 1 1-hour video	6 pages of notes	1 interactive prototype	15 slides; 1 Affinity Diagram; 4 pages of notes
Simulations	8	16 1-hour videos	22 pages of notes	1 interactive prototype (RISE)	10 slides; 14 pages of notes

4.2.6 Analysis

We analyzed session recordings using Affinity Diagramming, a technique leveraged in recent codesign research (c.f., Holstein et al., 2019; Lawrence et al., 2022), which allows for the clustering and re-clustering of discourse segments to identify themes (Martin and Hanington, 2012). An general overview of our analysis process is illustrated in Figure 4.3.

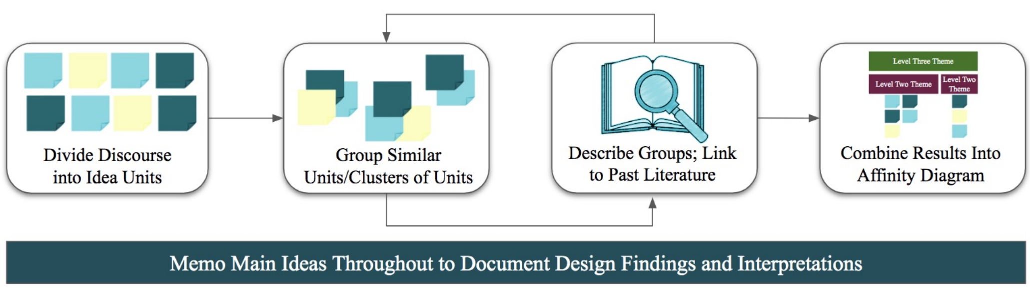


Figure 4.3: Process for analyzing discourse data from design sessions.

In order to accurately balance units of analysis, first divided the transcripts into smaller excerpts related to idea units (Jacobs and Morita, 2002), in which a single topic was discussed, and observation notes were paired based on time and relevance. Example idea units during Needs Analysis include, "*I mean, domain-specific results would be really useful for me, because then I could, because sometimes it's hard to know, for the kids to articulate what's hard for them. They'll just say, this is hard, or I'm confused, or I don't understand*" or "*For me, if I saw a bar graph or I saw graph and like 10 kids are right there, up front and it shows they* *didn't understand something, I can click on them and see who they were. Sometimes it's more stark if you see things in colors and visualizations than [just] the text itself.*" When all discourse was segmented, we created an Affinity Diagramming template on Miro and insert all idea units as post-it notes.

We then applied an iterative code development process by coding similar idea units and identifying major themes. This process was done by building on prior research on responsive teaching with technology. For instance, key themes identified in prior co-design research served as resources for coding, including *see student thought processes*, *detect student misconceptions*, *help me understand the why, not just the what*, and *engage in students' developing scientific ideas* (c.f., Holstein et al., 2019; Wiley et al., 2020). Agreement on the themes was established by internal meetings of the research team to make decisions about next steps in the prototype process. For instance, data visualizations were created to illustrate class, group, and individual performance in science and computing concepts and practices, targeting the excerpts above. Finally, we used memoing (Hatch, 2002) to summarize the main ideas of the design sessions in order to document our design process, key actionable and integrated teacher feedback, and support reflection on changes made to our design and theoretical conjectures. A portion of our final Affinity Diagram can be seen in Figure 4.4, with the idea units as pink and white post-it notes and blue labels identifying level-1 themes, yellow post-it notes representing level-2 themes, and white post-it notes (with a pink background) representing level-3 themes.

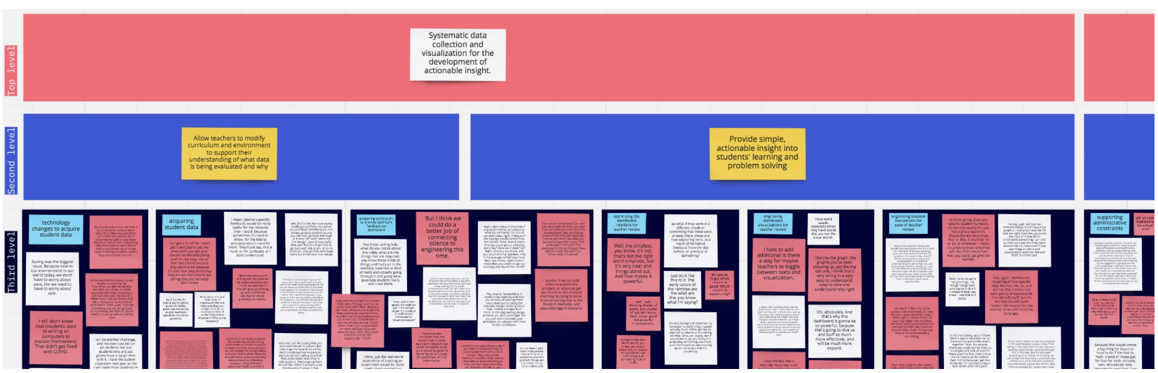


Figure 4.4: Conjecture map before the design process.

4.3 Design Narratives in Co-Designing a Teacher Tool

To integrate teacher insight into the design of RISE from start to end, we implemented a series of formative design studies with five teachers. Three teachers had previous experience with SPICE and two did not. The two inexperienced SPICE teachers were introduced to the computational modeling and engineering design curriculum prior to the co-design sessions, as we will describe below. For the purpose of these design narratives, we focus on feedback that directly associate with the conjecture map. In the final section, we provide an overview of all insights provided by the teachers and evaluated through the affinity diagramming. These insights may support future directions for research.

4.3.1 Needs Analysis

4.3.1.1 Session activity

The Needs Analysis phase leveraged key recommendations found in the literature from Section 4.1, including: (1) early-phase understanding of teachers' needs, values, concerns, and constraints in the context of the the target educational technology (Martinez-Maldonado et al., 2016; Holstein et al., 2019) and (2) using low-fidelity prototyping and physical artifacts that are more inviting of teachers' feedback and critiques as they convey a preliminary state (Matuk et al., 2016). This phase included a series of design activities adapted from established participatory design methods including card sorting, the love letter and the break up letter, and user journey maps (c.f., Martin and Hanington, 2012).

Understanding Experienced and Unexperienced Teachers' Needs, Values, and Concerns Regarding PBL Facilitation Prior to Prototype Development. The first step of the LATUX workflow (Martinez-Maldonado et al., 2016) is "Problem Identification." In addition to the literature review provided above (and in the Literature Review section of this dissertation), this step involved discussing and eliciting teacher insight regarding (1) how they perceive the integrating PBL, concerns they have, and preferences regarding classroom implementation, (2) what they need (e.g., educational technology requirements, curriculum and classroom resources, etc) to facilitate the implementation of PBL, and (3) what potential actions they may take to do support students if those needs are addressed. Distinct from prior work, this session needed to not only gain additional insight into how to conduct and visualize learning analytics to support the delivery of actionable insight, we needed to better understand middle school STEM teacher needs regarding implementing PBL in their classrooms (from both experienced and inexperienced teachers). Example artifacts from each session are provided in Figure 4.5.

First, in order to support a more free-flow of discussion regarding teachers' goals and concerns about teaching PBL, and specifically integrating computing and engineering into their science classrooms, we conducted an activity that avoided use of terminology such as "learning analytics," "artificial intelligence," and "technology support." In the first step, we adapted Holstein et al. 2019's Superpowers activity, a card sorting approach in which teachers describe and compare wanted superpowers for their job. In our approach, we:

- prompted teachers by asking, "If you were granted superpowers to better understand student learning and the successes and difficulties they experience during learning, what superpowers would you like to possess?",

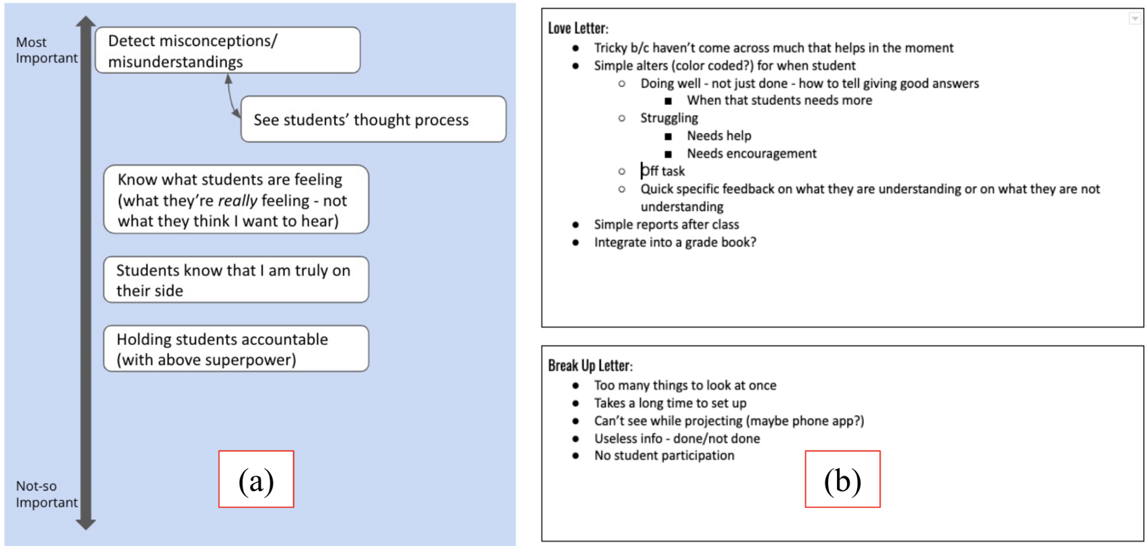


Figure 4.5: Example artifacts from the Superpowers and Love/Break-up Letter Activities.

- provided initial "superpower" cards based on results from Holstein et al. 2019's work to initiate discussions, and
- conducted the activity via shared Google Slides while using a video conferencing system (due to COVID protocols).

An example teacher's superpowers activity is shown in Figure 4.5(a). This approach allows us to view the problem at a high-level and determine what are example teacher priorities that we need to be sure to target in a teacher dashboard. For instance, in the example in Figure 4.5(a), this teacher focused on (1) identifying students' misunderstandings and the thought processes that may have led to those misunderstandings, (2) knowing what students are really feeling in the context of PBL (e.g., frustration, anxiety; this was commonly discussed in the context of the impact of the pandemic on student learning) and ensuring students' know that the teacher is on their side, and (3) holding students accountable, which is a difficulty identified in the PBL literature (e.g., (Hmelo-Silver and Barrows, 2015)). We also utilized results from this session as a key resource for UI visualization requirements. For instance, if determinations on what to eliminate from UI to prevent the display of too much data were needed, this resource reminds us of the key priorities of the teachers.

As a next step, following an initial presentation on SPICE (over video conferencing), teachers completed a written prompt in which they described a love letter and a breakup letter (Martin and Hanington, 2012) for using technology to evaluate students. In these letters, teachers described "what excites you and what you like about the availability of feedback on student learning processes and behaviors, including what it may help you do" and "what concerns you (or would cause you to "break up with" that technology), including what would prevent you from using that type of analysis during your classes." An example love letter and breakup letter for a participating teacher are shown in Figure 4.5(b). This activity extends our focus on teachers needs, values, and concerns regarding technology-enhanced PBL implementation by diving deeper into what would engage them in using educational technology, especially teacher-support technology, to support PBL in their classroom and what may cause them to stop using it (and potentially stop implementing PBL in general). To our knowledge, this is one of the first attempts at acquiring this detail regarding teacher insight for technology-enhanced PBL.

In this example, Figure 4.5(b), the teacher elected to use a bullet-point format. The teacher (an inexperienced SPICE teacher and not the same teacher as the Superpowers illustrated in Figure 4.5(a)) noted that they did not typically use dashboards, but wanted to be able to know when students are doing well in order to highlight their successes and to know when they may need an additional challenge and when students need help or encouragement (related to the experienced teachers' superpower about knowing how students are feeling). Technology-related, this teacher wanted simple reports after class and the ability to integrate the results into a grade book. Conversely, the teacher said that dashboards with too much information, that are slow to load, and contain information such as when students are finished or not finished would prevent them from using the technology. In addition, the teacher did not want to use the dashboard during class as they often project their computer screen and the teacher described the idea of student participation in the dashboard. We discussed this final idea further and the teacher noted that the dashboard should be viewable by students and allow for student engagement in their learning and the understanding of how they are doing and the behaviors identified. This idea was brought up by two other teachers.

Low-Fidelity Prototyping for Teacher Data Visualization Insights. The next step in the LATUX workflow includes low-fidelity prototyping. We were faced with unique issues in this process:

- problem-based, technology-enhanced learning approaches such as computational modeling can include a variety of data types and possible analyses to target student learning in multiple domains and to identify many different learning pathways students can employ and literature is scarce on what teachers need to know from this data to support evidence-based responses, and
- in our PBL curriculum, learning occurs through multiple, linked representations over 15 lessons and understanding student learning, what teachers need, and how teachers may respond may be impacted by representations that occurat different time-points in the curiculum.

In order to address these concerns, we adapted the concept of user journey maps to create a curriculum journey map, as partially seen in 4.6. Instead of a visualization of the experience people have at each step when using a product, these journey maps visualized the SPICE learning progression and were supported by three student examples per "moment" (or lesson) as well as accompanying class-level data visualizations.

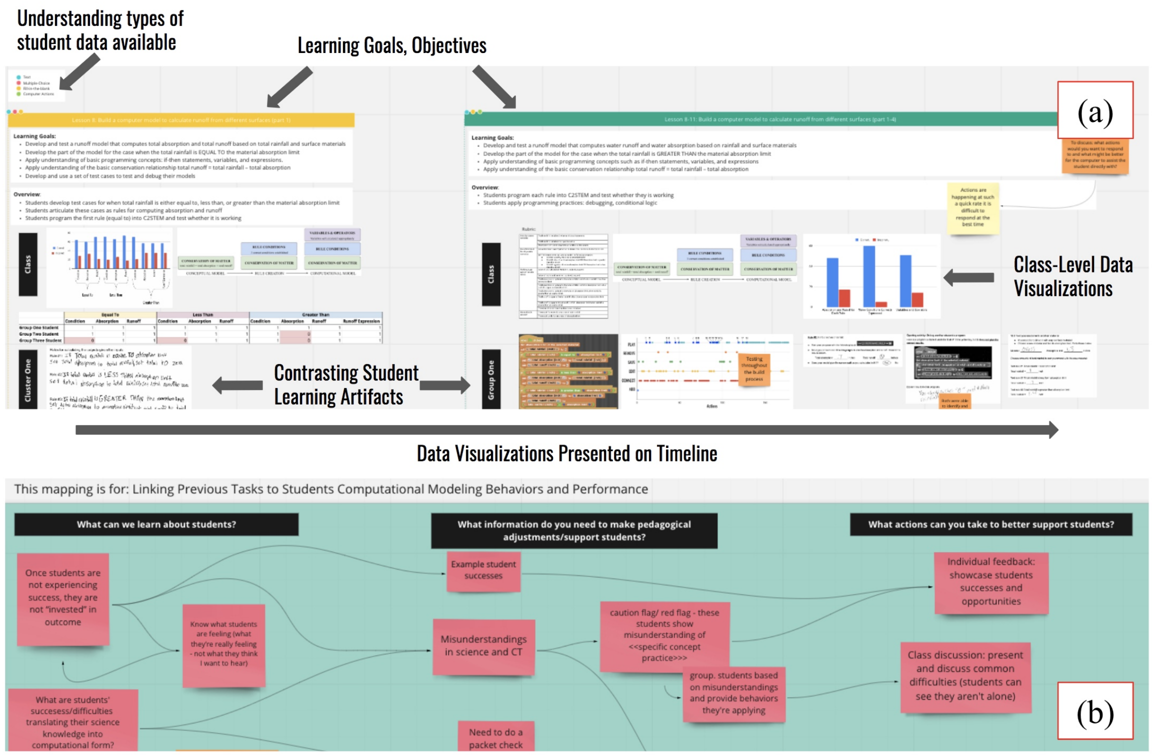


Figure 4.6: Curriculum Journey Map Components.

To begin, all teachers received a brief presentation over video conferencing (due to COVID protocols) about the SPICE curriculum. This included a review of the research team's prior experience implementing such curricula, such as learning performances in the integrated domains and example student difficulties. Teachers with no SPICE experience were also invited to complete the computational modeling activity and test design prototypes to gain user experience with the technology-enhanced learning environment.

Using the Miro software, we created a timeline of SPICE activities, shown in Figure 4.6(a). Each lesson included:

1. lesson objectives and goals (to orient each teacher),
2. details about the type of student artifacts available (to identify the type of student data available for analysis),
3. initial analysis of student learning (data visualizations to demonstrate initial dashboard feature ideas),
4. three example artifacts representing contrasting cases (to promote strong reactions by participating

teachers, see Matuk et al., 2016).

Goals of this visualization of the curriculum timeline included (1) mapping lesson objectives to examples of student work for in-depth discussion on quality of tasks and analysis, (2) linking multiple representations of student learning to discuss the impact of pedagogical approaches on learning over time (in prior research, we identified the importance of students' learning through multiple, linked representations to support integrated science, CT, and engineering learning; Manuscript One), and (3) supporting reflection of experienced teachers based on students results and inexperienced teachers based on their experience completing curriculum tasks (e.g., constructing the computational model).

In addition, a board was available throughout each lesson discussion (shown in green in 4.6(b)) that included our key leading discussion questions:

1. What can we learn about students?
2. What information do you need to support students? and
3. What actions can you take to better support students?

This board served as a shared note-taking tool in which teachers and researchers linked answers to those questions (input as post-it notes on the Miro board) to form a journey from what we can learn about students in the curriculum to potential evidence-based responses we might take. The curriculum journey map (and all of its elements) and the shared note-taking board served as boundary objects (**?**) to support researcher-teacher partnership discussions about responsive teaching in the curriculum and ways technology could support it.

4.3.1.2 Findings

Teachers validated our theoretical conjecture that assisting students in acquiring learning and problem-solving skills at relevant points in the curriculum will support improvements in domain-specific learning and problem solving. However, there were tensions regarding what constitutes supporting the "acquiring" of learning and problem solving skills and strategies as well as the importance of providing interactive tools that allow for the exploration of learning at the class, group, and individual level.

Regarding the mediated processes, teachers valued the performance feedback on the dashboard, but noted that they hoped their interventions would prompt students to engage in richer communication and application of their developing knowledge thus maintaining the student-centered approach to learning. For instance, one novice SPICE teacher noted, "*But I'm thinking how do you get the ones who didn't go beyond saying, okay, that's what [the teacher] just said [in class], you know, the guys that didn't really describe their thinking. The ones of you who did describe and put descriptions in about your thinking and about what your numbers* *mean, etc, etc. There's a strong correlation to that's gonna set you up for better success later. You know, and if you didn't, you want to really make sure you're trying to do that now, because it'll set you up for more success later.*" In this example, the teacher is focused on resources needed for them to best support students in communicating their learning and problem-solving skills in-the-moment to better prepare them for future applications of that knowledge. In addition, a teacher described "*This real time data on this would be really helpful for the rules...I know we often would go over the rules of the end [of the lesson] but it'll be really helpful if we could understand this [at multiple social levels]...The more conceptual knowledge they have during this process, I think the more that will translate over [to computing].*" In this example, the teacher described that if they had the results from the Rule Creation task earlier, they would be able to discuss the rules more based on student-specific responses so that difficulties could be better addressed and they could support student transitions to the computational modeling task. In doing so, teachers are supporting (1) applications of science developing knowledge, (2) linking multiple domains needed for the problem-solving task (science and computing), and (3) linking multiple representations of the same science phenomenon (water runoff described in the Rule Creation task and the computational model). In previous research, we identified these three activities as key for science, CT, and engineering learning (Manuscript One of this Dissertation).

Key design considerations were inserted into the preliminary love letter and break up letter task. For instance, in Figure 4.5b, the teacher noted that too much data visualizations and information such as who is done versus not done would have a negative effect on their engagement with the dashboard. In addition, this teacher (as well as two other teachers) emphasized the idea that they hoped the dashboards would be useful to their students as well if shown to them. All teachers described the benefit of after-class reports to support reflection and response processes. Other suggestions have been found in the literature, such as identifying students whom are off task (e.g., Holstein et al., 2019).

During the curriculum journey maps, teachers were provided with multiple examples of how data visualizations for student learning and problem-solving processes would appear and we received insight into dashboard user-interface preferences of the teachers. After all idea units were identified as user-interface recommendations, the team met to consider, reflect, and decide on designs for the initial high-fidelity prototype. For instance, as noted above, teachers wanted domain-specific performance information to narrow down the misunderstandings present that may have led to poorer performance or problem-solving struggles. In addition, all teachers requested data visualizations that grouped students based on common performance metrics. For example, a teacher answers a "what actions can you take to better support students?" with the response "*If 2/3 of class is demonstrating a misunderstanding, I will have class discussion.*" Although we do not know the substance of that class discussion, this pedagogical response may support students communication of their developing knowledge and problem-solving skills, which supports our PBL design goals and the mediated processes in our conjecture map. However, presenting student results at the class, group, and individual level may lead to the presentation of too much feedback. As such, the team decided on tools such as: (1) a grouping visualization that grouped students based on problem-solving processes in which teachers could interact with the visualization if they wanted more information about domain-specific performance and (2) presenting class-level feedback (e.g., one-third of the class has an error) as text and including both misunderstandings, successes, and problem-solving process applications so that teachers could reason about multiple viewpoints simultaneously. However, questions persisted regarding the design of key features, including how to test if a dashboard contains too much information and how to determine whether teachers were given sufficient information to promote actions that supported student-centered learning versus the more direct transfer of information.

Finally, one teacher brought up an anecdotal experience about prior dashboard usage noting that as dashboards typically give them too much information, the teacher often resorted to using physical post-it notes on everything they noticed and would stick them around the edge of their computer screen to remind them about things they needed to address the next day. This scenario inspired the development of a reflection tool that organized key insights, notes, and selected feedback to support a deeper reflection before deciding on any class adjustments.

4.3.2 High-Fidelity, Prototype-Supported Teacher Training

4.3.2.1 Session activity

The goal of the second design session was to support feedback for the refining and extending of features of the co-designed dashboard in the context of teacher through High-Fidelity prototyping (the next step in the LATUX framework). This work was inspired by Replay Enactments (Holstein et al., 2019) in the usage of real classroom data from prior SPICE implementations with the goal of simulating example situations in which teachers would review the dashboard, interpret the findings, and make in-the-moment decisions on next steps or evidence-based pedagogical responses. We conducted this session as part of a SPICE teacher training workshop. During these sessions, examples of which are described in (Chiu et al., 2021), teachers collaborate with researchers by going over curriculum lessons, discussing modifications to tasks, formative assessments, and instructional strategies. Teachers are also recommended to complete the computational modeling and engineering design activities as if they were students to support reflections on problems students may have. We paid careful attention to the literature regarding potential misalignment of co-design work with professional development (Boschman et al., 2014) to ensure meaningful contributions by the four participating teachers. Our reasoning behind integrating this design session with teacher training was to support contributions by teachers with no prior SPICE experience. An example view of the session is shown in Figure 4.7.

During the session, the lesson plan was introduced along with the overview of the lesson objectives, instructional strategy, and tasks. Teachers were then presented with the results from that lesson from a prior SPICE class. The research team then shared the high-fidelity RISE prototype with teachers with the goal of (1) simulating responsive teaching practices for each lesson and (2) targeting key design questions including a better understanding of how much is too much in terms of visualizations present and how to present information that supported facilitating PBL instruction and support. A think-aloud protocol (Martin and Hanington, 2012) was implemented, in which teachers were tasked with describing what they notice and their interpretation of the results based on the lesson objectives. Teachers then discussed what they might do both in terms of any potential changes needed prior to that lesson day or adjustments they would want to make in the next lesson. Researchers would ask probing questions as needed. In addition, as seen in Figure 4.7, teachers were shown three post-it notes at all times to help prompt collaborative discussion. Finally, in reflecting on potential pedagogical responses, teachers and researchers would reflect on the visualizations present and discuss design recommendations to aid in teachers noticing, interpreting, and responding process. These sessions were conducted and recorded over video conferencing software due to COVID protocols. Research team members tool observations to assist in the data analysis post-design session.

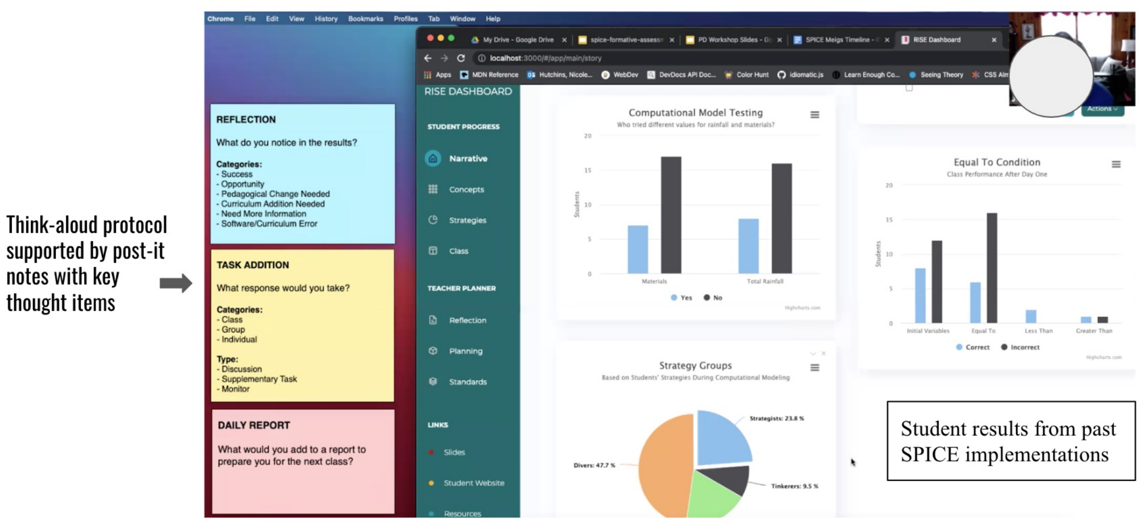


Figure 4.7: Example from the High-Fidelity, Prototype-Supported Teacher Training.

4.3.2.2 Findings

Findings from this design session impacted the tools and features (embodiment), design conjectures and theoretical conjectures. Teachers often asked clarifying questions regarding how data visualizations were created, linked, and the need for available information during class implementations to verify the accuracy of their interpretations. For instance, a teacher described: "*I'm absorbing what you're saying. And I'm suggesting to [researcher]...make these three sort of links. So if you click on one of these, so you're quite interested in seeing what happened in computational thinking. So if you click on that link, then a graph comes up with the computational thinking questions, and the concepts they're referred to, and you can actually see how they performed on individual questions. So you're connecting this graph to the more detailed graph, but But it's up to you to select what you want to look at in more detail.*" Researchers interpreted this idea unit in terms of changes to embodiment and design conjectures. For embodiment, we recognized the need for:

- additional interactive visualizations, and
- explainability in how data visualizations were created to not only support interpretation, but confidence in that interpretation.

These changes also impact resulting processes teachers implement. For instance, this example demonstrates the importance of giving teachers' agency on how much detail of student work they want to see and how they want it organized. These findings are similar to (Lawrence et al., 2022). However, it is also important to note that all teachers described that they would likely not review the complete set of feedback during a planning period. This impacted the design team, as further analysis and testing would be needed to provide the optimum amount of feedback to support responsive teaching and how to apply adaptivity in information presented to support teacher agency..

Moreover, teachers recognized that some errors may be better suited for immediate student feedback by the system due to, for example, potential careless errors by students. A novice teacher noted, "*What if there's some sort of automatic pop up alert for the kids? And this is for syntax, like, if it's, so that wouldn't be to me, it would be to students saying, like let's say they forgot to add a set button and they've tried [testing their program]and it's not working some sort of automatic pop up for the most common mistakes. Like, are you sure ... kind of thing? Like, can you double check that?*" In doing so, this teacher suggested that there may be some common mistakes, in this case during computational modeling, that a system can help provide in-the-moment response in ways that do not give students the correct answer, but force them to rethink and evaluate their code. For example, this teacher described that it might be difficult for them to find simple syntax errors as it would require reading through each line of code. In a classroom where multiple students need help, this may be too time-consuming to do.

Deeper discussions of the initial conjecture map continued in terms of the mediated process of assisting students, including the impact of understanding learning and problem solving at multiple social levels, and emphasizing that the goal was not to identify and correct errors, but to support students in building, applying, and communicating good learning and problem-solving behaviors so that they had agency in their learning.

For example, in describing potential responses, a teacher said, "*And I go back and forth with this, like, I don't know, if it's better to stick the kids who are all good with CT in a group and the kids who aren't like in smaller groups. I mean, it would be helpful for me to know, just so I can maybe mix it up a bit or if I made like, larger groups...to help each other.*" Researchers interpreted this idea from multiple perspectives in that it demonstrates a need for contextualized feedback at group and individual levels and, in this example pedagogical approach, would engage students in more opportunities to communicate their learning and problem solving as they collaborated with other students. However, it was positive to see pedagogical approaches more directed at PBL compared to the class discussion approach from the previous design session. We believe this is in line with the teacher learning benefits of codesign processes (Penuel et al., 2007). The research team met again following these design sessions to determine directions for (1) simplifying the user interface (too much feedback) and (2) improving interactions with key visualizations (e.g., allowing for additional information to be accessed from the visualizations that group students).

4.3.3 Updates to our Design Conjectures, RISE, and Overall Insights from Teacher Feedback

We made several adjustments to our conjecture map based on results of the co-design sessions (Figure 4.8 in *italics*). First, we refined the *embodiment* features, including tools and participant structures. From teacher feedback, the teachers discussed the need for the tool features including explanations of how feedback and visualizations were generated and tools to support systematic reflection and response processes. In terms of the participant structures, while teachers validated the importance of student agency in their learning, teachers also described that artificial intelligence features such as automatic, personalized feedback may be better at supporting students for in-the-moment needs that do not negatively impact the student-centered learning design.

Teacher feedback also allowed us to expand the mediated processes. In particular, they informed greater clarity in that the tool needed to help them develop responses that supported students in not only acquiring good learning and problem-solving behaviors, but also applying and communicating these skills (design conjectures). We made intentional modifications to RISE based on these insights, including updates to data visualizations to allow for teacher evaluation at multiple levels (e.g., the development of interactive data visualizations that grouped students based on problem-solving behaviors, but allowed for interactive exploration to learn more about domain knowledge deficiencies and successes individual students in each group demonstrated). While these modifications focused on how we could better support teachers, changes to our theoretical conjectures improved our hypotheses about how these mediated processes would improve student learning and problem-solving skill development. We extended this theoretical conjecture by elaborating that the implementation of those skills needed to support their linkages across science, computing, and engineer-

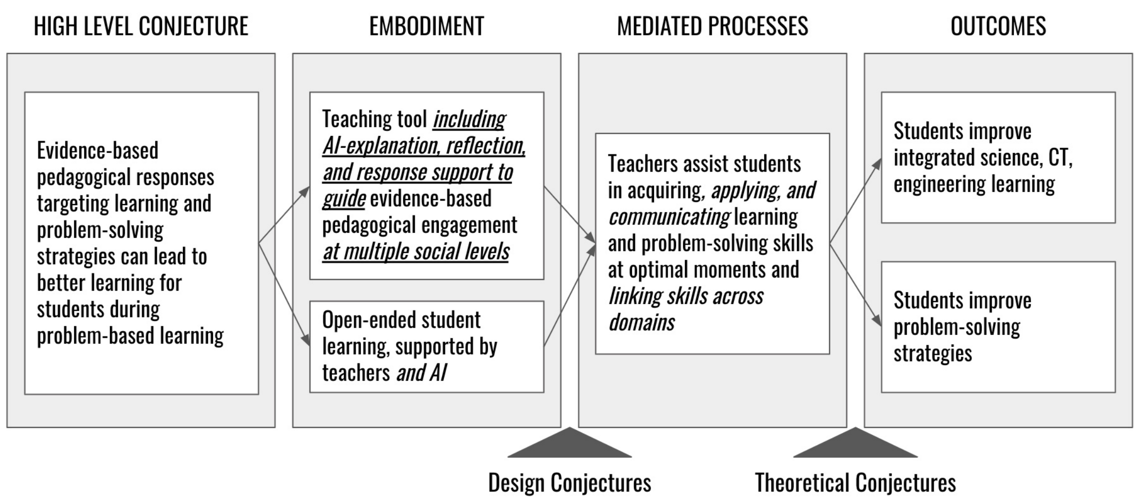


Figure 4.8: Conjectures produced following the co-design sessions.

ing. We liken this elaboration to work on preparation for future learning (Schwartz et al., 2005), the benefits of which we have seen in our previous computational modeling research (Hutchins et al., 2020a).

The final results of our Affinity Diagramming processes resulted in key insights for the design and development of teacher dashboard to support PBL. Our data processing resulted in 306 idea unit. These units were iteratively synthesized based on the processes described in Section 4.2, resulting in 18 level-1 themes, 6 level-2 themes, and 2 level-2 themes. Figure 4.9 details the top two level themes (level-2 and level-3).

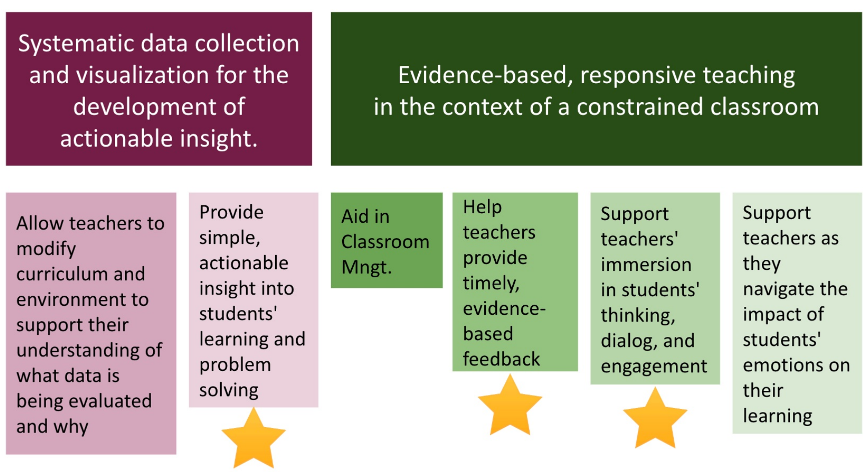


Figure 4.9: Affinity diagramming results from our co-design sessions.

Insights from the level-2 themes include:

- **Allow teachers to modify curriculum (including formative and summative assessments) and environment to support their understanding of what data is being evaluated and why:** feedback for this theme centered on integrating teacher feedback into technology and curriculum changes needed to support teachers' deeper understanding of the data being collected, why it is being collected, and how it is being analyzed. In SPICE, the initial curriculum was co-designed with participating teachers (outside of the scope of this paper); however, after reviewing and evaluating data visualizations over the course of the co-design sessions, teachers provided additional feedback on items such as adjusting instructional language to promote explanatory responses and scaffold some introductions to the potential implementation of key problem-solving processes. For example, related to a idea unit provided above, the teacher requested domain-specific insight. This may require updates to rubrics that identify applications of knowledge in different domains and in ways that can be acted on if successful or unsuccessful.

- **Provide simple, actionable insight into students' learning and problem solving:** This theme involved three key low-level themes, including (1) optimizing the dashboard interface for teacher review (e.g., including explanations of learning analytics to confirm teacher interpretations), (2) improving dashboard visualizations for teacher review (e.g., utilizing color schemes to denote successes and opportunities or allowing teachers to toggle between two types of visualizations depicting the same results to reinforce interpretations), and (3) organizing student evaluations for ease of teacher review (e.g., information should be simple and straightforward and detailed information such as a table of individual performance should be located on a different page from the landing and used by teacher preference or need for more detailed information).

- **Aid in classroom management:** This theme consisted of the greatest amount of low-level themes targeting traditional classroom constraints. These include supporting administrative constraints (e.g., including information about students that are absent), supporting timing constraints (e.g., an alert near the end of class to support the implementation and timeliness of exit tickets), allowing for lesson customization (e.g., dashboard should support ease-of-adjusting lesson plans based on decided evidence-based responses), grading (e.g., automatic scoring of assignments can focus teacher attention to lesson plan optimization), and supporting reflection (e.g., help organize what teachers notice to ensure they address key classroom needs).

- **Help teachers provide timely, evidence-based feedback:** feedback for this centered on providing tools for evidence-based responses and for providing in-the-moment automatic student support by the environment. Idea units for this theme include ideas for giving teachers agency in exploring and engaging in the data visualizations in ways that help them verify the accuracy of their interpretations, help them weigh potential evidence-based responses, and trust in their decisions. For instance, in an idea unit provided above, the teacher wanted to explore and interact with visualizations in their own way, so that they could better clarify what is impacting the group's problem-solving approach, while acknowledging that other teachers may find the original content and structure of the visualization to be sufficient. This theme also provided an opportunity for a future research agenda in which we could explore teacher-AI teams in which the learning technology provided in-the-moment feedback to students on teacher-decided topics, while giving agency to the teacher in deciding major class, group, and individual response needs.

- **Support teachers' immersion in students' thinking, dialog, and engagement:** feedback for this theme included engaging the teacher, providing a shared dashboard, and allowing teachers to integrate collaborative tasks. The majority of teachers described the dashboard as a collaborative tool and noted that they wanted to use the dashboard with students (e.g., the dashboard could show students that they improved from using difficult problem-solving processes to successfully debugging and testing their code, which may support their interest and engagement). In addition, all teachers noted that while they could think of ways to individually respond to students to support their problem solving, dashboard for PBL should also support the teacher in implementing simple collaborative tasks in which the teacher could then walk around the room and listen to how students communicate their knowledge and skills.

- **Support teachers as they navigate the impact of students' emotions on their learning:** A major theme across all design sessions was the impact of COVID on student learning, problem solving, and perseverance. Teachers wanted dashboard tools that helped them to encourage students as well as tools and resources to aid student anxiety during this complex curricular approach. We believe this is an important and rich research agenda.

In the following section, we have identified illustrative examples in which our dashboard led to actual teacher feedback recommendations labeled with a star in Figure 4.9.

Finally, these design sessions culminated in the RISE dashboard (Figure 4.10), in preparation for our Planning Period Simulations (to be discussed in the next section). Design tools created throughout and following these design sessions included interactive grouping visualizations, text-based feedback that highlights class, group, and individual successes and opportunities for improvement, curricular resources, reflection and response tools, and additional explanations of the artificial intelligence used resulting in the feedback presented.

An example interactive data visualization can be seen on the bottom-right corner of Figure 4.10. This visualization first groups students based on problem-solving strategies applied during computational modeling (e.g., students that implemented multiple depth-first construction sequences without testing their code or students that tested their code with multiple values of material and rainfall). Individual students are characterized by the shape of their circle to identify if they have completed their computational model or if they are in progress with no errors or contain errors. Teachers can explore deeper by hovering over students to identify if they had science or computing conceptual difficulties. We elected to not have additional modals or pop-ups as it would impact the usability of the visualization.

The reflection and response tools were added to support teachers' responsive teaching processes and were based on feedback themes described above. For example, in this initial implementation, teachers click on the green buttons available near each data visualization and a pop-up would appear (as seen in the "Reflection Form" on the right of Figure 4.10). Teachers can populate the form and select a category for the reflection. Teachers reflections are then stored by category and available for viewing on the Reflect page. On the reflect page, teachers can organize reflections based on importance. Tools to support reflections, including the curricular resources such as learning objectives for the day, rubrics utilized during analysis, and trajectory goals (e.g., what students are expected to know by the end of the day or unit) were also added through modals.

Finally, this initial implementation included modals composed of explanations for how the data visualizations were generated. For instance, the selection of text to use for "Successes" and "Opportunities" was based on rules. During the design sessions, teachers described that if two-thirds of a class demonstrated a particular misunderstanding or known difficulty, the teacher would likely do a class discussion or presentation the next day that focuses on that issue. As such, based on the analysis of student results text would be generated for any rubric item in which two-thirds of the class were either successful or needed additional support. This process was described similar to a decision-tree for that data visualization (e.g., Rokach and Maimon, 2005). Explanations of learning analytics used to support data visualizations for the Planning Period simulations can be found in our previous work (Manuscript One). All explanations were done manually by a research team member. However, future work can leverage recent advances of explainable AI (see Doran et al., 2017) and co-design for simple, automatic explanation generation.

4.4 From Thought to Action: Case Study

Utilizing the RISE dashboard (Figure 4.10) we conducted a series of Planning Period Simulations. These simulations were inspired by the Teacher Moments research at MIT (Benoit et al., 2021) and adapt the participatory design approach of simulations (Martin and Hanington, 2012). Planning Period Simulations were selected as 4 out of the 5 teachers that participated in the Needs Analysis co-design sessions preferred class reports at the end of the day (see Figure 4.5(b)) over dashboard usage in-the-moment or when class was in session (although teachers did show interest in simple alerts via phone or iPad in-the-moment).

During these simulations, eight teachers (three with prior SPICE experience, one with prior computa-

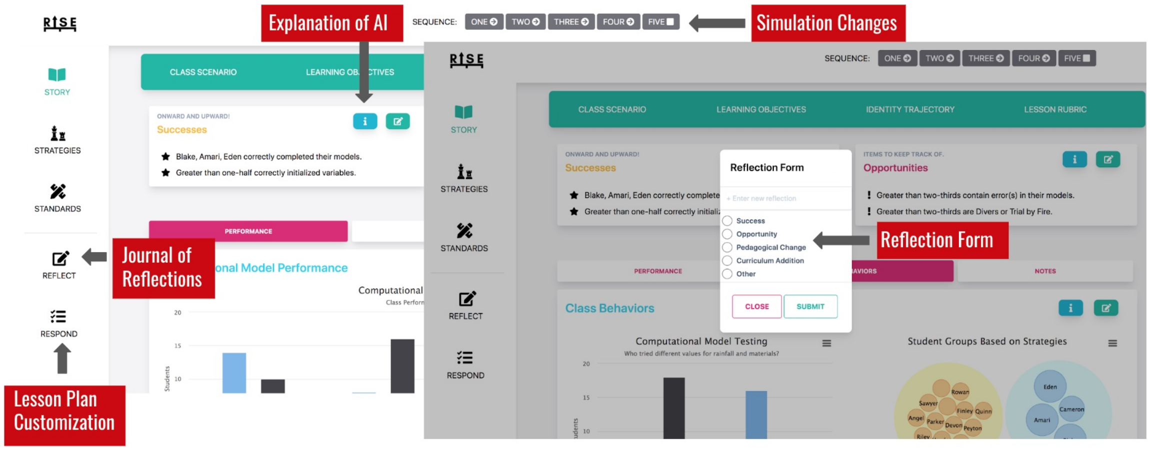


Figure 4.10: RISE dashboard following the co-design sessions.

tional modeling in science experience, and four with no experience) first completed a 90-minute professional development session in which they learned about or reviewed the SPICE curriculum and, for inexperienced teachers, they completed the computational modeling and engineering design tasks as students. Then the teachers completed a sequence of five simulations (as indicated in the top-middle of Figure 4.10), in which they would enact planning periods. For each simulation, a research team member first described the class scenario, including the class performance on the pretest, which targeted science, computing, and engineering knowledge, as well as other class results prior to the simulation "day." All information discussed was available on the dashboard. Teachers then had 15 minutes to review student results and feedback provided on the RISE dashboard, interpret what they saw, and customize class lesson plans for the next day, as though each class simulation was their class and they were evaluating what to do for the next day during their planning period.

A think-aloud protocol (Martin and Hanington, 2012) was implemented in which teachers described what they noticed, their interpretations of the feedback provided, ideas for responses, and reasoning behind their final class customization decisions. Data used for each simulation was pulled from prior SPICE implementations. This approach is similar to the Replay Enactment protocol implemented by Holstein et al. (2019). Student data from the prior implementations were de-identified and students were given gender-neutral names. A researcher was present at all times to answer questions and support evaluation processes as needed.

For the purpose and scope of this paper, we first identified illustrative examples of teachers (who participated in the co-design sessions) using tools developed or adjusted based on the feedback provided in those co-design sessions. We selected three examples that demonstrate applications of the three themes starred in Figure 4.9. The purpose of these examples is to demonstrate an application of a co-designed tool (or embodiment). As such, these examples are not intended to provide empirical evidence of the effectiveness of the tool.

4.4.1 Recommendation One: Support teachers' immersion in students' thinking, dialog, and engagement

The first example targets the feedback theme of immersing teachers in students' thinking, dialog, and engagement in the curriculum tasks. Specifically, we see a teacher's utilization of the strategy group visualization (described above). This simulation provided feedback for a class following the second day of computational modeling in SPICE. By the end of this day, students should have attempted all components of the computational model, including testing. A screencapture image from the video conferencing recording is provided in Figure 4.11. The image of the teacher and information saved in their browser have been hidden. Identification of tools used by the teacher are labeled in parentheses in the teacher quote provided. During a review of this simulation, the teacher thought aloud:

> *"Oh my goodness look at that, they're doing great...* (click to view page shown in Figure 4.11). *So behavior. Materials, they're doing a better job than other sections of testing out materials and rainfall. It's more evenly split for strategy groups...So trial by fire* (hovering over a strategy group named Trial by Fire and individual student in the group, Bellamy, as seen in Figure 4.11), *I think maybe they're not remembering any of their tests so for this class* (click to add reflection) *I would say for Asa, Drew, and Bellamy offer a paper table to record findings* (click to submit response). *And for the divers, I think they're doing great, but* (click to add reflection) *but have a smaller conversation with just those kids on how they can be more strategic."*

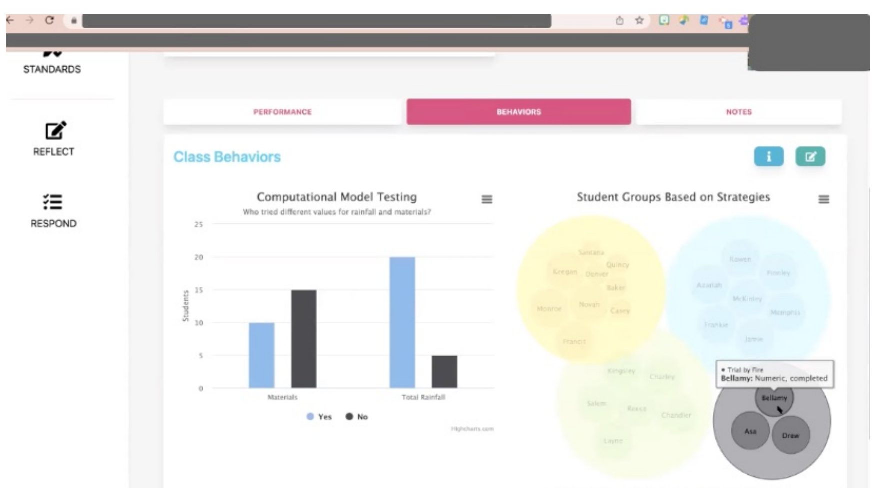


Figure 4.11: Example simulation targeting teachers' immersion in students' thinking, dialog, and engagement.

In this example, the teacher utilized the strategy group to interpret the performance of the Trial by Fire group, including hovering over individual students to notice their computational models were complete and their science was good (numeric indicates that students demonstrated success in calculating water runoff based on total rainfall and absorption limits during the science unit). Due to this interpretation, the teacher decided to develop an additional curriculum scaffolding to support a more systematic testing process for that group. The teacher then planned for a group discussion in which they would engage with the Divers, a group that consists of students that rely on depth-first approaches to constructing their computational models (and therefore do not test the model over time). The final customized lesson plan created by the teacher is shown in Figure 4.12.

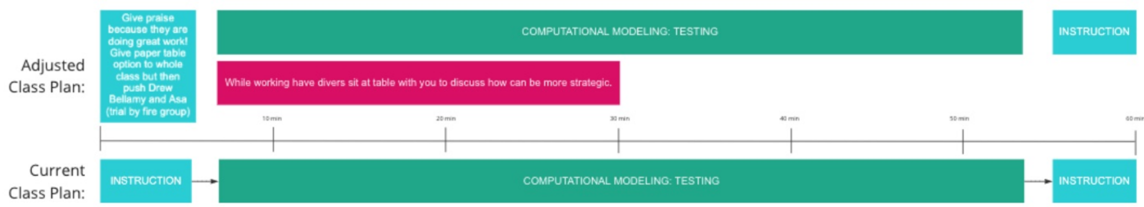


Figure 4.12: Final lesson plan created to engage in students' thinking and dialog.

4.4.2 Recommendation Two: Help teachers provide timely, evidence-based feedback

This example provides a demonstration of how a teacher leveraged information available from the dashboard to customize their lesson in a way that reinforced prior science concepts and practices reviewed to support the translation of that developing science knowledge into computational form (thereby also illustrating an application of our theoretical conjecture change described previously). In this teacher example the simulation provides results from a class after the first day of the computational modeling task, in which students were only tasked with completing variable initialization tasks and the first conditional logic statement. Similar to above, the screencapture image from the video recording software is provided, with identifiable information hidden, in Figure 4.13.

> *"So yeah, I guess knowing the low science pre-test score, I would say because they did not so great a job here* (hovering over initial variables bar graph, shown in Figure 4.13) *but they did a better job here* (hovering over equal to condition bar graph) *so they are getting this computing stuff better.*
>
> *So I would say* (clicks to add reflection) *add in a quick physical demo showing difference between absorption for sponge and paper, so they can understand why they need to initialize rainfall and*

absorption limit. So let me put in that."

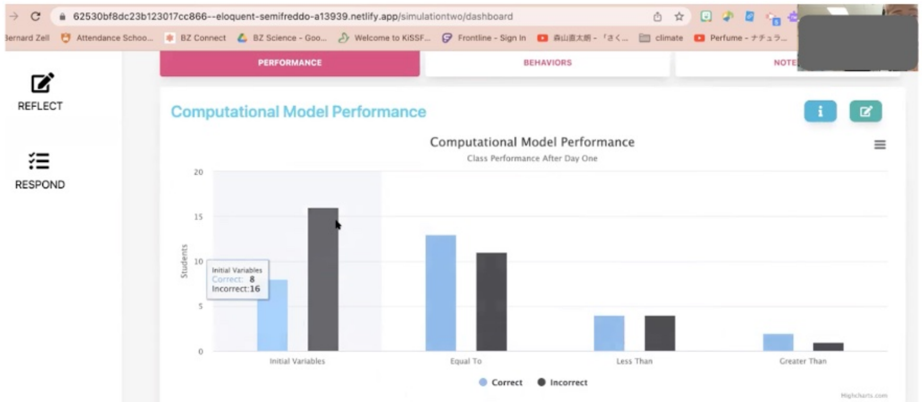


Figure 4.13: Example simulation targeting teachers' timely, evidence-based feedback.

In this example, teachers use feedback generated from the scoring of the computational models and links the results to feedback about prior science performance. Teachers are provided that rubric used to score the computational models (this teacher previously reviewed that rubric), which indicates that variable initialization issues may be due to difficulties translating science knowledge to a computational form as students must remember to initialize total rainfall and the absorption limit (needed for the computer to calculate water and total absorption). Information provided from the dashboard allowed the teacher to identify a domain issue early on in the transition to computing. The teacher then elected to do a science demonstration to reinforce science concepts and link them to the creation of the students' computational models. The teachers final lesson plan can be seen in Figure 4.14.

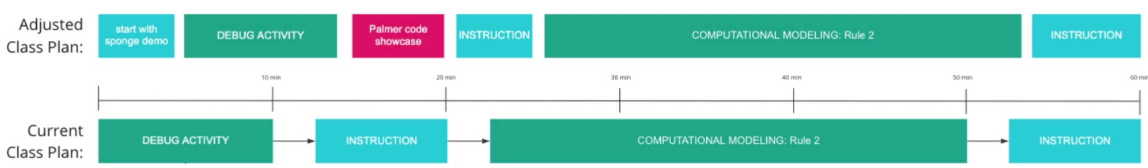


Figure 4.14: Final lesson plan created to provide timely, evidence-based feedback.

4.4.3 Recommendation Three: Provide simple, actionable insight into students' learning and problem solving

This example does not provide a screencapture as it involves a sequence of dashboard moves to multiple pages (identified in parentheses in the think-aloud quote). This example demonstrates the impact of changes in the embodiment component of our conjecture map, specifically, the need for insight at multiple social levels. The class data provided in this simulation was from the second-to-last day of the SPICE computational modeling unit.

"(viewing Story Page, Behavior tab) *Why are they still not testing materials? OK. . . So I'm going to pretend I see them all the time and I know which kids moved from here* (hovering over Divers in strategy group visualization, shown in Figure 4.11) *to here* (hovering over Strategists in the figure). *So* (clicks to add reflection) *Rowan to highlight their code and talk through strategy. So I would definitely want to do that as a way to give the kid props for moving groups.*

So I need to see the kids who are developing still. OK, going to the Standards (page click) *because it's the last day now. So at this point,* (click to add reflection) *Quinn and Jordan are partners and I am going to them to help*

And then I'm going to go back to here (clicks to view Story page, Behaviors tab). *I still have Ryan, Angel, Hayden, Riley, Devon all in same strategy group. So I'm going to say I'm going to mix up partners so my Divers and Trial by Fire are with Strategists and Tinkerers"*

In this example, the teacher first noticed that students were still not testing their model with different material options (e.g., different materials that could be used in the playground design). The teacher considered the impact of a student moving from an unproductive strategy group (such as a Diver) to a productive group (such as a Strategist). This teacher recommended this type of visualization as they were concerned that in their classes, they typically called on students they knew were good at coding and it felt as though that strategy was demoralizing for students that may not have a lot of computing experience but were trying. As such, the teacher elected to first do a class demonstration with that student.

The teacher then returned to the dashboard to focus on students that continued to struggle in science (categorized as "developing" to represent their science knowledge as developing; see Hutchins et al., 2021a). The teacher viewed the Standards page, which provides detailed, individual student feedback and identified the two students that were still struggling in science and paired them for the final day. It is interesting to note that the teacher acknowledged going to the more detailed Standards page "*because it's the last day*." This may indicate that certain detail is more important based on the curriculum timeline.

Finally, the teacher went back to the main page and to the visual feedback on student strategies and identified the remaining Divers group (described above). To support problem-solving processes, the teacher decided to pair remaining unproductive strategy groups (Divers and Trial by Fire) with productive strategy groups (Strategists and Tinkerers). We presume this approach would allow for students to learn from eachother and identify productive strategies on their own, while also allowing the teacher an opportunity to walk around the class and observe student pair discussions. The teacher's final lesson plan is in Figure 4.15.

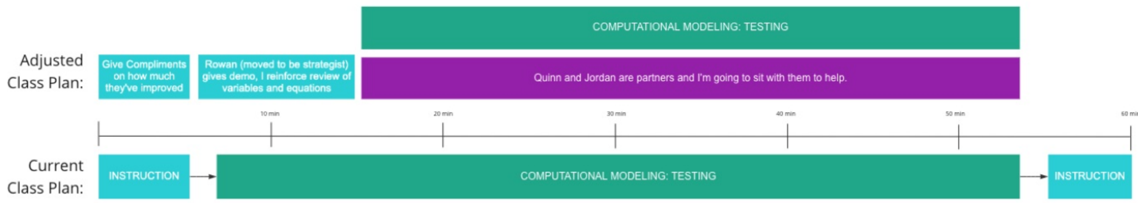


Figure 4.15: Final lesson plan created to engage in students' thinking and dialog.

4.5 Insights for Co-Designing Teacher Support Technology for PBL Instruction

The importance of co-design research continues to be highlighted in the learning sciences and learning analytics research community. In terms of the creation of educational technology, this literature has demonstrated the importance of teacher input at every stage of the design process - including prior to the development of low-fidelity prototypes (Martinez-Maldonado et al., 2016). These approaches are particularly relevant to support the implementation of technology-enhanced, problem-based learning in K-12 science classrooms as teachers must grapple with how to support student-centered learning as they construct knowledge in multiple domains (e.g., science and computing) and how to understand and engage in the variety of problem-solving strategies and pathways students may implement to get to a solution. However, end-to-end design demonstrations are still rare (Holstein et al., 2019), especially for technology that supports problem-based learning (PBL) in K-12 science.

In this paper, we demonstrated a novel approach to co-design that integrated experienced and inexperienced teacher insight in the design and development of teaching-support technology that improves responsive teaching during PBL. Our methods centered on eliciting teacher insight on engaging in students integrated learning of multiple domains, supporting students in learning through multiple, linked representations over time, and promoting productive problem-solving strategies to support that learning. In addition, we identified rich research agendas for (1) evaluating teacher-AI teams for more complete, timely support of students during PBL, (2) for using instructor-support technology to help teachers support students' emotional needs, especially following the impact of the pandemic on students' mental health, and (3) for advancing explainable AI research by leveraging teacher insights to build trust and support action in the learning analytics used.

On the one hand, we followed a broadly applied workflow for co-design (LATUX) and our findings are similar to prior approaches, including the benefits of beginning with stakeholder needs, *prototyping user tasks and usage scenarios early and often*, *using real-world data sets to in prototyping* (Holstein et al., 2019). However, our adaptations to this process that directly supported our goal of supporting teachers during PBL (and therefore may be supportive of co-design under similar contexts) include:

- **Regularly link student results across multiple, linked representations:** Problem-based learning often requires the application of knowledge and skills from multiple domains and highlights the importance of NGSS cross-cutting concepts. Teachers should be actively involved in the creation of the representations so that they better understand the linkages and the data used to evaluate students. In addition, regularly linking representations allows for critical reflection of the impact of pedagogical responses on student learning over the course of the PBL curriculum.

- **Immerse teachers, especially novice teachers, in the student experience prior to promote rich insight into visualizing student problem-solving processes:** In this research, we integrated the high-fidelity prototyping session into our SPICE professional development with the goal of receiving feedback from novice teachers (who had not previously implemented SPICE). In doing so, teachers had recent knowledge of their own problem-solving difficulties and could better connect to the real-world student results.

- **Regularly reflect on instructional strategies at different social levels:** An important update to our conjecture map was the need for teachers to understand student performance at different social levels (e.g., class, group, individual), so that they could systematically weigh potential class adjustments at those different levels. This process not only supported our understanding of considerations teachers make in deciding the type of responses to take (e.g., conducting a class discussion if two-thirds of the class demonstrate an issue), but it also helped us refine data visualizations that allowed teachers to explore student learning from multiple perspectives.

Finally, in this work, low-fidelity prototypes, student artifacts, curriculum journey maps, and high-fidelity prototypes all served as effective boundary objects to support researcher-teacher discussion, negotiation, and decisions on how to design and implement interpretable, actionable insight into student learning during PBL.

4.6 Conclusions and Future Implications

This work demonstrates a complete co-design process (with evidence of actionable insight from Planning Period Simulations) for the design and development of teacher-support technology targeting the implementation of problem-based learning in K-12 science. In this work we have demonstrated the importance for teachers (experienced and inexperienced) to be actively engaged in the co-design of AI-based teacher tools to support understanding students learning processes and translating them into actionable instructional moves. We have extended established participatory design techniques in order to effectively acquire teacher feedback in this context. In addition, we have provided the first, to our knowledge, design process that investigated teacher needs in the integration of problem-based learning in K-12 science (with applicability in other domains).

Limitations of our work include the small participation numbers. As such, this work provides depth over breadth in the demonstration of our co-design process. In addition, due to the impact of the pandemic, we have been unable to conduct a classroom experiment, thereby completing the LATUX workflow (Martinez-Maldonado et al., 2016). We aim to conduct this study in the near future.

CHAPTER 5

Manuscript Three: Using Teacher Dashboards to Customize Lesson Plans for a Problem-Based, Middle School STEM Curriculum

5.1 Introduction

Prior research has demonstrated the importance of teacher engagement in students' developing ideas and strategies to support their STEM learning. In applications of student-centered learning approaches, such as problem-based learning (PBL), this engagement poses challenges as teachers must interpret and respond to student progress in ways that target learning and problem-solving needs while also maintaining the intent of the learning design (e.g., not always address a specific knowledge gap through direct instruction; Chen et al., 2021). Technology-enhanced approaches can mitigate these challenges by visualizing student learning and problem-solving behaviors to support teachers using orchestration technologies such as teacher dashboards (Matuk and Linn, 2015). However, little research has targeted (1) dashboard-supported responsive teaching (Walkoe et al., 2017) and (2) processes that middle school science teachers use to bridge the noticing and understanding of AI-based instructional support with the determination of an evidence-based pedagogical response (Campos et al., 2021).

Understanding how teachers use dashboards to support evidence-based teaching practices during technology-enhanced curricula is critical for improving teacher support and preparation and serves as the context for this research. Through a systematic co-design process with expert (prior experience with the learning environment) and novice (no prior experience with the learning environment) teachers, we have created the Responsive Instruction for STEM Education (RISE) dashboard (Manuscript Two; Hutchins and Biswas, 2022) to support the implementation of a technology-enhanced, PBL curriculum known as Science Projects Integrating Computing and Engineering (SPICE; Manuscript One). The goals of the RISE dashboard are to support teachers in:

- noticing and responding to students' learning successes and opportunities (e.g., misunderstandings),
- facilitating student integrated learning of science, computational thinking (CT), and engineering across multiple, linked representations,
- aiding student-centered development of productive problem-solving strategies, and
- promoting student communication and application of their developing integrated knowledge through class and group discourse and problem solving.

In our prior research, the impact of learning through multiple, linked representations (Hutchins et al., 2021a; Manuscript One), productive problem-solving strategies (Zhang et al., 2021), and collaborative, open-ended problem solving (Emara et al., 2020) have proven to facilitate learning in our technology-enhanced, PBL approach. However, more research must target the complex task of translating what we know as scientists and researchers into a language that classroom teachers can interpret and convert to actionable information (Wiley et al., 2020). In this first step, we aim to evaluate the strength of RISE in supporting teachers' application of those PBL pedagogical processes.

This study examined eight teachers' use of a RISE to assess and respond to students' learning and strategies during SPICE. Teachers completed a series of 5 "Planning Period Simulations" leveraging the dashboard. Think-aloud protocols were implemented, supported by semi-structured interview questions, to enable the teachers to verbalize their thought and evaluation processes. Our analyses focused on the following research questions:

1. **How do expert and novice teachers implement responsive teaching to customize lesson plans using RISE?**

   To answer this question, we first conduct statistical analysis on the coding of expert and novice teachers' simulation discourse to identify the types of student work (e.g., performance scores, strategies applied) teachers notice and how they respond (e.g., teacher lectures, class discussions, group activities). Codes were developed based on prior work in responsive teaching (c.f., (Johnson and Forsythe, 2015; Chen et al., 2021)). We then conduct epistemic network analysis (ENA; Csanadi et al., 2018) evaluating the temporal discourse patterns expert and novice teacher implement as they complete each planning period simulation. We compare the networks and provide initial findings based on the results.

2. **What processes are involved in teachers determination of lesson plan customizations?**

   We conduct inductive analysis and constant-comparative analysis (Charmaz, 2006) to provide initial, exploratory patterns in the reasoning processes teachers implement in the transition from developing interpretations of student results to selecting evidence-based pedagogical responses based on those interpretations. We provide comparative case examples to illustrate identified processes.

In this paper, we first provide background on technology-supported responsive teaching as well as an overview of research targeting teachers usage of dashboards to support their practice. We then describe our instructor-support techonology known as the Responsive Teaching for STEM Education (RISE) dashboard and outline the co-design procedures taken to systmatically design and develop this tool. Next, we provide our methods, including the instructional context, our procedures for implementing the planning period simulations, our participants, and the data collection and analyses processes. Our results are organized by research questions in the next section and we conclude with a discussion of the results, limitations of our work, and future directions.

5.2 Background and Related Work

This work targets the novel exploration of teachers' responsive teaching practices as they leverage a co-designed dashboard to evaluate student learning and problem solving, and develop evidence-based lesson plan customizations as needed.

5.2.1 Responsive Teaching for Technology-Enhanced, Problem-Based Learning in Science

Science and math education reform has led to the promotion of fluid classroom environments that allow for pedagogical adjustments during instruction (van Es and Sherin, 2002). This pedagogical decision-making paradigm leverages responsive teaching in which the teacher makes in-the-moment pedagogical decisions based on what and how students are thinking, assessed through what students are saying or doing (Bywater et al., 2019; Wendell, 2016; Hammer et al., 2012).

This responsive approach is in contrast to traditional methods, in which lesson plans are predetermined and direct students' "flow of thought" (Hammer et al., 2012, p.54). This predetermined, traditional approach limits student opportunities to develop and assess their own ideas, which is needed for inquiry learning (Jiménez-Aleixandre et al., 2000) and open-ended learning approaches that include learning-by-modeling (Wilkerson-Jerde et al., 2015) and learning-by-design (Bywater et al., 2021; Watkins et al., 2018), such as that targeted in this proposed research.

Attending and responding to the disciplinary substance of student ideas is considered a core teaching practice in science, math, and engineering (NRC, 2007; Levin et al., 2009; Lampert et al., 2013; Coffey et al., 2011; Johnson et al., 2017). Responding to student ideas as they unfold in class has proven to:

1. help students engage in science practices (Hammer et al., 2012; Coffey et al., 2011),
2. focus student attention on the disciplinary substance of their thought (Warren and Rosebery, 1995), and
3. improve students' conceptual understandings (e.g., Robertson et al., 2016; Empson and Jacobs, 2008).

This process is akin to formative feedback, providing students information to support adjustments in their thinking, guide them towards the desired learning goals, and improve knowledge development (Shute, 2008; Bransford et al., 2000).

However, Van Es and Sherin note that successful applications of responsive teaching requires teachers to develop new ways to engage in and interpret classroom interactions (2002). The complex, challenging practice of responding to student ideas requires that teachers consider and evaluate copious amounts of classroom information (e.g., student discourse, performance) as well as the intrinsic and extrinsic constraints of the classroom environment (e.g., learning standards and objectives, time, assessment needs), and make in-the-moment decisions on what and how to engage in their students' ideas (Bywater et al., 2019; Sherin, 2002; van Es and Sherin, 2002).

The complexity of this practice can be exacerbated during technology-enhanced, problem-based learning due to:

1. teachers' limited background in computing and teaching using technology (Bocconi et al., 2016),
2. the decreased visibility of student thinking, as it is now applied through mouse clicks and other user-interface interactions and, therefore, not easily or readily apparent to the teacher (an important feature of lesson design to support teacher noticing; e.g., National Council of Teachers of Mathematics, 2014),
3. problem-based learning is akin to open-ended learning, in which students may implement a variety of problem-solving approaches during solution construction (Walkoe et al., 2017; Zhang et al., 2021) that teachers must grapple with and engage in, and
4. software constraints or user-interface difficulties that may impact teachers' abilities to adequately respond to student thinking or issues (Walkoe et al., 2017).

Finally, while these environments support key processes highlighted in state and national standards, these strategies are often not engaged in by teachers during instruction (Walkoe et al., 2017). This is particularly challenging for teaching through student-centered learning approaches such as PBL, as teachers must interpret and respond to student progress, represented through data visualizations on a dashboard, in ways that target learning and problem-solving needs while also maintaining the intent of the learning design (Chen et al., 2021).

These experiences motivate a deeper understanding of what it means to notice student thinking during technology-enhanced, problem-based learning and the processes teachers take in the transition from their interpretation of student learning and problem solving to the creation of evidence-based pedagogical responses supportive of the problem-based, student-centered learning design.

5.2.2 From Instruction to Action

Learning analytics research has progressed significantly and has led to the development of instructor support-technology proven effective for teaching with intelligent tutoring systems, collaborative learning scripts, and much more (please see Dissertation Literature review). However, research on teachers' usage of instructor-support technologies such as dashboards is still scarce, especially for the implementation of problem-based learning curricula (Chen et al., 2021).

A careful analysis of prior research models representing dashboard-supported responsive teaching results in the identification of key research opportunities and directions involved in the understanding of how teachers use dashboards, and how to support them. The first research area targets the transition from educational event to the visualization of student results on a dashboard. Research in the area has targeted the co-design methods for integrating teacher insight into the presentation of such visualizations (Wiley et al., 2020), improving transparency in algorithm development (Holstein et al., 2019), and supporting teacher agency in the representations shown on dashboards (Ahn et al., 2021). In our research, we have implemented a multi-step co-design process for the creation of the Responsive Instruction for STEM Education dashboard (Manuscript Two, and below).

Another research opportunity involves a deeper understanding of how teachers make sense of the information provided on dashboards. Recently, Campos et al. conducted a study with teachers and educational coaches to examine this sensemaking process and developed a typology of responses to data visualizations (Campos et al., 2021). Others found sensemaking heuristics which include comparing, monitoring, and exploring by teachers as they leveraged tools to support technology-supported collaborative learning (Voyiatzaki and Avouris, 2014). In addition, Molenaar et al. investigated how teachers make data visualizations actionable and the responses they implemented (Molenaar and Knoop-van Campen, 2019). Specific to our work, Chen et al. explored teacher dashboard usage to support problem-based collaborative learning at the college level (Chen et al., 2021). To our knowledge, limited research exists that explores how teachers notice, interpret, and develop evidence-based responses to students learning and problem-solving strategies for a K-12 PBL curriculum in science. As such, this research seeks to provide novel findings to support a deeper understanding of this critical need.

A third research opportunity targets supporting teachers in the interpretation process for improved decision making. For instance, researchers have evaluated the impact of different interpretive aids on teachers sensemaking to support collaborative learning (van Leeuwen et al., 2019). Although we do not provide specific interpretation aids in this research, we believe a deeper understanding of the processes teachers implement to transition from learning analytics data visualizations to decisions on pedagogical responses can aid in the future development of such tools.

Finally, there is a need to understand how resulting teacher interpretations of dashboard visualization facilitate evidence-based pedagogical actions (Campos et al., 2021). Unfortunately, not a lot of information can be found on the pedagogical actions teachers take as a result of using instructor-support technology, such as dashboards, especially for K-12 instruction. This research provides novel findings on example pedagogical responses resulting from the noticing, interpretation, and reasoning about student data during a problem-based, middle school science curriculum.

5.3 Co-Design of RISE Dashboard

This research focuses on teachers' responsive teaching practices supported by a teacher dashboard for a problem-based learning curriculum known as, Science Projects Integrating Computing and Engineering (SPICE) curriculum.

5.3.1 Instructional Context: SPICE

SPICE supports teachers in the implementation of the Water Runoff Challenge (Zhang et al., 2020; Chiu et al., 2019; McElhaney et al., 2020; Hutchins et al., 2021a). The Water Runoff Challenge (WRC) is a three-week, NGSS-aligned unit that challenges students to redesign their schoolyard using different surface materials to minimize the amount of water runoff after a storm, while adhering to a series of design constraints. These include the overall cost and accessibility, while providing for different functionalities for the schoolyard (Chiu et al., 2019). The problem-based learning curriculum consists of five core units, illustrated in Figure 5.1. These units include: physical experiments, conceptual modeling, paper-based computational thinking tasks, computational modeling of the water runoff phenomenon, and engineering design, in which students use their computational models to redesign their schoolyard. This learning context is authentic and relevant to students facing similar problems (limited usability and pollution) in their own schools, therefore, the WRC is potentially engaging and personally meaningful to the learners (McElhaney et al., 2020). The WRC targets NGSS performance expectations for upper elementary and middle school Earth science and engineering design curricula, emphasizing the movement of surface water in a system after heavy rainfall and the human impact of this runoff on the environment, and leverages evidence-centered design (Mislevy and Haertel, 2006) for the systematic creation of summative and formative assessments to evaluate student learning in science, computing, and engineering.

We focus this paper on Planning Period Simulations that target students' efforts to construct a model of a scientific process, i.e., water runoff after a heavy rainfall. These curriculum lessons offer unique perspectives on how teachers evaluate student data as pedagogical planning requires the evaluation of items such as how well students are translating their developing science knowledge into computational form, understanding the multiple paths students can take to successfully construct a computational model in science, and identifying successes and opportunities students are having in using difficult computational constructs such as conditional logic. Moreover, in the Literature Review we identified teachers' limited background in computing as an issue

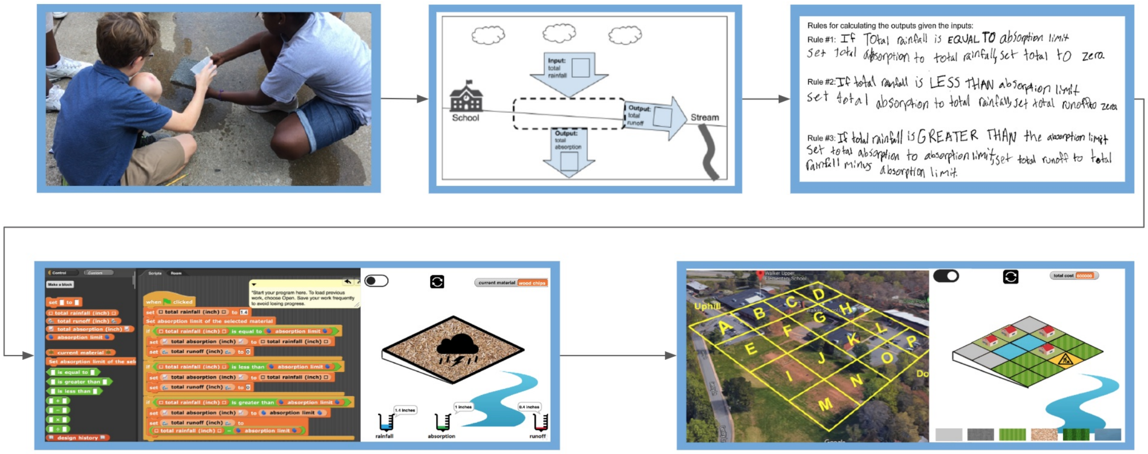


Figure 5.1: SPICE Curriculum Progression

for implementing such problem-based learning approaches and this allows us to examine ways in which the dashboard can help novice teachers.

5.3.2 Creating the RISE Dashboard

The dashboard leveraged in this work was created through a series of co-design design sessions with experienced and inexperienced SPICE teachers. For a more detailed presentation of our design process, please see (Manuscript Two).

As a first step, researchers used student data from prior implementations to increase our knowledge about how students learn and problem solving during SPICE. This involved the systematic analysis of student science, computing, and engineering learning as demonstrated through summative and formative assessments, evaluating the impact of student learning over a sequence of multiple, linked representations, and identifying key learning and problem-solving strategies students use to construct computational models and engineering design prototypes, based on their user actions in the learning environment, that support their learning in each domain (Manuscript One).

Then researchers initiated the co-design sessions by first using low-fidelity prototypes (e.g., linked data visualizations) and contrasting student artifacts as boundary objects to discuss, negotiate, and come to an understanding about what information teachers need about their students so they may better help their student during this student-centered, problem-based curriculum. Feedback from these sessions were used to inform the creation of the first high-fidelity prototype. In the second design sessions, we integrated the use of the high-fidelity prototype into a professional development workshop with SPICE teachers. As we reviewed the curriculum and discussed instructional strategies with participating teachers, we used the dashboard as a tool to discuss prior student performance on each lesson and assessment. Teachers thought aloud, describing what they noticed, how they interpreted the results, and possible actions they might take knowing this information. Researchers intervened and responded to questions as necessary. Teachers also provided us with more specific recommendations for user-interface adjustments (as opposed to abstract ideas from the first session). The research team used results and feedback from this session to create the Responsive Instruction for STEM Education (RISE; Figure 5.2) dashboard, used for the Planning Period Simulations.

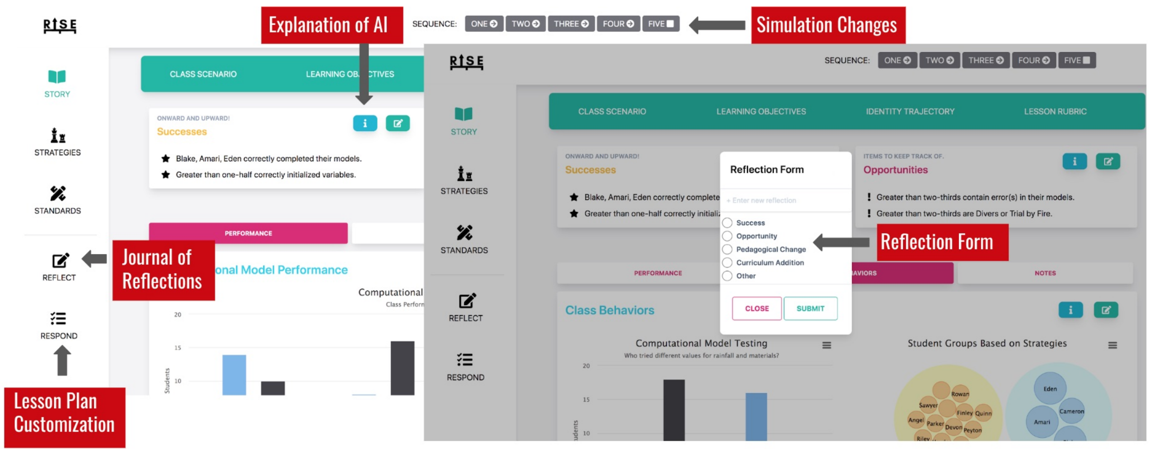


Figure 5.2: RISE Dashboard

The RISE dashboard consists of three core student result pages: the Story, the Strategies, and the Standards. The Story provides an overview of the class performance based on key immediate, or landing page, feedback recommended by teachers. This included text-based feedback highlighting class successes and opportunities using performance (items scored by pre-defined rubrics) and strategies (productive and unproductive strategies pre-defined based on the impact on student learning results). Interactive data visualizations, such as the grouping of students based on strategies, with additional performance-based results, could be accessed using information in the bottom right visualization shown in Figure 5.2. The Strategies page provided a progression of student performance over the course of the curriculum (e.g., up to the "day" simulated in each planning period simulation) and the strategy group they currently are identified with. Finally, the Standards provided a data table of all students with their scores on each completed curriculum task and identified strategy groups. All data visualizations in which artificial intelligence was used to calculate or provide feedback included an explanation of analysis done (for example, a modal pops-up with the information when the blue button with an "i" is clicked).

The RISE dashboard is equipped with a Reflection Tool in which teachers can add reflections as they reviewed the results (identified as "Reflection Form" in Figure 5.2) and select categories for the type of reflection. Submitted forms were populated on the Reflection page based on the category selected (the page link is identified on the left-side menu bar in Figure 5.2). In the Reflection page, teachers can re-order and reorganize reflections as they see fit. Finally, teachers are also provided a Response page. This page includes the current class plan for the next class and tools to plan for any adjustments they deem necessary based on student performance. Finally, teachers are provided a number of curriculum resources, including learning objectives and lesson plans relevant for the "day" to aid in their evaluation process.

5.4 Methods

5.4.1 Participants

Eight middle school STEM teachers (5 female, 3 male) participated in the planning period simulations. The teachers were from varying urban and rural locations, including Tennessee, Illinois, Virginia, New York, Wyoming, and the US Virgin Islands. Three teachers had prior SPICE implementation experience, one teacher had prior experience with C2STEM (the core learning environment; Hutchins et al., 2020a), and four teachers had no prior experience. All teachers consented to participate in the Vanderbilt University IRB-approved study.

5.4.2 Planning Period Simulation

Utilizing the RISE dashboard (Figure 5.2) we conducted a series of Planning Period Simulations. These simulations were inspired by the Teacher Moments research at MIT (Benoit et al., 2021) and were derived from the participatory design approach to simulations (Martin and Hanington, 2012).

During these simulations, each teacher first completed a 90-minute professional development session led by the research team in which they learned about the SPICE curriculum. Novice teachers completed the computational modeling and engineering design tasks much like students would in the classroom. Then the teachers completed a sequence of five simulations (as indicated in the top-middle of Figure 5.2), in which they would enact planning periods. For each simulation, a research team member first described the class scenario, including the class performance on the pretest, which targeted science, computing, and engineering knowledge, as well as other class results prior to the simulation "day" (all information was available on the dashboard as well). Teachers then had 15 minutes to complete the simulation exercise. Fifteen minutes was selected based on an estimated class period time length of 60 minutes and an average estimated class roster of 4 classes per teacher, therefore 15 minutes per planning period for each class.

Student data used for each simulation was pulled from prior SPICE implementations in an approach similar to the Replay Enactment protocol implemented by Holstein et al. (2019). The simulations were created based on prior class performance on the SPICE summative assessments. We first identified the median learning gains split for all potential classes based on summative assessment results in science, CT, and engineering and labeled classes as low or high performing based on these splits. Three of the classes were identified from prior SPICE implementations. These classes consist of (1) high performing in science and low performing in CT, (2) low performing in science and high performing in CT, and (3) low performing in both science and CT. We create an additional simulation class in which we combined 24 (the average class size) high performing students in order to have one class that was high performing in each subject. Two of the simulations visualized student learning and problem-solving strategies after the first day of computational modeling and three of the simulations target the second (of three) days. The high performing in science and low performing in CT class was used as the first simulation and the fifth simulation in order to get teachers to reflect what they may have done differently in the first simulation (first), knowing the results of the fifth simulation (the second day). Student data from the prior implementations were de-identified and students were given gender-neutral names.

Using a think-aloud protocol (Martin and Hanington, 2012), teachers reviewed student results and feedback provided on the RISE dashboard, interpreted what they saw, and customized class lesson plans for the next day (as they saw fit), as though each class simulation pretained to what happened in their class and they were making decisions during the planning period on what to do for the next day. Prior research has noted the benefits of think-aloud protocols on tasks involving building interpretations (Charters, 2003), including providing a low-entry barrier (Campos et al., 2021) and tracing users' thinking (Liu and Stasko, 2010). In order to obtain verbalizations that accurately reflected the cognitive processes teachers implemented during responsive teaching, we refrained from providing detailed instructions or interpretation of results (other than noting the ultimate goal was to customize tomorrow's lesson plan, as needed). Instead, we utilized prompts such as "what possible actions would you take with this group?" and answered questions about technology that did not impact class evaluations (e.g., describing how to use the reflection form). This approach is modeled after Campos et al. 2021's approach for evaluating teacher sensemaking. This helped minimize issues concerning bias in data if researcher support or feedback impact teachers' responses (Sherin and Russ, 2014).

Finally, these procedures were conducted using videoconferencing software. For each simulation, teachers shared their screen so that the researcher could view as well. Researchers completed an observation sheet during the simulations. The observation sheet consisted of a table for researchers to identify (1) discussed idea (e.g., computational model scores), (2) visualization targeted, when applicable (e.g., bar graph of class performance on computational model), and (3) key words used or links made (e.g., poor initialization of science variables score relating to class science performance during the science unit). These observations were used to support our analysis approach, discussed below. Figure 5.3 provides an example of the researcher view. The teacher's video (top-right) and browser information (saved tabs, including identifiable information) have been hidden.

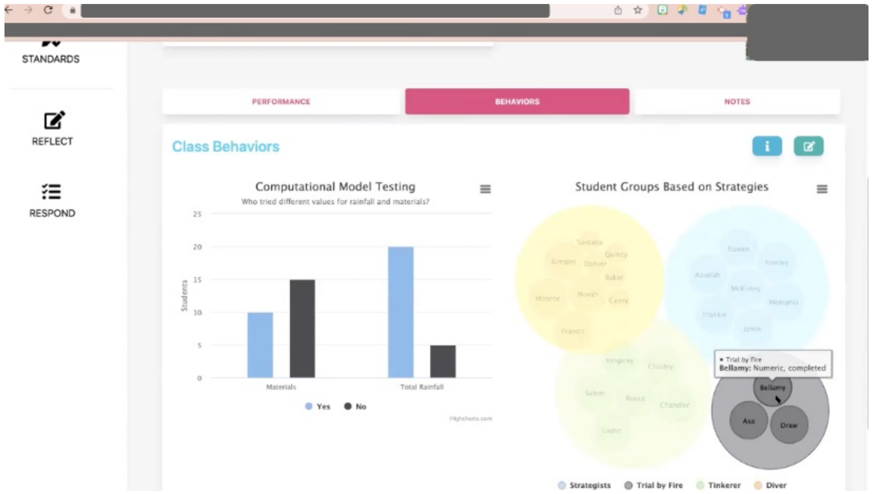


Figure 5.3: Example view during Planning Period Simulations

5.4.3 Data Collection and Analysis

All Planning Period Simulations were conducted virtually and recorded using a video conferencing platform. In total, we had approximately 12 hours of video data, which we transcribed using an online transcription service. For the purpose of this paper, we segmented the transcripts into episodes of pedagogical reasoning (Horn and Little, 2010). In this case an episode of pedagogical reasoning was initiated when the researcher completed the opening statement about the class scenario and ended when the teacher submitted their customized lesson plan. An example episode is available in Appendix C. These segments formed the base unit of analysis to answer both research questions.

To answer **research question one**, we utilized epistemic network analysis (ENA; Csanadi et al., 2018) to interpret how expert and novice teachers' interpret and respond to students science and CT knowledge and problem-solving strategies as they construct their computational models. Recent code-and-count analytic approaches have been criticized for ignoring temporal contexts of discourse, which is particularly relevant to the understanding of the processes teachers implement from using and understanding data visualizations of student learning to enacting evidence-based pedagogical responses. ENA has been shown to overcome this limitation and find temporal relationships in data (Csanadi et al., 2018). In education, ENA has been used to analyze collaborative problem-solving (Hutchins et al., 2021b), how collaboration support science knowledge construction (Bressler et al., 2019), and understanding students' assessment responses (Irgens et al., 2020). More recently, ENA has been used to evaluate the impact of alerting dashboards for teachers on student learning through science inquiry (Dickler et al., 2021), and serves as the motivation for our analytical approach.

To conduct this analysis, we first divided the episodes of pedagogical reasoning into smaller excerpts related to idea units, in which a single topic was discussed (Jacobs and Morita, 2002), in order to balance our units of analysis. This resulted in 735 idea units. A coding scheme (described in Table 5.1) targeting noticing and interpretations was developed by leveraging past work the analysis of responsive teaching during video clubs (Johnson and Forsythe, 2015) and teacher dashboard usage (Chen et al., 2021) and incorporating additional categories pertinent to our work, including teachers' discussion of problem-solving strategies.

Table 5.1: Coding Scheme for Teacher Dashboard Evaluations

Code	Definition	Example
Curricular (CURR)	Questions or comments focused on the teachers own understanding of the ideas in the lesson (adapted from Johnson and Forsythe, 2015)	"How much instruction do students get to complete the first rule?"
Descriptive (DESC)	Discussed content-based information they obtained from the dashboard (adapted from Chen et al., 2021)	"OK so 12 students completed their model correctly."
Interpreting Performance (N-PERF)	Questions or comments focused on the simulation students' understanding of the science, computing, engineering concepts (adapted from Johnson and Forsythe, 2015)	"Ok, it looks like they really do not understand how to calculate total runoff when rainfall is greater than"
Interpreting Strategies (N-STRAT)	Questions or comments focused on the classroom students' application of strategies	"It looks like this class is really struggling with testing materials" "There are a lot of divers!"
Integrating Multiple, Linked Domains (N-MLR)	Questions or comments focused on the sequencing of content and trajectories of student learning (adapted from Johnson and Forsythe, 2015)	"Another benefit of testing materials is that I can help them relate it to the science experiments we did and it will reinforce things to look for when they design their playground."
Regulative (REG)	Reflections on the teacher's pathways of exploring the dashboard or strategies they used to interpret the visualizations (adapted from Chen et al., 2021)	"So now I will look at strategies." "I love looking at bar graphs so I will go there first"
Instructional (INST)	Questions or comments focused on the resources and pedagogical moves used to convey science, CT, or engineering ideas (adapted from Johnson and Forsythe, 2015)	"I'm not sure if the debugging task is in the right place. If they are struggling with debugging after the first day, it will continue unless we intervene"
Technology (TECH)	Thoughts on how to explore the dashboard and to look at different visualizations, including recommendations for dashboard adjustments (adapted from Chen et al., 2021)	"When I look at these circles, I'm looking for students that moved to more productive strategies. It would be nice to highlight or color those changes."

We also developed a coding scheme to evaluate teacher discussions on evidence-based response creation. To do so, we targeted discourse that discussed the social level (Dillenbourg, 2015) of the activity (e.g., teacher lecture, class activity, group activity, or individual activity) and the context of the response (e.g., is the response focused on conceptual knowledge, problem-solving behaviors, linking multiple representations, or technology issues). The codes for evidence-based responses can be found in Table 5.2.

Researchers met to code idea units using these schemas together. Differences were discussed and refinements were made to the coding scheme. The researchers then coded 20 percent of the idea units and achieved

Table 5.2: Coding Scheme for Teacher Evidence-Based Responses

Code	Definition	Example
Teacher Lecture (LECT)	Teacher plans to add class lecture on a topic based on data	"Students are struggling with initializing variables and so do I so I will add 5 minutes at the beginning of class to connect their struggles to mine and how these initial variables are like what we did in the conceptual model."
Class Activity (CLASS)	Teacher plans to add activity involving the class as a whole	"I will have Taylor present how they completed the first rule and I will be sure to ask questions or discuss how students can check if the rule is correct"
Group Activity (GROUP)	Teacher plans to add activity in which students work in groups	"I will group Divers and Strategist so that Divers can see the importance of testing materials"
Individual Feedback (IND)	Teacher schedules individual student feedback based on data	"This student continues to struggle in science, so I will set aside time as the class works to help them with their science knowledge"
Conceptual (CONC)	Teacher response targets domain-specific knowledge	"We will discuss the difference between total absorption and absorption limit"
Strategy (STRAT)	Teacher plans activity demonstrating productive testing strategies (e.g., to help student(s) debug models)	"At the beginning of class, we will do the debugging tasks together and I will demonstrate the benefits of testing different values of rainfall or materials" (also, any discussion of Divers, Strategists, Trial by Fire, or Tinkerers)
Linking Multiple Domains (LMD)	Teacher plans activity that links multiple domains (e.g., teacher links testing rainfall in model to their physical science experiments testing different amounts of water)	"This class is going back outside to continue testing different rainfall values, and then implementing similar tests on the computer!"
Instrumental (TECH)	Teacher response targets the use of a technology tool (e.g., clicking on the design history table, how to change materials)	"This student has not changed any materials. I will demonstrate how to tomorrow"

good IRR agreement (k $>$ 0.80). The researchers discussed differences and once they were resolved, the main author coded the remaining idea units. These coded units were used to build the epistemic networks. The epistemic networks (see Figure 5.6) were created using the ENA online graphical interface (epistemic-network.org). Nodes represented the codes from Tables 5.1 and 5.2. The lines (and strength of the lines) represent the connections between nodes and the frequency of co-occurrence. This allows us to evaluate temporal patterns in discourse and we evaluate differences in epistemic networks of expert and novice teachers during the episodes of pedagogical reasoning to answer the research question.

To answer **research question two**, we evaluated simulations as episodes of pedagogical reasoning (Horn and Little, 2010). We used methods of inductive coding and constant-comparative analysis (Charmaz, 2006) as opposed to theoretically developed codes. To our knowledge, there is very little research examining how

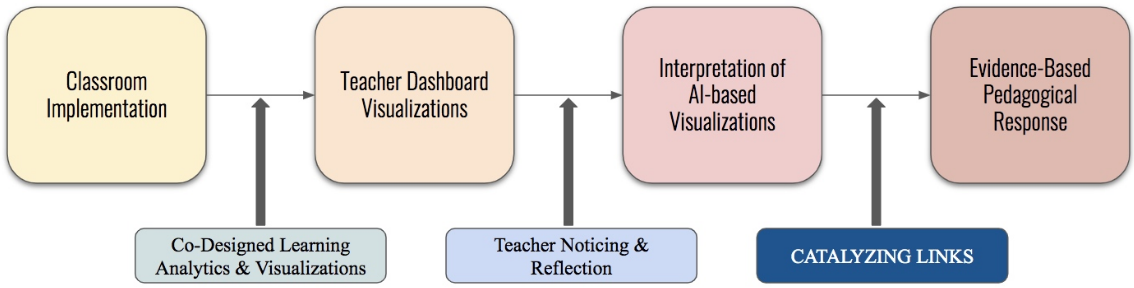


Figure 5.4: A dashboard-supported responsive teaching process.

the resulting interpretations facilitate pedagogical actions. This exploratory work led us to identify *catalyzing links* that teachers used to transition from their interpretations of AI-based data visualizations to evidence-based lesson customizations. Figure 5.4 details a resulting dashboard-supported responsive teaching process.

We develop conjectures about these links and their implications on teacher responses. Team members met to discuss episodes and the links that teachers applied in these episodes. We created analytical memos (Hatch, 2002) to help us then compare teachers' catalyzing links. In the discussion of the links identified, we reverted back to the literature on processes that support learning in integrated domains to refine our understanding of the links to help us define the emerging patterns. In the context of the full picture of all pedagogical episodes, we noticed the recurrence of similar patterns in catalyzing links (e.g., supporting learning through multiple, linked representations) and planning period simulations, which suggested patterns exist in the relationship between interpreted classroom needs and class performance distributions. We provide illustrative examples of each pattern and provide a contrasting case.

5.5 Results and Discussion

5.5.1 RQ1: Teachers' responsive teaching practices using RISE

Following the data processing of the 8 teachers there were 453 idea units generated by the expert teachers and 278 idea units generated by the novice teachers. We argue it was partly due to the nature of the idea units. For example, novice teachers had a greater amount of Curricular codes, a median of 12 per simulation by the novice teachers and 3 by the expert teachers, which include questions or comments focused on teachers understanding of the curriculum. These idea units typically involved researcher response, and, therefore, a higher number of researcher input during the allotted 15 minute time.

Figure 5.5 illustrates the breakdown of noticing codes (labels identified in Table 5.1). As seen in the novice teacher pie chart on the right, novice teachers spent almost half of their time discussing the curriculum and the dashboard technology. Interestingly, both expert and novice teachers had about the same number of idea units targeting performance and student strategy interpretations (in yellow and green in Figure 5.5).

Novice teachers had a median of 9.5 and expert teachers 9 segments targeting the interpretation of student performance. In addition, novice teachers demonstrated a median of 8 interpretations of student strategy usage while expert teachers had a median of 12 (we argue the higher amount by the expert teachers is reflective of teachers' experience with testing strategies from prior classroom implementations). The key difference between the groups in terms of noticing and interpreting involved interpretations of the results from the perspective of multiple-linked representations, with novice teachers not demonstrating any such segments, while it was the focus of 7% of expert teacher noticing. This is important as it connects to our dashboard goal of supporting teachers in facilitating the integrated learning of science, CT, and engineering through multiple, linked representations. While it did support expert teachers, more work needs to be done to support novice teachers. In addition, these results impacted teachers evidence-based response codes, as illustrated by the ENA graphs in Figure 5.6.

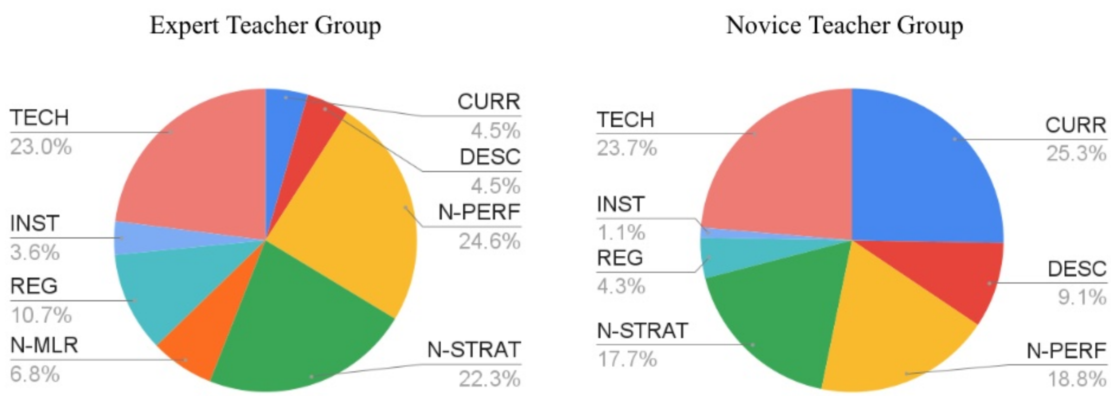
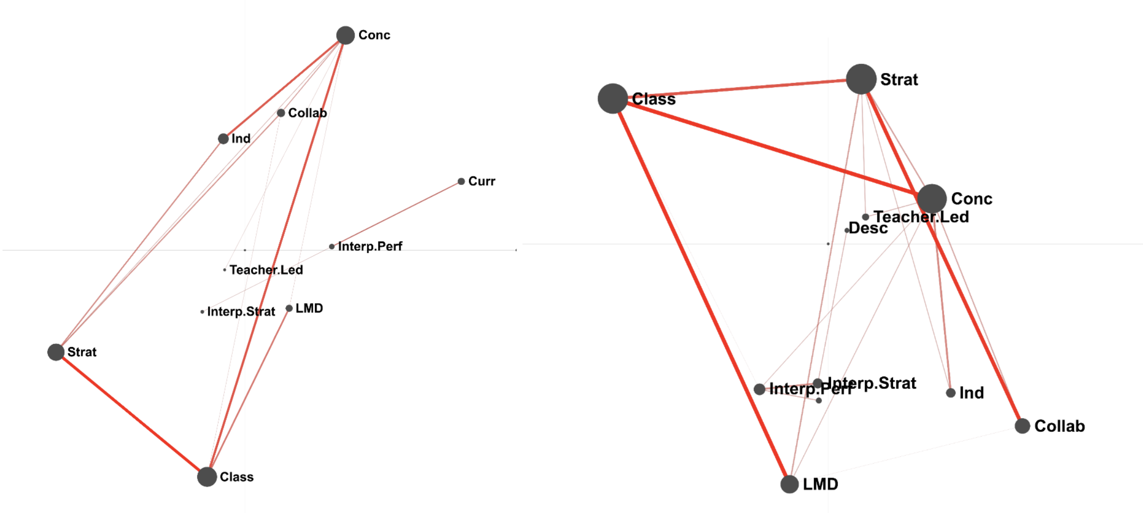


Figure 5.5: Expert and novice teacher noticing results.

Following the coding of the pedagogical episodes, we ran epistemic network analysis to evaluate the links between idea units. The highest three link probabilities for the novice group of teachers were (1) class-level activity response and strategy response (0.36), (2) class-level activity response and concept-targeting response (0.28), and (3) individual student response and concept-targeting response (0.26). The expert group's highest link probabilities were (1) class-level activity response and concept-targeting response (0.38), (2) class-level activity response and multiple-linked representations-targeting response (0.37), and (3) collaboration-level activity response and strategy-targeting response. These results seem to indicate a link between the role of interpreting student results on the dashboard from the perspective of multiple-linked representations and developing responses that support students in making those links. In addition, it is interesting to note that expert SPICE teachers were more likely to customize lesson plans to target strategy improvements using a collaboration approach (e.g., pairing students to compare debugging processes) and novice teachers were more likely to rely on individual student responses when faced with conceptual issues (e.g., speaking one-on-one to a student struggling to initialize needed variables).



Inexperienced SPICE Teacher Experienced SPICE Teacher

Figure 5.6: ENA Graphs

Overall, these results demonstrate that teachers reflected on the data and developed evidence-based responses at multiple social levels. In addition, both groups were able to develop pedagogical customizations that targeted both conceptual knowledge improvements, and the development of problem-solving skills or strategies using the dashboard. We hypothesize that data visualizations or tools to aid in data visualizations, such as those developed by van Leeuwen to support teachers interpretation of collaborative learning results (van Leeuwen et al., 2019), may better support novice teachers noticing and interpretation processes.

One major limitation of this analysis approach, is we are not able to see the processes that transition teachers from noticing and interpreting to the development of those evidence-based responses identified in these figures. We tackle that in the next section.

5.5.2 RQ2: From Interpretations of AI-based Learning Analytics to Evidence-Based Customizations

The results are discussed in two sections: we first detail the patterns catalyzing links identified through our analysis process with example discourse and we then provide a contrasting case comparing teacher processes as they customize a lesson plan for the same simulation.

5.5.2.1 Patterns of *Catalyzing Links*

Utilizing the exploratory analysis process described in Section 5.4, the team identified five key catalyzing link patterns utilized by teachers to support the decision-making processes needed to transition from interpretation of AI-based analysis of students results to evidence-based lesson plan customizations. We describe each below with illustrative examples identified by the research team.

Supporting Student Understanding Across Multiple, Linked Representations. As represented in the ENA results, a key process undertaken by teachers is determining how to support students understanding across multiple linked representations based on multiple performance and strategy result data. For instance, a novice SPICE teacher noted was weighing different options for lesson customizations, including running another physical science experiment, and said *"I'm still thinking about the materials. How to get them to transfer that original [engineering design] grid you'd set up to, you know, to that they have to have the different values for the materials. Because it's still more than half [that aren't testing].And that's why I told you, I love to see the Data Summary. I think those avert connections between the lab experiment [in science] and the [computational] model. We make those implicitly as adults, but I think it needs to be you know, it it needs to be more obvious for a younger brain. Yeah. To connect the model to the real thing."* In this example the teacher recognized that as adults, we may automatically connect the SPICE computational modeling practices (e.g., testing the computational model with different materials) to the material experiments conducting in the previous SPICE unit; however, more effort needs to be made to support students in deriving those links because understanding these connections can be very useful during the playground design task.

Similarly, an expert SPICE teacher was reasoning about why they wanted to return to the multiple conceptual models students make during the science unit in order to help them identify patterns in the computational model representation. The teacher said, *"That's that was my point about the multiple representations, because they're figuring out the patterning. But do they really know what that's doing? Realistic. What the actual [model is doing]. That it's raining this much, and this much runoff is this and as much as absorbed and all that. So that's where you're doing something like where we're having to literally explain. So here's what you coded. And here's what it did. Why did it do that? What actually dos that mean?"* In this example, the teacher reflects back on it being necessary to specifically ask students about what a model represented or meant and that students struggled with it. The use of multiple, linked representations here is to help students make the connection between patterns identified in the conceptual model to the computational model and, hopefully, support their understanding of what the computational model represents. Moreover, although not explicitly discussed, these multiple representations are also helping the transition from the conceptual model (e.g., understanding the conservation principle in science) to the construction of a computational model (which requires additional thinking and application about specific CT concepts and practices).

As mentioned in the previous analysis section, additional work is needed to support novice teachers in this process.

Leveraging Student Successes. A common process implemented by all instructors was to take advantage of the "Successes" feedback shown on the landing page of the dashboard (developed through the co-design process). Teachers used this data as a method for (1) planning lesson plan timelines and promoting future success and (2) motivating and engaging students that aren't known to be computing enthusiasts or are new to computing.

In the first example, an expert teacher described, *"Well, I think in two, you probably need to reflect on the success of the [initializing variables]. Because they did. They were successful for the majority, but I think it would be good to reflect [on that] because that may push them to do better on this, the equal to condition. Does that makes sense? So how would I say that? Initial reflect important reflection on initial variables as a class could lead to success. Because if you if you focus on success, it drives success. Rather than say, Oh, y'all did a good job, let's go on to the next one."* In this case, the teacher was developing a timeline by taking advantage of known successes in order to support students' construction of other difficult computing constructs such as conditional logic.

In another example, an expert teacher was describing how changes in data visualizations can help target students in a manner that is motivating and engaging. In this case, the teacher was reviewing the data visualization shown on the bottom-right of Figure 5.2 of the Strategy Groups. She noticed the change in a student from an unproductive strategy group (e.g., did not systematically test their code during the initial code construction process) to a productive group (e.g., checking code with different values of total rainfall and materials). The teacher commented, *"I think this is just on a thing. This is the nicest thing I like about this is here, like, this is great. I'm going to tell you why I like this. This part particularly, is that I think you know, what I'm trying to do as a teacher is I am trying to get, you know, kids to be more the strategist, or even the tinkerer kind of thing and, definitely with coding, getting getting kids away from being a diver [not testing code]. So seeing who's doing those techniques, is really going to help me and then seeing who changes because sometimes in the moment, I'm only picking on the kids that I know are strong and CT, to show examples. And I think that can be a bit demoralizing for other students. So like as this goes on, let's say Kendall, all of a sudden jumps into strategists or something like that, that's like a great thing. But if I'm able to see like, someone made the jump from here to here, or here to here, I can then highlight them and hopefully give them a you know, some nice positive praise, reinforcement kind of thing that I think would be really helpful."* In this example, the teacher demonstrated the benefits of how the analysis groups students based on strategy and how we can track student group changes over time. The teacher reflected on their own practice, noting they recognized the inefficiencies of calling on students known to be proficient in computing and how AI-based tools can help teachers identify students they may not have thought of. We will see more detail of this process in the case example below.

Representing, Addressing, and Leveraging Productive Failure in Problem-Based Learning. A process implemented by only 3 of the 8 teachers centered on understanding, addressing, and utilizing productive failure during PBL, and in particular, computational modeling. A novice teacher described, *"The other thing that I look at a lot is normalizing mistakes. And so if I would, it becomes tricky. And it really you have to normalize it from the first day. We all make mistakes, but mistakes can help us get better at it and have someone share a mistake related to the materials where everybody looks at it together to figure out"* In this example, the teacher recognized that the testing of different materials during computational modeling was a common problem and reasoned about how to use the issue as a productive tool for building the problem-solving skill.

This teacher, using previous experience and pedagogical content knowledge, then discussed a possible approach for targeting productive failure: *"One of the things I would be I would be doing in response to that is, the way I do it in my class, is it is becoming really standard and is called notice and wonder, where [students] look at something like you could have them look at that. What do you notice? And what do you wonder, and then everybody talks about it. And so you could, it's a way to frame either that debugging piece or where a student is successfully tested a new material. But it takes away some of the fear of being wrong. Because what something you notice, how can that be wrong? And what do you wonder? You know, it's a really nice framework for jump starting a conversation or getting kids to start to dig into something I want to focus on."* Taking initially negative students results, the teacher utilized a process in which they reasoned about productive failure and what it looks like in this curriculum, leveraged pedagogical content knowledge and prior experience, and created a potential lesson adjustment to target the identified issue.

Weighing Responses at Multiple Social Levels. A common process implemented by all teachers was the determination of the optimal pedagogical response amongst options targeting different social levels (e.g., teacher lecture, class activity, smaller group activity, or individual support). One teacher approached it from the perspective of a medical doctor triaging: *"When you look at the dashboard, and you see those bigger amounts of needs, that's when you have to go triage time, and you got to think about okay, I'm gonna have to do something much different here. Because I've got a lot of misconception. Yeah, that's where a bigger action will occur."* Understanding the impact of lesson plan customizations and the intrinsic and extrinsic constraints of the classroom (c.f., Dillenbourg, 2015), weighing potential activities at different social levels becomes a very complex task.

One common issue is determining how much time to spend with groups or individuals based on conceptual and behavioral needs. One teacher described, *"Yeah, and because every day that you get deeper into an investigation, there's a bigger leg between the top of the group and the bottom of the group. Yes. And I don't want my top end, twiddling their thumbs waiting for the bottom end to catch up. They need to keep keep going."* This teacher described the need to potentially have additional curriculum resources for advanced students to ensure they were engaged. In a similar situation, to be described further below, a teacher elected to have students that finished their models return to the conceptual model written responses and improve what they had written previously.

Integrating Real-World Contexts. Problem-based learning immerses students in understanding, solving, and communication solutions to real-world problems (Hmelo-Silver, 2004). This curricular context was shown to provide a unique opportunity to support the transition from understanding students results to implementing lesson plan customizations. However, it was only implemented by 3 teachers.

In one example, the teacher went back to the overall context of the engineering design problem, that students would become a project manager in which they complete engineering designs for a playground that meets specific stakeholder constraints.

> *"And I always worry about those students in the classroom who are done. Okay, now, what do you do that you're done? I would be assigning them to work with someone who, who needed more support, and say, this person, okay, now, Reese is going to be an expert. You can consult an expert with your work and bring in a consultant. And you can ask them three questions...you got to explain. But I would, you know, I would put a constraint on it. But I think that's why this could be really valuable to know that they're completed. So that you can say, okay, this group is going to be consultants, but kind of put a constraint on it. Like, you can only ask a consultant a question, they can't just tell you stuff. Yeah. And it can't be a question like, how do I write the code? Because the whole framing of the problem is, you are a project manager and you're designing a playground. So you need like, you're gonna work with this team. And you're, you know, it's okay that you're a consultant. You are a part of the team that knows the computing really well or that's what do you call somebody like that in the company? Is that an IT? Like I think that it's manager or I think the designation for somebody that works in the company there."*

This teacher identified an opportunity to connect the real-world context of the curriculum's problem to create a lesson customizations that supported collaborative work and allowed for students that completed to reinforce their knowledge by assisting other students. Moreover, by implementing a group activity, the teacher could observe the types of questions created by students as well as consultants' responses. The teacher said they would "become a spy" and help "if they got stuck," but would not provide direct support too often to ensure the student-centered learning design.

5.5.2.2 Contrasting Cases

The identified patterns in our exploratory analysis provide an initial framing to understand how teachers may transition from their data interpretations to evidence-based pedagogical responses. In this section, we explore these transition in more depth through a comparison of two episodes of pedagogical reasoning by an expert and a novice teacher. Both cases involve an evaluation of a class that was low performing on the pretest in science, but high performing on the computing pretest.

Expert SPICE Teacher. This first example involves an expert SPICE teacher. After the researcher provided the Simulation Scenario, the teacher began their think aloud process:

> TEACHER: *I always like looking at graphs first, that's just I love graphs, I'm gonna go to that.*
>
> TEACHER: *So they had this same problem with initializing their variables. They did better at equal to, and then less than and greater than so not too bad. And this is probably reflecting that like, they were numeric.*
>
> TEACHER: Okay, great. So with this? How am I maybe hold on a minute, let me look at this to see if there isn't see if there's an answer to this. Right, so they didn't test different materials.
>
> **Teacher asks technical question about the availability of data**
>
> TEACHER: *So yeah. So knowing I guess, knowing that, I would say because they did just on this because they did not so great a job here but they did a better job here. So they are getting some of this computing stuff. Better. So I would say add in. Quick physical demo. Why? demo showing difference between absorption Have sorption for sponge, and like paper, something like that. So they can just understand why they need to do that. So let me put in that*

In this first segment, the teacher linked students' prior performance in science with issues concerning the initialization of science variables in the computational model. The teacher then checked student behaviors to identify if any other data was available to explain this issue. They then clicked to add a Reflection (a technical issue arose) and the teacher then added a reflection noting they would need to conduct another science demonstration to show the difference between the absorption of different materials to help students understand why they would need the science variables in their computer models. As such, the teacher utilized the process of ***Supporting Student Understanding Across Multiple, Linked Representations*** to determine a possible evidence-based response. So far, the teacher has only viewed bar graphs illustrating the number of students who got initial variables and the conditions correct and how many students tested more than 2 materials. The teacher continues.

> TEACHER: *Let me look at behaviors. So yeah, so this is like, I think that I'm going to add in this there. So this is a put this opportunity need to see more physical examples to see importance of material. So I think that's really important. Because they did a better job here.*
>
> TEACHER: *And yeah, we've got a lot of divers. So we need to we need to fix we need to, we need to fix that again. And I'm going to add that add into class.*

> TEACHER: *Let's see, let's do Palmer, their circle. Does that do the circles within the circles that circle like that Palmer has a bigger circle than justice. Does that mean like Palmer? Did it better than Justice? Because that's how I read that.*
>
> *RESEARCHER:...[researcher describing what arrow sizes mean]*
>
> TEACHER: *okay. So they're gonna, they're gonna show and talk through their code. I think this is just on a thing. This is the nicest thing I like about this is here, like, this is great. Like, this is really, I like I'm going to tell you why I like this. This part particularly, is that I think you know, what I'm trying to do as a teacher is I am trying to get, you know, kids to more the strategist, or even the tinkerer kind of thing and getting definitely with coding, getting getting kids away from being a diver. So seeing who's doing those techniques, is really going to help me and then seeing who changes because sometimes in the moment, I'm only picking on the kids that I know are strong and CT, to show examples. And I think that can be a bit demoralizing for other students. So like as this goes on, let's say Kendall, all of a sudden jumps into strategists or something like that, that's like a great thing. But if I'm able to see like, someone made the jump from here to here, or here to here, I can then highlight them and hopefully give them a you know, some nice positive praise, reinforcement kind of thing that I think would be really helpful. So I'm really I'm digging this right here.*

In this segment, the teacher further acknowledged the need to understand why testing different materials, now from the perspective that there are a lot of students that did not test their computational models (e.g., they were categorized as divers). As such, the teacher connects science practices and computational practices. In addition to conducting the science activity, the teacher elects to do a class presentation in which they will select a student to demonstrate their code and their testing practices. Using the ***Leveraging Student Successes*** approach, the teacher selects a student based on the strategy group visualization and the students change to a more productive strategy group in order to promote the students' good work and improvement. The teacher concludes with their customized lesson plan:

> TEACHER: *All right. And then so we still have the initializing variables, and the two thirds are drivers or drivers or trial by fire. Okay, so I think let me go to reflect. Yeah, so I think having that physical example is really important for this class. And then maybe Palmer you know, maybe I bring in Palmer towards the end of class instead, for this group, and it because maybe the physical demo will help more. And then I can add Palmer in to wrap up.*

To address a conceptual issue regarding initializing variables and poor strategy preformances by the class, this teacher used the processes of ***Supporting Student Understanding Across Multiple, Linked Representa-***

tions and ***Leveraging Student Successes*** to determine and finalize an evidence-based lesson plan customization.

Novice SPICE Teacher. In this case, a novice SPICE teacher is presented with the same class simulation. In this case, the teacher utilizes process of ***Weighing Responses at Multiple Social Levels.*** and ***Integrating Real-World Contexts*** to create an evidence-based customized lesson plan for the class.

The episode begins with the teacher identifying a class issue:

> TEACHER: *Would it help to have samples of those materials on display in the classroom? Are they already doing that?*
>
> *RESEARCHER:[researcher stating materials are available]*
>
> TEACHER: *I mean, it looks to me, like that's the biggest need for the next day is to address the materials portion.*
>
> TEACHER: *And I like that this makes it clear. You know, you have a majority of students who are okay with that they're still you know.*
>
> *RESEARCHER:... [researcher agreement]*

The teacher first begins with a curriculum question about the availability of physical materials (related to the results from our ENA analysis, above, regarding novice teachers' curriculum codes). The teacher then identifies that testing materials is a problem and the ease of the visualization in identifying that. The teacher continues:

> TEACHER: *And I always worry about those students in the classroom who are done. Okay, now, what do you do that you're done?*
>
> TEACHER: *I would be assigning them to work with someone who, who needed more support, and say, this, this person, okay, now, Reese is going to be an expert. You can consult an expert with your, your work and bring in a consultant. And you can ask them three questions.*
>
> RESEARCHER:... [researcher agreement]
>
> TEACHER: *You got to explain. But I would, you know, I would put a constraint on it. But I think that's a challenge, that's why this could be really valuable to know that they're completed.*
>
> TEACHER: *So that you can say, okay, the group that are going to be consultants, but kind of put a constraint on it Like, you can only ask a consultant a question, they can't just tell you stuff. Yeah. And it can't be a question like, how do I write the code? Yeah.*
>
> RESEARCHER:... [researcher agreement]

In this segment, the teacher applies ***Weighing Responses at Multiple Social Levels*** to reason about a potential lesson plan customization. The teacher also utilizes pedagogical content knowledge (the consultant activity was described by the teacher in prior discussions during SPICE training) to determine a productive response. In discussing the idea further, the teacher said:

> TEACHER: *And this I think, I like also, because you can see who needs the most support. Yeah, you know, that they're kind of stalled.*
>
> RESEARCHER:... [researcher agreement]
>
> TEACHER: *I like looking at the written feedback that you're pinpointing, you know, where there are opportunities, but that can also help you target the consultant. Yeah. I like that.*
>
> TEACHER: *...Because the whole framing of the problem is, you are a project manager and you're designing a playground. So you need like, you're gonna work with this team. And you're, you know, it's okay that you're a consultant. You are a part of the team that knows the computing really well.*
>
> RESEARCHER:... [researcher tech issue]

In this segment, the teacher describes the opportunity presented by this pedagogical approach - being able to listen in on what consultants and project managers (the finished and in-progress students) are saying during this paired activity to identify students that may need additional support. The teacher reasons about the choice of lesson activity by ***Integrating Real-World Contexts***. The teacher concludes:

> TEACHER: *Yeah. Yeah, I think this is kind of invaluable. Alright, so the lesson plan is [coming together] I think the lesson plan for this is more, we'll have the consultants and the students that need work will create the three questions. And yeah, to have more complete the testing behavior. Make sure everyone has like I also call them experts. calling you an expert. I do let them [talk to me] if they're stuck, I do become a spy. But you still remember, don't do that often there. You know.*

The teacher reiterates the reasoning behind the choice of activity. In addition, they acknowledge the underlying, student-centered design of the curriculum by noting that they will intervene when necessary, but limit that approach.

This contrasting case demonstrates key catalyzing links implemented by an expert and an novice teacher and highlights differences in each approach. For this simulation, the expert teacher utilized the catalyzing links of ***Supporting Student Understanding Across Multiple, Linked Representations*** and ***Leveraging Student Successes*** to determine their lesson plan customizations, while the novice teacher utilized ***Weighing Responses at Multiple Social Levels*** and ***Integrating Real-World Contexts*** in their lesson plan customization. Similar to the ENA results, the expert teacher recognized the importance of a strong science knowledge on the translation of science into computational form and how group discussions on the underlying science phenomenon in the context of the computational modeling unit may help students link those two representations. In addition, to promote class discussion while also motivating students that may not have prior CT knowledge (but show growth), the teacher leveraged a student success after reflecting on how they typically call on experience CT students for class demonstrations.

While the novice teacher did not identify the integrated learning issues, this teacher did incorporate activities to promote student communication about their developing science and CT knowledge. In this case, the teacher selected students who would become consultants for the class, allowing other students an opportunity to ask the consultants three questions. The activity, inspired by real-world project management processes, would also allow the teacher to "spy" on student discourse. We hypothesize that this may allow the teacher to better understand what student difficulties may be (potentially including translating science into computational form).

5.6 Conclusions and Future Implications

This research presents a novel exploration into the processes teachers take to notice and interpret learning analytics from a co-designed dashboard and then reason and enact evidence-based pedagogical adjustments through lesson plan customizations. In particular, this research illuminates differences between expert and novice teachers' dashboard-supported responsive teaching practices as they prepare to teach a problem-based learning curriculum. In addition, this exploratory work provides a preliminary framework for identifying and evaluating catalyzing links teachers implement to decide and create evidence-based pedagogical adjustments based on AI-based analyses of student learning and problem solving.

Despite efforts to promote data-informed decision making in the classroom, there is scarce research examining how teachers utilize instructor-support technology such as teacher dashboards (Farrell and Marsh, 2016). This is exacerbated in the context of problem-based learning designs, as teachers must not only understand complex data analyses of students' problem-solving behaviors, they must leverage that information to design evidence-based pedagogical adjustments that enact a student-centered approach to learning. Evaluating teachers evaluation processes not only contributes to our understanding of how data promotes changes in instruction (Farrell and Marsh, 2016), but it can:

- support the development of tools to aid in teachers' noticing by interpreting the complex learning analytics (e.g., van Leeuwen, 2015) that target their background and experience (such as supporting novice teachers understanding and confidence in the curriculum and the impact of student results on students' learning trajectories, as seen in our work),

- improve resources to support evidence-based responses (e.g., teachers anecdotally recommended a list of expert teachers customizations based on similar class results as those in the simulations to support response decision making in the future),
- improve teacher training on responsive teaching for PBL (e.g., in the future after novice teachers have completed their simulations, they could be presented with examples of what expert teachers did in the same situation and reflect on the options), and
- improve visualization of feedback based on teachers' pedagogical needs (e.g., supporting teacher and coach sensemaking using data visualizations Campos et al., 2021).

In our work, although novice teachers utilized greater time on better understanding the curriculum (as expected due to lack of classroom implementation), all teachers (1) implemented responses that targeted student-centered learning design, (2) interpreted and evaluated student problem-solving strategies and integrated that interpretation into classroom responses, and (3) created group activities to support students communication about their developing problem-solving skills and knowledge. We believe this demonstrates the effectiveness of our dashboard in supporting both expert and novice teachers plan for the integration of a problem-based learning curriculum. We believe future work should explore the use of simulations such as these to increase teacher experience and comfort in dashboards that target not only performance, but students behaviors and problem-solving strategies as they complete such a complex curriculum.

We recognize limitations in our work. On the one hand, the low participation number for this study resulted in analyses focused on depth instead of breadth. Future work should increase the participant cohort to validate if these results hold and to better ensure that teacher preparation is inclusive and supports equity in future problem-based learning applications. In addition, in terms of the selection of classes for each simulation, we recognize a limitation in the use of a high- vs low-performing dichotomy in the selection of classes as that approach may not fully represent the nuances learning and problem solving behaviors from a classroom context. Future work in selecting data for simulations (and co-design) can look into more nuanced approaches to evaluating classes, groups within classes, and individual students. Finally, we aim to complete a full, iterative dashboard cycle in which the participating teachers will implement SPICE (supported by the accompanying RISE dashboard) in their classrooms, and then researcher-teacher partners will reflect on their simulation and classroom experiences.

CHAPTER 6

Discussion, Conclusions, and Future Directions

The empowerment and support of teachers as drivers of problem-based STEM learning and knowledge construction by computational modeling and problem solving among their students using advanced learning environments, while assuring learning objectives and standards are met, is essential for seamless integration of technology into classrooms, and for advancing instruction and assessment practices. This process is especially important as national and state standards now emphasize the need to integrate computing and engineering into K-12 classrooms. To facilitate successful empowerment, teachers need to be involved in the design and development of education technologies.

6.1 Contributions

This research includes one of the first instances of a co-designed teacher dashboard to support and prepare teachers for responsive teaching during technology-enhanced, problem-based learning in middle school science. The analyses leverage a novel system of assessments approach to understanding student learning and problem-solving behaviors in a student-centered, middle school STEM curriculum that combines the learning of science and engineering using a computational thinking framework. Using new prototyping techniques, teachers supported design improvements that targeted their preferences, needs, and concerns and improved our understanding of what constitutes actionable insight for PBL facilitation. Using this dashboard, experienced and inexperienced teachers enacted planning periods by reviewing automated feedback from past SPICE implementations to notice, interpret, and respond to students learning and problem solving, while also preparing teachers for such a classroom implementation. As part of this dissertation research, we have developed:

- *STEM Learning Environment*: providing an integrated approach and learning environment for science and engineering curriculum using computational thinking (CT),
- *Assessments*: improving, aligning, and evaluating integrated assessments using evidence-centered design that cover science, CT, and engineering concepts and practices over the course of the PBL curriculum,
- *Learning Analytics*: using artificial intelligence (AI) and machine learning (ML) methods to infer, from the assessments, student learning performance and behaviors, and the difficulties they face, taking into account teacher needs and requirements,

- *Co-Design Methods*: adapting established participatory design approaches, we have developed new kinds of prototyping methods to elicit and leverage teacher feedback for the co-construction of actionable insight for PBL curriculum facilitation,

- *Teacher Dashboard*: using this co-design approach we developed a teacher dashboard (RISE) to help middle school teachers with their noticing and response of student learning and problem-solving processes during STEM,

- *Evaluation Metrics*: using a mixed-method case study approach to systematically evaluate teachers' noticing and response to students learning and problem-solving processes during an integrated, middle school STEM curriculum, and

- *Responsive Teaching Preparation*: leveraging the dashboard as a resource for classroom data, we identified a novel method for evaluating and preparing teachers for noticing and response during computational modeling in science.

6.1.1 Understanding Student Learning and Problem Solving in Science, Computing, and Engineering

In the first manuscript, this dissertation presented a principled, design and implementation approach for the assessment of student learning and problem solving during a problem-based, technology-enhanced STEM curriculum. Using a system of assessments, we tracked students science, computing, and engineering learning over the course of the curriculum. In addition, leveraging student interaction data with the learning environment we are able to evaluate the impact of student problem-solving behaviors on learning in each domain.

Contributions of this work include:

- An NGSS-aligned curriculum integrating science, CT, and engineering,

- A system of assessments that support the tracking of science, CT, and engineering learning over the course of the intervention,

- Insight into the role of CT, from unplugged tasks to applications of important CT behaviors (e.g., debugging, testing, see Grover et al. (2016); Hutchins et al. (2021b); Ehsan et al. (2020)), on learning in science and engineering, and

- Applications of AI-based analytics to identify more nuanced problem-solving processes implemented during computing and engineering to support learning in the integrated domains.

These findings inspired low-fidelity prototypes of data visualizations on student learning and behaviors for the teacher co-design work. In addition, the identification of productive learning and problem-solving strategies on science, CT, and engineering learning, including the importance of learning through multiple, linked representations and systematic testing strategies with multiple variables (e.g., different rainfall and material values) motivated our approach and discussions with teachers on how to best target such practices.

6.1.2 Co-design Methods for Meaningful K-12 Teacher Contributions

The second manuscript detailed our novel co-design approaches supporting the creation of the Responsive Instruction for STEM Education (RISE) dashboard. Co-designing teacher-support technologies for the implementation of technology-enhanced, problem-based learning presents unique challenges, including how to represent complex learning analytics on open-ended, problem-solving strategies and knowledge construction across multiple domains in a way that is interpretable and actionable by the classroom teacher. This research adapted established co-design techniques for the purpose of eliciting meaningful feedback from experienced and inexperienced teachers for the development of a teacher dashboard to support PBL implementations in middle school science classroom. Findings and methods developed can also be applied to the development of teacher-support technology in other domains, at other grade-levels, and for different populations of teachers and students.

This work resulted in a number of key contributions. We detailed co-design techniques at multiple stages in the design process that (1) supported experienced and inexperienced teacher feedback, (2) elicited insight into the needs, values, concerns, and preferences of these teachers regarding what they need and how they may support students as they complete a technology-enhanced, problem-based learning curriculum, and (2) engaged researcher-teacher partners in reflection, discussion, and negotiations for how to facilitate PBL across multiple, linked representations. We provided key teacher insight themes regarding their dashboard needs to support teacher facilitation of PBL which can be leveraged by other designers and developers targeting this objective. In addition, we describe co-design recommendation based on our experience that support the systematic co-design of dashboards for PBL, including:

- Regularly link student results across multiple, linked domains,
- Immerse teachers, especially novice teachers, in the student experience prior to co-design to promote rich insight into visualizing student problem-solving processes, and
- Regularly reflect on instructional strategies at different social levels.

Finally, our research identified rich research agendas for evaluating teacher-AI teams for more complete, timely support of students during PBL, for using instructor-support technology to help teachers support students' emotional needs, and for co-designing explainable-AI that supports teacher understanding of the algorithms used and trust in the technology.

6.1.3 Teacher Support and Preparation for the Integration of Problem-Based, STEM Curricula

The third manuscript introduced a new classroom simulation method to better support our understanding of how teachers of different experiences use dashboards to support evidence-based pedagogical responses. Limited research has examined teacher dashboard usage (Campos et al., 2021), especially the pedagogical actions K-12 teachers take as a result of using instructor-support technology, such as dashboards (Wiley et al., 2020). To target this, this research presented Planning Period Simulations in which teachers leverage the RISE dashboard, equipped with class data and visualizations from prior SPICE implementations, to notice, interpret, and develop evidence-based lesson plan customizations based on class, group, and student performance.

This research extended the literature on teacher dashboard usage by contributing to a deeper understanding of how teachers use dashboard visualizations to conduct responsive teaching in PBL, including the evaluation processes implemented to create lesson plans that target the linkages between science, CT, and engineering (an insight found to be effective in promoting integrated student learning in the first manuscript). In addition, this research identified common processes, deemed *catalyzing links*, implemented by teachers to transition from their interpretation of learning analytics and data visualizations to evidence-based responses.

Finally, the Planning Period Simulations provided a novel way to prepare teachers for and to evaluate teachers noticing and response during technology-enhanced, PBL curricula such as computational modeling. This is particularly important as student ideation and problem-solving processes are implemented through interactions with the technology and are difficult to view and interpret. Moreover, the RISE dashboard was equipped with reflection and response tools that aided teachers during their reflections of class performance to more systematically transition from their performance interpretations to evidence-based pedagogical responses.

Evaluating teachers' evaluation processes not only contributes to our understanding of how data promotes changes in instruction (Farrell and Marsh, 2016), but it can:

- support the development of tools to aid in teachers' noticing and interpretation of students' problem-based learning in STEM,
- improve resources to support evidence-based pedagogical responses, and
- improve visualizations of feedback on students problem-based learning in STEM.

Moreover, this paper demonstrates the effectiveness of the RISE dashboard in supporting both experienced and inexperienced teachers as they plan for PBL implementations.

6.2 Limitations and Future Work

Limitations of the present work focus on the population sizes of the studies. In terms of the classroom implementation (n=99), we seek to increase our application of the SPICE curriculum to larger, more diverse populations of students to determine if our findings hold and to evaluate additional considerations we must make to ensure the social-relevance of and engagement in our curriculum. For the co-design methodologies, the relatively small (n=9) cohort focused our analysis on depth over breadth. We aim to conduct more extensive studies of teacher noticing using our technology to continue improving and adapting to the needs of diverse groups of teachers. These limitations are in addition to those described in each manuscript.

Finally, due to COVID we did not have opportunity to compete full cycle of design, development, and classroom implementation (see LATUX, Martinez-Maldonado et al., 2016). While we were able to get initial pilot data in terms of the Planning Period Simulations, we will conduct classroom studies with the dashboard to evaluate its impact and continue improving.

This dissertation also creates opportunities for future research in at least four directions: (1) online teacher support to engage teachers in student learning and problem solving, (2) developing teaching assistant agents, (3) improving teacher training for the integration of computing and engineering in K-12 science, and (4) addressing bias and promoting equity in the design and application of classroom support technologies.

6.2.1 Online Teacher Support to Engage Teachers in Student Learning and Problem-solving

The RISE dashboard demonstrates an effective implementation for visualizing learning analytics to support teachers' evidence-based lesson plan customizations, but more work needs to be done to move this work online. First, we plan to conduct classroom implementations with RISE to more deeply explore teachers' responsive PBL teaching strategies as they occur in classrooms. In the future, this may also require additional software design and development. For example, although the majority of teachers requested class reports, all teachers also recommended the use of alerts (e.g., on a phone or iPad) to inform them about student successes and opportunities during PBL in-the-moment.

A significant portion of our work, and common to STEM classrooms, involves student written responses to prompts to explain their developing scientific ideas. Initial work is underway, including innovations such as the Teacher Action Planner that provides insight into students developing ideas as they complete web-based inquiry tasks (Gerard et al., 2020). In our work, we have made advancements in the evaluation of students' causal reasoning as they complete science formative assessments in our SPICE curriculum (Cochran et al., 2022). These approaches, including the on-demand delivery of results to teachers, can support more meaningful, timely engagement in students' developing ideas and problem-solving processes.

6.2.2 Teaching Assistant Agents

An interesting feedback resulting from the codesign sessions (Manuscript Two) was the request for the learning environment to provide some in-the-moment feedback to students. Teachers recognized that AI can more easily provide specific types of in-the-moment needs (e.g., off-task alerts, concept knowledge recommendations), especially when those feedback tasks may be time-consuming for the teacher to implement in the context of a busy, PBL classroom.

As such, future research could explore the creation of teacher-AI teams to support problem-based learning in STEM. The concept of teacher-AI teams has been explored to support teachers in the integration of Intelligant Tutoring Systems (e.g., Holstein et al., 2019); however, PBL approaches are prone to unique challenges in the context of open-ended, problem solving. For example considerations must be made for (1) supporting teacher agency (a key need identified in Manuscript Two) in deciding what situations can or should be supported by the AI, (2) ensuring the AI facilitates the student-centered nature of the learning that is occuring and preventing "lethal mutations" (Brown and Campione, 1996) in the intended design, (3) building teacher trust and confidence in the AI-agent, and (4) promoting equity and eliminating bias in the application of this new AI tool.

6.2.3 Improving Teacher Training for the Integration of Computing and Engineering in K-12 Science

The Planning Period Simulations offered a unique opportunity for inexperienced teachers to engage with the PBL curriculum and explore how students learn and problem solve in SPICE. However, more work can be done to support PBL teacher training and professional development. For RISE, we plan to implement more studies using these simulations to further explore responsive teaching practices and ways to improve our teacher-feedback tool. This work will also allow us to develop a database of evidence-based response ideas that inexperienced teachers can leverage to help them in preparing for and implementing this PBL curriculum in their classrooms. For instance, this may involve providing new SPICE teachers with example expert teacher responses following each simulation as a tool to increase reflection and knowledge of the SPICE curriculum, and, in particular, the importance of the multiple, linked representations.

Recent work has leveraged video clubs as tools for teacher professional development, particularly for responsive teaching in STEM (e.g., Johnson and Forsythe, 2015). While the Planning Period Simulations provided a novel approach for training and for evaluating how and what teachers notice from RISE, in that the multiple sources of data were specifically co-designed to support key performance and strategy data teachers would be interested in or would leverage in their lesson plan decision making, video club research can be leveraged to improve our approach. For instance, while the data visualizations served as the boundary object supporting discussions between teacher and researcher (as opposed to classroom video), we aim to extend this work by:

- conducting simulations with pairs or groups of teachers so they can leverage each others' prior domain and pedagogical content knowledge to discuss and reflect on student, group, and class results, and
- collecting classroom video data in which teachers leverage RISE to have an additional resource for how example responses are enacted in a real classroom context and to promote further discussions.

We hypothesize that these approaches will enhance teacher training for (1) problem-based learning approaches that leverage technology-enhanced learning environments, especially computational modeling and engineering design in science and (2) supporting teachers with the complex task of integrating science, computing and/or engineering in their K-12 science classrooms.

6.2.4 Addressing Bias and Promoting Equity in the Design and Application of Classroom Support Technologies

Future work must further our understanding of how we can address bias and promote equity in the application of AI-backed technologies in education. Although not directly targeted in this dissertation, these issues are particularly concerning as we need to create educational technology and opportunities for all students, as well as support teachers as they engage their students in these complex curricular approaches. Research could target:

1. how might we address bias in the feedback presented to teachers and/or the interpretation of that feedback in the wild?,
2. how can we explicitly think about developing dashboards that promote equity in their applications?,
3. what do increased data needs mean in terms of surveillance in the classroom?, and, further,
4. what dangers exist from the tools we provide if they are misused?

Further work on the development of design principles is needed to help us better address or be prepared for this issues.

For example, in the first manuscript, path analysis was used to evaluate student learning across multiple, linked representations and clustering analysis was used to group students and evaluate the impact of problem-solving processes on learning in each domain. However, during these processes, outliers (students) were removed for the purpose of the analysis. In the context of providing evidence-based feedback, this may impact the type of and whether such students receive feedback. Future work may look into such impacts and how they may be addressed.

In recent work, researchers identified that AI-based personalization impacted users' conceptualization of self, including how these algorithms inform users' understanding of their identities and their relationship to others (Lee et al., 2022). With the increased introduction of AI-backed technology in the classroom, there is a potential that the characterization of students and the accompanying personalization of feedback may impact students' conceptualization of their student self - from their identities as learners, their relationship with the complex STEM domains, and how they see themselves as compared to their classmates. Future work should examine these impacts to better support students (and their teachers) to meaningfully engage all students in learning and problem solving that prepares them for an enriched future.

References

Ahmed, S., Wallace, K. M., and Blessing, L. T. (2003). Understanding the differences between how novice and experienced designers approach design tasks. *Research in Engineering Design*, 14(1):1–11.

Ahn, J., Nguyen, H., and Campos, F. (2021). From visible to understandable: Designing for teacher agency in education data visualizations. *Contemporary Issues in Technology and Teacher Education*.

Ainsworth, S. (2006). Deft: A conceptual framework for considering learning with multiple representations. *Learning and Instruction*, 16(3):183–198.

Alavi, H. S. and Dillenbourg, P. (2012). An ambient awareness tool for supporting supervised collaborative problem solving. *IEEE Transactions on Learning Technologies*, 5(3):264–274.

Aleven, V., Roll, I., McLaren, B. M., and Koedinger, K. (2015). Help helps, but only so much: Research on help seeking with intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 26:205–223.

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*, 29.

Araujo, I. S., Veit, E. A., and Moreira, M. A. (2008). Physics students' performance using computational modelling activities to improve kinematics graphs interpretation. *Computers Education*, 50(4):1128–1140.

Atman, C. J., Kilgore, D., and McKenna, A. (2008). Characterizing design learning: A mixed-methods study of engineering designers' use of language. *Journal of Engineering Education*, 97(3):309–326.

Azevedo, R., Johnson, A., Chauncey, A., and Burkett, C. (2010). Self-regulated learning with metatutor: Advancing the science of learning with metacognitive tools. In *New science of learning*, pages 225–247. Springer.

Baker, R., Xu, D., Park, J., Yu, R., Li, Q., Cung, B., Fischer, C., Rodriguez, F., Warschauer, M., and Smyth, P. (2020). The benefits and caveats of using clickstream data to understand student self-regulatory behaviors: opening the black box of learning processes. *The Elementary School Journal*, 17.

Bakharia, A., Corrin, L., de Barba, P., Kennedy, G., Gašević, D., Mulder, R., Williams, D., Dawson, S., and Lockyer, L. (2016). A conceptual framework linking learning design with learning analytics. In *Proceedings of the Sixth International Conference on Learning Analytics Knowledge*, LAK '16, page 329–338, New York, NY, USA. Association for Computing Machinery.

Ball, D. L., Hill, H. C., and Bass, H. (2005). Knowing mathematics for teaching: Who knows mathematics well enough to teach third grade, and how can we decide? *American Educator*, 29(1):14–17, 20–22, 43–46.

Ball, D. L., Sleep, L., Boerst, T. A., and Bass, H. (2009). Combining the development of practice and the practice of development in teacher education. *The Elementary School Journal*, 109(5):458–474.

Barab, S. and Squire, K. (2004). Design-based research: Putting a stake in the ground. *Journal of the Learning Sciences*, 13(1):1–14.

Barab, S. A. and Luehmann, A. L. (2003). Building sustainable science curriculum: Acknowledging and accommodating local adaptation. *Science Education*, 87(4):454–467.

Barnhart, T. and van Es, E. (2015). Studying teacher noticing: Examining the relationship among preservice science teachers' ability to attend, analyze and respond to student thinking. *Teaching and Teacher Education*, 45:83–93.

Basu, S., Biswas, G., and Kinnebrew, J. S. (2016a). Using multiple representations to simultaneously learn computational thinking and middle school science. In *AAAI*, pages 3705–3711.

Basu, S., Biswas, G., and Kinnebrew, J. S. (2017). Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment. *User Modeling and User-Adapted Interaction*, 27(1):5–53.

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., and Clark, D. (2016b). Identifying middle school students' challenges in computational thinking-based science learning. *Research and practice in technology enhanced learning*, 11(1):13.

Basu, S., Dickes, A., Kinnebrew, J. S., Sengupta, P., and Biswas, G. (2013). Ctsim: A computational thinking environment for learning science through simulation and modeling. In *CSEDU*, pages 369–378.

Basu, S., Dukeman, A., Kinnebrew, J. S., Biswas, G., and Sengupta, P. (2014). Investigating student generated computational models of science. Boulder, CO: International Society of the Learning Sciences.

Basu, S., McElhaney, K., Grover, S., Harris, C., and Biswas, G. (2018). A principled approach to designing assessments that integrate science and computational thinking. In *Proceedings of the 13th International Conference of the Learning Sciences (ICLS)*, pages 384–391. International Society of the Learning Sciences.

Basu, S., McElhaney, K., Rachmatullah, A., Hutchins, N. M., Biswas, G., and Chiu, J. (2022). Promoting computational thinking through science-engineering integration using computational modeling. In Chinn, C., T. E. C. C. . K. Y., editor, *16th International Conference of the Learning Sciences – ICLS2022*. International Society of the Learning Sciences (ISLS).

Basu, S., Rutstein, D. W., Xu, Y., Wang, H., and Shear, L. (2021). A principled approach to designing computational thinking concepts and practices assessments for upper elementary grades. *Computer Science Education*, 0(0):1–30.

Baumer, E. P. (2017). Toward human-centered algorithm design. *Big Data & Society*, 4(2):2053951717718854.

Baxter, I. D., Yahin, A., Moura, L., Sant'Anna, M., and Bier, L. (1998). Clone detection using abstract syntax trees. In *Proceedings. International Conference on Software Maintenance (Cat. No. 98CB36272)*, pages 368–377. IEEE.

Benoit, G., Slama, R., Moussapour, R. M., Reich, J., and Anderson, N. (2021). Simulating more equitable discussions: using teacher moments and practice based teacher education in mathematical professional learning.

Bienkowski, M., Snow, E., Rutstein, D., and Grover, S. (2015). Assessment design patterns for computational thinking practices in secondary computer science: A first look. Technical report.

Bille, P. (2005). A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1):217–239.

Biswas, G., Leelawong, K., Schwartz, D., Vye, N., and The Teachable Agents Group at Vanderbilt (2005). Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence*, 19(3-4):363–392.

Biswas, G., Segedy, J. R., and Bunchongchit, K. (2016). From design to implementation to practice a learning by teaching system: Betty's brain. *International Journal of Artificial Intelligence in Education*, 26(1):350–364.

Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S., and Koller, D. (2014). Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming. *Journal of the Learning Sciences*, 23(4):561–599.

Blumenfeld, P., Fishman, B. J., Krajcik, J., Marx, R. W., and Soloway, E. (2000). Creating usable innovations in systemic reform: Scaling up technology-embedded project-based science in urban schools. *Educational Psychologist*, 35(3):149–164.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., and Punie, Y. (2016). Developing computational thinking in compulsory education - implications for policy and practice. Technical report.

Bolger, M. S., Kobiela, M., Weinberg, P. J., and Lehrer, R. (2012). Children's mechanistic reasoning. *Cognition and Instruction*, 30(2):170–206.

Boschman, F., Susan, M., and Voogt, J. (2014). Understanding decision making in teachers curriculum design approaches. *Educational technology research and development*, 62(4):393–416.

Bransford, J. D., Brown, A., and Cocking, R. R. (2000). *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. The National Academies Press, Washington, DC.

Bredeweg, B., Liem, J., Beek, W., Linnebank, F., Gracia, J., Lozano, E., Wissner, M., Bühling, R., Salles, P., Noble, R., Zitek, A., Borisova, P., and Mioduser, D. (2013). Dynalearn - an intelligent learning environment for learning conceptual knowledge. *AI Mag.*, 34:9–.

Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the Annual Meeting of the American Educational Research Association*. AERA.

Bressler, D., Bodzin, A., Eagan, B., and Tabatabai, S. (2019). Using epistemic network analysis to examine discourse and scientific practice during a collaborative game. *Journal of Science Education and Technology*, 28.

Broll, B. (2018). *Collaborative Educational Environment Design for Accessible Distributed Computing*. PhD thesis, Vanderbilt University.

Brown, A. L. and Campione, J. C. (1996). *Psychological theory and the design of innovative learning environments: On procedures, principles, and systems.*, pages 289–325. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, US.

Bucciarelli, L. (2003). *Engineering philosophy*. DUP Satellite.

Bywater, J. P., Chiu, J. L., Hong, J., and Sankaranarayanan, V. (2019). The teacher responding tool: Scaffolding the teacher practice of responding to student ideas in mathematics classrooms. *Computers Education*, 139:16–30.

Bywater, J. P., Floryan, M., and Chiu, J. L. (2021). Discs: A new sequence segmentation method for openended learning environments. In *International Conference on Artificial Intelligence in Education*. Springer.

Campos, F., Ahn, J., DiGiacomo, D. K., Nguyen, H., and Hays, M. (2021). Making sense of sensemaking: Understanding how k–12 teachers and coaches react to visual analytics. *Journal of Learning Analytics*, 8(3):60–80.

Carrol, J. (1999). Five reasons for scenario-based design. In *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32. Abstracts and CD-ROM of Full Papers*, volume Track3, pages 11 pp.–.

CCSSO (2011). The common core state standards for mathematics. Available at http://www.corestandards.org/Math/ (2020/03/12).

Chang, C.-J., Liu, C.-C., Wen, C.-T., Tseng, L.-W., Chang, H.-Y., Chang, M.-H., Fan Chiang, S.-H., Hwang, F.-K., and Yang, C.-W. (2020). The impact of light-weight inquiry with computer simulations on science learning in classrooms. *Computers Education*, 146:103770.

Charmaz, K. (2006). *Constructing grounded theory: A practical guide through qualitative analysis*. Sage.

Charters, E. (2003). The use of think-aloud methods in qualitative research an introduction to think-aloud methods. *Brock Education Journal*, 12.

Chen, Y., Hmelo-Silver, C., Lajoie, S., Zheng, J., Huang, L., and Bodnar, S. (2021). Using teacher dashboards to assess group collaboration in problem-based learning. *Educational Technology Research and Development*, 15(2).

Chi, M. T. H. (2008). Three types of conceptual change: Belief revision, mental model transformation, and categorical shift. In Vosniadou, S., editor, *Handbook of research on conceptual change*, pages 61—-82. Erlbaum, Hillsdale, NJ, USA.

Chiu, J., McElhaney, K., Zhang, N., Biswas, G., Fried, R., Basu, S., Alozie, N., and Hong, J. (2019). A principled approach to ngss-aligned curriculum development integrating science, engineering, and computation: A pilot study. In *NARST Annual International Conference*. NARST.

Chiu, J. L., Fick, S. J., McElhaney, K. W., Alozie, N., and Fujii, R. (2021). Elementary teacher adaptations to engineering curricula to leverage student and community resources. *Journal of Pre-College Engineering Education Research (J-PEER)*, 11(1).

Cober, R., Tan, E., Slotta, J., So, H.-J., and Konings, K. (2015). Teachers as participatory designers: two case studies with technology-enhanced learning environments. *Instructional Science*, 43(2):203–228.

Cochran, K., Cohn, C., Hutchins, N., Biswas, G., and Hastings, P. (2022). Improving automated evaluation of formative assessments with text data augmentation. In *23rd International Conference on Artificial Intelligence in Education*. International AIED Society.

Coffey, J. E., Hammer, D., Levin, D. M., and Grant, T. (2011). The missing disciplinary substance of formative assessment. *Journal of Research in Science Teaching*, 48(10):1109–1136.

Colella, V. S., Klopfer, E., and Resnick, M. (2001). *Adventures in Modeling: Exploring Complex, Dynamic Systems with Starlogo*. Teachers College Press, USA, 1st edition.

Conley, D. T. and Darling-Hammond, L. (2013). Creating systems of assessment for deeper learning. Retrieved from http://scee.groupsite.com/uploads/files/x/000/09e/76f/creatingsystems-assessment-deeper-learning.pdf (2021/04/01).

Cook-Sather, A. (2014). Multiplying perspectives and improving practice: What can happen when undergraduate students collaborate with college faculty to explore teaching and learning. *Instructional Science*, 42.

Corrin, L., De Barba, P., Lockyear, L., Gašević, D., Williams, D., Dawson, S., Mulder, R., Copeland, S., and Bakharia, A. (2016). *Completing the Loop: Returning Meaningful Learning Analytic Data to Teachers*. Australian Government Office for Learning and Teaching, Australia.

Cowie, B. and Bell, B. (1999). A model of formative assessment in science education. *Assessment in Education: Principles, Policy & Practice*, 6(1):101–116.

Crismond, D. P. and Adams, R. S. (2012). The informed design teaching and learning matrix. *Journal of Engineering Education*, 101(4):738–797.

Csanadi, A., Eagan, B. R., Kollar, I., Shaffer, D. W., and Fischer, F. (2018). When coding-and-counting is not enough: using epistemic network analysis (ena) to analyze verbal data in cscl research. *International Journal of Computer-Supported Collaborative Learning*, 13:419–438.

Cuban, L. (1984). *How Teachers Taught: Constancy and Change in American Classrooms, 1890-1980*. Research on Teaching Monograph Series. Longman, New York NY.

Cunningham, C., Knight, M., Carlsen, W., and Kelly, G. (2007). Integrating engineering in middle and high school classrooms. *International Journal of Engineering Education*, 23(1):3–8.

Cunningham, C. M. (2008). Elementary teacher professional development in engineering: lessons learned from engineering is elementary. In *Paper presented at the National Academy of Engineering Annual Meeting*. The National Academy of Engineering.

Cunningham, C. M. and Carlsen, W. S. (2014). Teaching engineering practices. *Journal of Science Teacher Education*, 25(2):197–210.

Cunningham, C. M. and Kelly, G. J. (2017). Epistemic practices of engineering for education. *Science Education*, 101(3):486–505.

Cuny, J. (2012). Transforming high school computing: A call to action. *ACM Inroads*, 3(2):32–36.

Dawson, S., Gašević, D., Siemens, G., and Joksimovic, S. (2014). Current state and future trends: A citation network analysis of the learning analytics field. In *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*, LAK '14, page 231–240, New York, NY, USA. Association for Computing Machinery.

de Figueiredo, A. D. (2008). Toward an epistemology of engineering. In *2008 Workshop on Philosophy and Engineering*. The Royal Academy of Engineering.

Dede, C. (2010). Technological supports for acquiring twenty-first-century skills. In Peterson, P., Baker, E., and McGaw, B., editors, *International Encyclopedia of Education (Third Edition)*, pages 158–166. Elsevier, Oxford, third edition edition.

Diana, N., Eagle, M., Stamper, J., Grover, S., Bienkowski, M., and Basu, S. (2017). An instructor dashboard for real-time analytics in interactive programming assignments. In *Proceedings of the Seventh International Learning Analytics Knowledge Conference*, LAK '17, page 272–279, New York, NY, USA. Association for Computing Machinery.

Dickler, R., Gobert, J., and Sao Pedro, M. (2021). Using innovative methods to explore the potential of an alerting dashboard for science inquiry. *Journal of Learning Analytics*, 8(2):105–122.

Dillenbourg, P. (2015). *Orchestration Graphs: Modeling Scalable Education*. Taylor Francis Group.

DiSalvo, B. and DiSalvo, C. (2014). Designing for democracy in education: Participatory design and the learning sciences. In Joseph L. Polman, Eleni A. Kyza, D. K. O. I. T. W. R. P. A. S. J. K. O. T. L. and D'Amico, L., editors, *Tearning and Becoming in Practice: The International Conference of the Learning Sciences (ICLS) 2014.*, pages 793–799. International Society of the Learning Sciences (ISLS).

DiSalvo, B., Yip, J., Bonsignore, E., and DiSalvo, C. (2017). *Participatory Design for Learning*, pages 15–18. Routledge.

diSessa, A. (2001). *Changing minds: Computers, learning, and literacy*. MIT Press, Cambridge MA USA.

Doran, D., Schulz, S., and Besold, T. R. (2017). What does explainable AI really mean? A new conceptualization of perspectives. *CoRR*, abs/1710.00794.

Dyckhoff, A. L., Zielke, D., Bültmann, M., Chatti, M. A., and Schroeder, U. (2012). Design and implementation of a learning analytics toolkit for teachers. *Journal of Educational Technology Society*, 15(3):58–76.

Echeverria, V., Martinez-Maldonado, R., Buckingham Shum, S., Chiluiza, K., Granda, R., and Conati, C. (2018). Exploratory versus explanatory visual learning analytics: Driving teachers' attention through educational data storytelling. *Journal of Learning Analytics*, 5(3):72—97.

Ehsan, H., Rehmat, A. P., and Cardella, M. E. (2020). Computational thinking embedded in engineering design: capturing computational thinking of children in an informal engineering design activity. *International Journal of Technology and Design Education*.

Ellis, B. and Wong, W. H. (2008). Learning causal bayesian network structures from experimental data. *Journal of the American Statistical Association*, 103(482):778–789.

Emara, M., Grover, S., Hutchins, N., Biswas, G., and Snyder, C. (2020). Examining students' debugging and regulation processes during collaborative computational modeling in science. In *The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020*, pages 1325–1332. International Society of the Learning Sciences (ISLS).

Emara, M., Hutchins, N., Grover, S., Snyder, C., and Biswas, G. (2021). Examining student regulation of collaborative, computational, problem-solving processes in open-ended learning environments. *Journal of Learning Analytics*, 8(1):49–74.

Empson, S. B. and Jacobs, V. R. (2008). Learning to listen to children's mathematics. In Tirosh, D. and Wood, T., editors, *The International Handbook of Mathematics Teacher Education, Volume 2: Tools and Processes in Mathematics Teacher Education*, pages 257–281. Sense Publishers, The Netherlands.

English, L. and King, D. (2017). Engineering education with fourth-grade students: Introducing design-based problem solving. *International Journal of Engineering Education*, 33(1):346–360.

Farrell, C. C. and Marsh, J. A. (2016). Contributing conditions: A qualitative comparative analysis of teachers' instructional responses to data. *Teaching and Teacher Education*, 60:398–412.

Ferris, T. L. J. (2012). Engineering design as research. In M. Mora, O. Gelman, A. S. M. R., editor, *Research methodologies, innovations and philosophies in software systems engineering and information systems*, page 389–402. IGI Global, Hershey PA.

Fischer, C., Pardos, Z. A., Baker, R. S., Williams, J. J., Smyth, P., Yu, R., Slater, S., Baker, R., and Warschauer, M. (2020). Mining big data in education: Affordances and challenges. *Review of Research in Education*, 44(1):130–160.

Fischer, F., Kollar, I., Stegmann, K., and Wecker, C. (2013). Toward a script theory of guidance in computer-supported collaborative learning. *Educational Psychologist*, 48(1):56–66.

Frederiksen, J. R., White, B. Y., and Gutwill, J. (1999). Dynamic mental models in learning science: The importance of constructing derivational linkages among models. *Journal of Research in Science Teaching*, 36(7):806–836.

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., and Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23):8410–8415.

Friedman, B., Kahn, P., and Borning, A. (2002). Value sensitive design: Theory and methods. *University of Washington technical report*, 2:12.

Furtak, E. M. and Heredia, S. C. (2014). Exploring the influence of learning progressions in two teacher communities. *Journal of Research in Science Teaching*, 51(8):982–1020.

Gašević, D., Dawson, S., and Siemens, G. (2015). Let's not forget: Learning analytics are about learning. *TechTrends*, 59(1):64–71.

Gerard, L., Wiley, K., Bradford, A., Chen, J. K., Lim-Breitbart, J., and Linn, M. (2020). Impact of a teacher action planner that captures student ideas on teacher customization decisions. In *The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020*. International Society of the Learning Sciences (ISLS).

Gobert, J. D., Pedro, M. S., Raziuddin, J., and Baker, R. S. (2013). From log files to assessment metrics: Measuring students' science inquiry skills using educational data mining. *Journal of the Learning Sciences*, 22(4):521–563.

Gomoll, A., Hmelo-Silver, C. E., and Šabanović, S. (2022). Co-constructing professional vision: Teacher and researcher learning in co-design. *Cognition and Instruction*, 40(1):7–26.

Graesser, A. C., Conley, M. W., and Olney, A. (2012). *Intelligent tutoring systems.*, pages 451—-473. American Psychological Association, Washington, DC, US.

Grosch, J. and Emmelmann, H. (1990). A tool box for compiler construction. In *International Workshop on Compiler Construction*, pages 106–116. Springer.

Grover, S. and Basu, S. (2017). Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '17, page 267–272, New York, NY, USA. Association for Computing Machinery.

Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., and Stamper, J. (2017). A framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments. *ACM Trans. Comput. Educ.*, 17(3).

Grover, S., Basu, S., and Schank, P. (2018a). What we can learn about student learning from open-ended programming projects in middle school computer science. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, page 999–1004, New York, NY, USA. Association for Computing Machinery.

Grover, S., Basu, S., and Schank, P. (2018b). What we can learn about student learning from open-ended programming projects in middle school computer science. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, page 999–1004, New York, NY, USA. Association for Computing Machinery.

Grover, S., Bienkowski, M., Niekrasz, J., and Hauswirth, M. (2016). Assessing problem-solving process at scale. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*, L@S '16, page 245–248, New York, NY, USA. Association for Computing Machinery.

Grover, S., Cooper, S., and Pea, R. (2014). Assessing computational learning in k-12. In *Proceedings of the 2014 Conference on Innovation Technology in Computer Science Education*, ITiCSE '14, page 57–62, New York, NY, USA. Association for Computing Machinery.

Grover, S. and Pea, R. (2013). Computational thinking in k–12: A review of the state of the field. *Educational Researcher*, 42(1):38–43.

Grover, S. and Pea, R. (2018). *Computational Thinking: A Competency Whose Time Has Come*. Bloomsbury.

Grover, S., Pea, R., and Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2):199–237.

Hagger, M. S. and Hamilton, K. (2018). Motivational predictors of students' participation in out-of-school learning activities and academic attainment in science: An application of the trans-contextual model using bayesian path analysis. *Learning and Individual Differences*, 67:232–244.

Haklev, S., Faucon, L. P., Hadzilacos, T., and Dillenbourg, P. (2017). Frog: rapid prototyping of collaborative learning scenarios. *EC-TEL*.

Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., and Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. *SIGCSE Bull.*, 41(1):183–187.

Hammer, D. (1997). Discovery learning and discovery teaching. *Cognition and Instruction*, 15(4):485–529.

Hammer, D., Goldberg, F., and Fargason, S. (2012). Responsive teaching and the beginnings of energy in a third grade classroom. *Review of Science, Mathematics and ICT Education*, 6(1):51—-72.

Hammer, D. and van Zee, E. (2006). *Seeing the science in children's thinking: Case studies of student inquiry in physical science*. The National Academies Press.

Hannafin, M., Land, S., and Oliver, K. (1999). Open learning environments: Foundations, methods, and models. *Instructional-design theories and models: A new paradigm of instructional theory*, 2:115–140.

Hannafin, M. J., Hill, J. R., Land, S. M., and Lee, E. (2014). *Student-Centered, Open Learning Environments: Research, Theory, and Practice*, chapter 51, pages 641–651. Springer New York, New York, NY.

Hashem, K. and Mioduser, D. (2011). The contribution of learning by modeling (lbm) to students' understanding of complexity concepts. *International Journal of e-Education, e-Business, e-Management and e-Learning*, 1(2):151–157.

Hatch, J. A. (2002). *Doing qualitative research in education settings*. SUNY Press.

Hernández-Leo, D., Martinez-Maldonado, R., Pardo, A., Muñoz-Cristóbal, J. A., and Rodríguez-Triana, M. J. (2019). Analytics for learning design: A layered framework and tools. *British Journal of Educational Technology*, 50(1):139–152.

Hirsch, L., Berliner-Heyman, S. L., Carpinelli, J., and Kimmel, H. (2012). Introducing middle school students to engineering and the engineering design process. In *Proceedings of the American Society for Engineering Education Annual Conference Exposition*. Association Society for Engineering Education.

Hirsch, L. S., Carpinelli, J. D., Kimmel, H., Rockland, R., and Bloom, J. (2007). The differential effects of pre-engineering curricula on middle school students' attitudes to and knowledge of engineering careers. In *2007 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports*, pages S2B–17–S2B–21.

Hmelo-Silver, C. and Barrows, H. (2015). *Problem-based learning: Goals for learning and strategies for facilitating*, pages 69–84. Purdue University Press, West Lafayette, IN.

Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? *Educational Psychology Review*, 16(3):235–266.

Hoadley, C. (2002). Creating context: Design-based research in creating and understanding cscl. In *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*, pages 453–462.

Holstein, K., McLaren, B. M., and Aleven, V. (2017). Intelligent tutors as teachers' aides: Exploring teacher needs for real-time analytics in blended classrooms. In *Proceedings of the Seventh International Learning Analytics Knowledge Conference*, LAK '17, page 257–266, New York, NY, USA. Association for Computing Machinery.

Holstein, K., McLaren, B. M., and Aleven, V. (2019). Co-designing a real-time classroom orchestration tool to support teacher–ai complementarity. *Journal of Learning Analytics*, 6(2):27–52.

Horn, I. S. and Little, J. W. (2010). Attending to problems of practice: Routines and resources for professional learning in teachers' workplace interactions. *American Educational Research Journal*, 47(1):181–217.

Hsu, M., Cardella, M., and Purzer, (2014). Assessing design. In Ş. Purzer, M. E. Cardella, . J. S., editor, *Engineering in pre-college settings: Synthesizing research, policy, and practices*, page 303–313. Purdue University Press, West Lafayette, IN.

Hsu, T.-C., Chang, S.-C., and Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers Education*, 126:296–310.

Hutchins, N. and Biswas, G. (2022). Teacher noticing and response to students' computational and engineering design strategies. In *American Educational Research Association 2022 Symposium on AI and the Future of STEM Instruction: Designing New Models to Automate Feedback to Teachers*.

Hutchins, N., Biswas, G., Conlin, L., Emara, M., Grover, S., Basu, S., and McElhaney, K. (2018). Studying synergistic learning of physics and computational thinking in a learning by modeling environment. In *Proceedings of the 26th International Conference on Computers in Education (ICCE)*, pages 153–162.

Hutchins, N., Biswas, G., Grover, S., Basu, S., and Snyder, C. (2019a). A systematic approach for analyzing students' computational modeling processes in c2stem. In Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., and Luckin, R., editors, *Artificial Intelligence in Education*, pages 116–121, Cham. Springer International Publishing.

Hutchins, N., Biswas, G., Maróti, M., Lédeczi, Á., Grover, S., Wolf, R., Blair, K. P., Chin, D., Conlin, L., Basu, S., et al. (2020a). C2stem: a system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology*, 29(1):83–100.

Hutchins, N., Biswas, G., Zhang, N., Snyder, C., Lédeczi, Á., and Maróti, M. (2020b). Domain-specific modeling languages in computer-based learning environments: a systematic approach to support science learning through computational modeling. *International Journal of Artificial Intelligence in Education*, 30(4):537–580.

Hutchins, N., Shi, C., and Biswas, G. (2019b). A high school computational modeling approach to studying the effects of climate change on coral reefs. In *The American Educational Research Association Annual Meeting*.

Hutchins, N. M., Basu, S., McElhaney, K., Chiu, J., Fick, S., Zhang, N., and Biswas, G. (2021a). Coherence across conceptual and computational representations of students' scientific models. In *The International Society of the Learning Sciences Annual Meeting 2021*. International Society of the Learning Sciences (ISLS).

Hutchins, N. M., Biswas, G., Wolf, R., Chin, D., Grover, S., and Blair, K. P. (2020c). Computational thinking in support of learning and transfer. In *The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020*, pages 1405–1412. International Society of the Learning Sciences (ISLS).

Hutchins, N. M., Snyder, Caitlin, E. M., Grover, S., and Biswas, G. (2021b). Analyzing debugging processes during collaborative, computational modeling in science. In *The International Society of the Learning Sciences Annual Meeting 2021*. International Society of the Learning Sciences (ISLS).

Irgens, G. A., Dabholkar, S., Bain, C., Woods, P., Hall, K. C., Swanson, H., Horn, M. S., and Wilensky, U. (2020). Modeling and measuring high school students' computational thinking practices in science. *Journal of Science Education and Technology*, 29:137–161.

Jacobs, J. and Morita, E. (2002). Japanese and american teachers' evaluations of videotaped mathematics lessons. *Journal for Research in Mathematics Education*, 33(3):154–175.

Jacobs, V. R., Lamb, L. L. C., and Philipp, R. A. (2010). Professional noticing of children's mathematical thinking. *Journal for Research in Mathematics Education*, 41(2):169–202.

Jiménez-Aleixandre, M. P., Bugallo Rodríguez, A., and Duschl, R. A. (2000). "doing the lesson" or "doing science": Argument in high school genetics. *Science Education*, 84(6):757–792.

Johnson, A. W., Wendell, K. B., and Watkins, J. (2017). Examining experienced teachers' noticing of and responses to students' engineering. *Journal of Pre-College Engineering Education Research (J-PEER)*, 7(1).

Johnson, H. and Forsythe, M. (2015). Developing preservice teachers' knowledge of science teaching through video clubs. *Journal of Science Teacher Education*, 26:393–417.

Johnson, H. J. and Mawyer, K. K. N. (2019). Teacher candidate tool-supported video analysis of students' science thinking. *Journal of Science Teacher Education*, 30(5):528–547.

Jona, K., Wilensky, U., Trouille, L., Horn, M., Orton, K., Weintrop, D., and Beheshti, E. (2014). Embedding computational thinking in science, technology, engineering, and math (ct-stem). In *In future directions in computer science education summit meeting*.

Jonassen, D., Strobel, J., and Gottdenker, J. (2005). Model building for conceptual change. *Interactive Learning Environments*, 13(1-2):15–37.

Kali, Y., McKenney, S., and Sagy, O. (2015). Teachers as designers of technology enhanced learning. *Instructional Science*, 43(2):173–179.

Keating, T., Barnett, M., Barab, S. A., and Hay, K. E. (2002). The virtual solar system project: Developing conceptual understanding of astronomical concepts through building three-dimensional computational models. *Journal of Science Education and Technology*, 11(3):261–275.

Khribi, M. K., Jemni, M., and Nasraoui, O. (2015). *Recommendation Systems for Personalized Technology-Enhanced Learning*, pages 159–180. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kim, M. C. and Hannafin, M. J. (2011). Scaffolding problem solving in technology-enhanced learning environments (teles): Bridging research and theory with practice. *Computers Education*, 56(2):403–417.

Kinnebrew, J. S., Loretz, K. M., and Biswas, G. (2013). A contextualized, differential sequence mining method to derive students' learning behavior patterns. *JEDM— Journal of Educational Data Mining*, 5(1):190–219.

Kirschner, P. A. and van Merriënboer, J. J. (2013). Do learners really know best? urban legends in education. *Educational Psychologist*, 48(3):169–183.

Kline, R. B. (2015). *Principles and practice of structural equation modeling*. Guilford publications.

Kokoska, S. and Zwillinger, D. (2000). *CRC standard probability and statistics tables and formulae*. CRC Press.

Kolodner, J., Crismond, C., Gray, J., Holbrook, J., and Puntambekar, S. (1998). Learning by design from theory to practice. In *Proceedings of the International Conference of the Learning Sciences (ICLS) 2006*, pages 16–22. International Society of the Learning Sciences.

Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., Puntambekar, S., and Ryan, M. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting learning by design™ into practice. *Journal of the Learning Sciences*, 12(4):495–547.

Käser, T. and Schwartz, D. L. (2020). Modeling and analyzing inquiry strategies in open-ended learning environments. *International Journal of Artificial Intelligence in Education*, 30(3):504–535.

Könings, K., Seidel, T., and Van Merrienboer, J. J. G. (2014). Participatory design of learning environments: Integrating perspectives of students, teachers, and designers. *Instructional Science*, 42.

Lachapelle, C. P., Oh, Y., Shams, M. F., Hertel, J. D., and Cunningham, C. M. (2015). Hlm modeling of pre/post-assessment results from a large-scale efficacy study of elementary engineering. In *2015 ASEE Annual Conference & Exposition*, Seattle, Washington. ASEE Conferences.

Lampert, M., Franke, M. L., Kazemi, E., Ghousseini, H., Turrou, A. C., Beasley, H., Cunard, A., and Crowe, K. (2013). Keeping it complex: Using rehearsals to support novice teacher learning of ambitious teaching. *Journal of Teacher Education*, 64(3):226–243.

Land, S. M. (2000a). Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research and Development*, 48(3):61–78.

Land, S. M. (2000b). Cognitive requirements for learning with open-ended learning environments. *The Interdisciplinary Journal of Problem-based Learning*, 48(3):61–78.

Lawrence, L., Guo, B., Yang, K., Echeverria, V., Kang, Z., Bathala, V., Li, C., Huang, W., Aleven, V., and Rummel, N. (2022). Co-designing ai-based orchestration tools to support dynamic transitions: Design narratives through conjecture mapping. In *The Interdisciplinarity of the Learning Sciences, 16th International Conference of the Learning Sciences (ICLS) 2022*, pages 139–146.

Lee, A. Y., Mieczkowski, H. E., Ellison, N., and Hancock, J. (2022). The algorithmic crystal: Conceptualizing the self through algorithmic personalization on tiktok. In *The 25th ACM Conference On Computer-Supported Cooperative Work And Social Computing*. CSCW.

Leelawong, K. and Biswas, G. (2008). Designing learning by teaching agents: The betty's brain system. *International Journal of Artificial Intelligence in Education*, 18(3):181–208.

Lehrer, R. and Schauble, L. (1998). Reasoning about structure and function: Children's conceptions of gears. *Journal of Research in Science Teaching*, 35(1):3–25.

Levin, D. M., Hammer, D., and Coffey, J. E. (2009). Novice teachers' attention to student thinking. *Journal of Teacher Education*, 60(2):142–154.

Little, J. W., Gearhart, M., Curry, M., and Kafka, J. (2003). Looking at student work for teacher learning, teacher community, and school reform. *Phi Delta Kappan*, 85(3):184–192.

Liu, Z. and Stasko, J. (2010). Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):999–1008.

Lucas, B. and Hanson, J. (2016). Thinking like an engineer: Using engineering habits of mind and signature pedagogies to redesign engineering education. *International Journal of Engineering Pedagogy*, 6(2):4–13.

Luckin, R. and du Boulay, B. (2016). Reflections on the ecolab and the zone of proximal development. *International Journal of Artificial Intelligence in Education*, 26(1):416–430.

Mangaroska, K. and Giannakos, M. (2018). Learning analytics for learning design: A systematic literature review of analytics-driven design to enhance learning. *IEEE Transactions on Learning Technologies*, 12(4):516–534.

Mangaroska, K., Sharma, K., Gasevic, D., and Giannakos, M. (2020). Multimodal learning analytics to inform learning design: Lessons learned from computing education. *Journal of Learning Analytics*, 7(3):79–97.

Mangiante, E. S. and Gabriele-Black, K. A. (2020). Supporting elementary teachers' collective inquiry into the "e" in stem. *Science Education*, 29(4):1007–1034.

Martin, B. and Hanington, B. (2012). *Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport Publishers.

Martin, K., Bain, C., Swanson, H., Horn, M., and Wilensky, U. (2020). Building blocks: Kids designing scientific, domain-specific, block-based, agent-based microworlds. International Society of the Learning Sciences (ISLS).

Martinez-Maldonado, R., Pardo, A., Mirriahi, N., Yacef, K., Kay, J., and Clayphan, A. (2016). Latux: an iterative workflow for designing, validating and deploying learning analytics visualisations. *Journal of Learning Analytics*, 2(3):9–39.

Matuk, C., Gerard, L. F., Lim-Breitbart, J., and Linn, M. C. (2016). Gathering requirements for teacher tools: Strategies for empowering teachers through co-design. *Journal of Science Teacher Education*, 27:79–110.

Matuk, C. F. and Linn, Marcia C.and Eylon, B.-S. (2015). Technology to support teachers using evidence from student work to customize technology-enhanced inquiry units. *Instructional Science*, 43(2):229–257.

McElhaney, K., Basu, S., Wetzel, T., and Boyce, J. (2019). Three-dimensional assessment of ngss upper elementary engineering design performance expectations. In *NARST Annual International Conference*. NARST.

McElhaney, K. W., Zhang, N., Basu, S., McBride, E., Biswas, G., and Chiu, J. L. (2020). Using computational modeling to integrate science and engineering curricular activities. In *The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020*, pages 1357–1364. International Society of the Learning Sciences (ISLS).

Means, B., Chen, E., DeBarger, A., and Padilla, C. (2011). Teachers' ability to use data to inform instruction: Challenges and supports. Technical report.

Mehalik, M. M., Doppelt, Y., and Schuun, C. D. (2008). Middle-school science through design-based learning versus scripted inquiry: Better overall science concept learning and equity gap reduction. *Journal of Engineering Education*, 97(1):71–85.

Mishra, P. and Koehler, M. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6):1017–1054. Copyright: Copyright 2019 Elsevier B.V., All rights reserved.

Mislevy, R. J. and Haertel, G. D. (2006). Implications of evidence-centered design for educational testing. *Educational Measurement: Issues and Practice*, 25(4):6–20.

Molenaar, I. and Knoop-van Campen, C. A. N. (2019). How teachers make dashboard information actionable. *IEEE Transactions on Learning Technologies*, 12(3):347–355.

Muller, M. J., Wildman, D. M., and White, E. A. (1992). Taxonomy of participatory design practices: A participatory poster. In *Posters and Short Talks of the 1992 SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, page 34, New York, NY, USA. Association for Computing Machinery.

NAE and NASEM (2019). *Science and Engineering for Grades 6-12: Investigation and Design at the Center*. National Academies Press.

Neamtiu, I., Foster, J. S., and Hicks, M. (2005). Understanding source code evolution using abstract syntax tree matching. In *Proceedings of the 2005 international workshop on Mining software repositories*, pages 1–5.

NGSS (2013). *Next Generation Science Standards: For States, By States*. The National Academies Press.

Noss, R. and Hoyles, C. (1996). *Windows on Mathematical Meanings: Learning Cultures and Computers*. Springer Netherlands, 1 edition.

NRC (2000). *How people learn: Brain, mind, experience, and school: Expanded edition*. National Academies Press.

NRC (2007). *Taking Science to School: Learning and Teaching Science in Grades K-8*. National Academies Press.

NRC (2010). *Exploring the Intersection of Science Education and 21st Century Skills: A Workshop Summary*. National Academies Press.

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

NRC (2014). *STEM integration in K-12 education: Status, prospects, and an agenda for research*. National Academies Press.

Oxman, R. E. (1994). Precedents in design: a computational model for the organization of precedent knowledge. *Design Studies*, 15(2):141–157.

Papert, S. (1991). Situating constructionism. In Papert, I. H. . S., editor, *Constructionism*, pages 193—-206. Ablex Publishing, Westport, CT, USA.

Parker, M. C. and DeLyser, L. A. (2017). Concepts and practices: Designing and developing a modern k-12 cs framework. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '17, page 453–458, New York, NY, USA. Association for Computing Machinery.

Peel, A., Dabholkar, S., Anton, G., Wu, S., Wilensky, U., , and Horn, M. (2020). A case study of teacher professional growth through co-design and implementation of computationally enriched biology units. In *The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020*, pages 1950–1957. International Society of the Learning Sciences (ISLS).

Penner, D. E., Giles, N. D., Lehrer, R., and Schauble, L. (1997). Building functional models: Designing an elbow. *Journal of Research in Science Teaching*, 34(2):125–143.

Penuel, W., Roschelle, J., and Shechtman, N. (2007). Designing formative assessment software with teachers: an analysis of the co-design process. *Research and Practice in Technology Enhanced Learning*, 2:51–74.

Piech, C., Huang, J., Nguyen, A., Phulsuksombati, M., Sahami, M., and Guibas, L. (2015). Learning program embeddings to propagate feedback on student code. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1093–1102. JMLR.org.

Pierson, J. (2008). *The Relationship Between Patterns of Classroom Discourse and Mathematics Learning*. PhD thesis, University of Texas at Austin.

Prieto, L. P., Rodríguez-Triana, M. J., Martínez-Maldonado, R., Dimitriadis, Y., and Gašević, D. (2019). Orchestrating learning analytics (orla): Supporting inter-stakeholder communication about adoption of learning analytics at the classroom level. *Australasian Journal of Educational Technology*, 35(4).

Purzer, S., Hilpert, J. C., and Wertz, R. E. H. (2011). Cognitive dissonance during engineering design. In *2011 Frontiers in Education Conference (FIE)*.

Rabinovich, M., Stern, M., and Klein, D. (2017). Abstract syntax networks for code generation and semantic parsing. *arXiv preprint arXiv:1704.07535*.

Redish, E. F. and Wilson, J. M. (1993). Student programming in the introductory physics course: M.u.p.p.e.t. *American Journal of Physics*, 61(3):222–232.

Redish, E. F. and Wilson, J. M. (2011). Research notebook: Computational thinking—what and why. *The Link Magazine*, pages 20–23.

Reimann, P. (2016). Connecting learning analytics with learning research: the role of design-based research. *Learning: Research and Practice*, 2(2):130–142.

Reiser, B. J., Spillane, J. P., Steinmuller, F., Sorsa, D., Carney, K., and Kyza, E. A. (2000). Investigating the mutual adaptation process in teachers' design of technology-infused curricula. In *Fourth International Conference of the Learning Sciences*, pages 342–349.

Repenning, A., Webb, D., and Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, SIGCSE '10, page 265–269, New York, NY, USA. Association for Computing Machinery.

Robertson, A. D., Scherr, R., and Hammer, D. (2016). *Responsive Teaching in Science and Mathematics*. Routledge, Taylor Francis Group, New York, NY, USA.

Rodríguez-Triana, M. J., Prieto, L. P., Martínez-Monés, A., Asensio-Pérez, J. I., and Dimitriadis, Y. (2018). The teacher in the loop: Customizing multimodal learning analytics for blended learning. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, LAK '18, page 417–426, New York, NY, USA. Association for Computing Machinery.

Rokach, L. and Maimon, O. (2005). *Decision Trees*, pages 165–192. Springer US, Boston, MA.

Roschelle, J. and Penuel, W. (2006). Co-design of innovations with teachers: Definition and dynamics. 2:606–612.

Roth, W.-M. (1997). Interactional structures during a grade 4–5 open-design engineering unit. *Journal of Research in Science Teaching*, 34(3):273–302.

Sandoval, W. (2014). Conjecture mapping: An approach to systematic educational design research. *Journal of the Learning Sciences*, 23(1):18–36.

Sands, P., Yadav, A., and Good, J. (2018). *Computational Thinking in K-12: In-service Teacher Perceptions of Computational Thinking*, pages 151–164. Springer International Publishing, Cham.

Schauble, L., Klopfer, L. E., and Raghavan, K. (1991). Students' transition from an engineering model to a science model of experimentation. *Journal of Research in Science Teaching*, 28(9):859–882.

Schreiber, J. B., Nora, A., Stage, F. K., Barlow, E. A., and King, J. (2006). Reporting structural equation modeling and confirmatory factor analysis results: A review. *The Journal of Educational Research*, 99(6):323–338.

Schwartz, D. L., Bransford, J. D., and Sears, D. (2005). Efficiency and innovation in transfer. In Mestre, J., editor, *Transfer of learning: Research and perspectives*, page 1–52. Information Age Publishing, Greenwich, CT.

Schwarz, C., Passmore, C., and Reiser, B. (2017). *Helping Students make Sense of the World through Next Generation Science and Engineering Practices*.

Schwendimann, B. A., Rodríguez-Triana, M. J., Vozniuk, A., Prieto, L. P., Boroujeni, M. S., Holzer, A., Gillet, D., and Dillenbourg, P. (2017). Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Transactions on Learning Technologies*, 10(1):30–41.

Segedy, J. R., Kinnebrew, J. S., and Biswas, G. (2015a). Coherence over time: understanding day-to-day changes in students' open-ended problem solving behaviors. In *International Conference on Artificial Intelligence in Education*, pages 449–458. Springer.

Segedy, J. R., Kinnebrew, J. S., and Biswas, G. (2015b). Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *Journal of Learning Analytics*, 2(1):13–48.

Segedy, J. R., Kinnebrew, J. S., and Biswas, G. (2015c). Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *Journal of Learning Analytics*, 2(1):13–48.

Sengupta, P., Dickes, A., Farris, A. V., Karan, A., Martin, D., and Wright, M. (2015). Programming in k-12 science classrooms. *Commun. ACM*, 58(11):33–35.

Sengupta, P. and Farris, A. V. (2012). Learning kinematics in elementary grades using agent-based computational modeling: A visual programming-based approach. In *Proceedings of the 11th International Conference on Interaction Design and Children*, IDC '12, page 78–87, New York, NY, USA. Association for Computing Machinery.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., and 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.

Sherin, B., diSessa, A. A., and Hammer, D. (1993). Dynaturtle revisited: Learning physics through collaborative design of a computer model. *Interactive Learning Environments*, 3(2):91–118.

Sherin, B. L. (2001). How students understand physics equations. *Cognition and Instruction*, 19(4):479–541.

Sherin, M. and Russ, R. (2014). *Teacher Noticing via Video: The Role of Interpretive Frames*, pages 11–28. Routledge.

Sherin, M. G. (2002). A balancing act: Developing a discourse community in a mathematics classroom. *Journal of Mathematics Teacher Education*, 5(3):205–233.

Sherin, M. G. and Han, S. Y. (2004). Teacher learning in the context of a video club. *Teaching and Teacher Education*, 20(2):163–183.

Sherin, M. G., Philipp, R. A., and Jacobs, V. (2011). *Mathematics teacher noticing: Seeing through teachers' eyes*. Routledge, New York NY USA.

Shi, Y., Shah, K., Wang, W., Marwan, S., Penmetsa, P., and Price, T. (2021). *Toward Semi-Automatic Misconception Discovery Using Code Embeddings*, page 606–612. Association for Computing Machinery, New York, NY, USA.

Shibani, A., Knight, S., and Shum, S. B. (2019). Contextualizable learning analytics design: A generic model and writing analytics evaluations. In *Proceedings of the 9th International Conference on Learning Analytics Knowledge*, LAK19, page 210–219, New York, NY, USA. Association for Computing Machinery.

Shrader, G., Gomez, K., Gomez, L., Lachance-Whitcomb, J., and Finn, L.-E. (2001). Participatory design of science curricula: The case for research for practice. In *Paper presented at the Annual Meeting of the American Educational Research Association*.

Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1):153–189.

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

Siemens, G. and Baker, R. S. J. d. (2012). Learning analytics and educational data mining: Towards communication and collaboration. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, LAK '12, page 252–254, New York, NY, USA. Association for Computing Machinery.

Sims-Knight, J., Upchurch, R., Pendergrass, N., Meressi, T., Fortier, P., Tchimev, P., VonderHeide, R., and Page, M. (2004). Using concept maps to assess design process knowledge. In *34th Annual Frontiers in Education, 2004 (FIE 2004)*.

Slotta, J. D. and Linn, M. C. (2009). *WISE Science: Web-Based Inquiry in the Classroom*. Teachers College Press, New York.

Slotta, J. D., Tissenbaum, M., and Lui, M. (2013). Orchestrating of complex inquiry: Three roles for learning analytics in a smart classroom infrastructure. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, LAK '13, page 270–274, New York, NY, USA. Association for Computing Machinery.

Snyder, C., Hutchins, N., Biswas, G., Emara, M., Grover, S., and Conlin, L. (2019a). Analyzing students' synergistic learning processes in physics and ct by collaborative discourse analysis. In *Proceedings of the International Conference on Computer Supported Collaborative Learning*, pages 360–367. International Society of the Learning Sciences (ISLS).

Snyder, C., Hutchins, N., Biswas, G., Emara, M., Grover, S., and Conlin, L. (2019b). Analyzing students' synergistic learning processes in physics and ct by collaborative discourse analysis. In *Computer-supported collaborative learning*.

Snyder, C., Narasimham, G., Hutchins, N., Biswas, G., and Yett, B. (2022). Examining how prior knowledge impacts students' discussions and knowledge construction during computational model building. In *American Educational Research Association Annual Meeting*.

Soloway, E. (1993). Should we teach students to program? *Commun. ACM*, 36(10):21–24.

Streveler, R. A., Litzinger, T. A., Miller, R. L., and Steif, P. S. (2008). Learning conceptual knowledge in the engineering sciences: Overview and future research directions. *Journal of Engineering Education*, 97(3):279–294.

Swanson, H., Anton, G., Bain, C., Horn, M., and Wilensky, U. (2019). *Introducing and Assessing Computational Thinking in the Secondary Science Classroom*, pages 99–117. Springer Singapore, Singapore.

Taub, M., Azevedo, R., Rajendran, R., Cloude, E. B., Biswas, G., and Price, M. J. (2019). How are students' emotions related to the accuracy of cognitive and metacognitive processes during learning with an intelligent tutoring system? *Learning and Instruction*.

Tissenbaum, M., Lui, M., and Slotta, J. D. (2012). Co-designing collaborative smart classroom curriculum for secondary school science. *j-jucs*, 18(3):327–352.

Tissenbaum, M. and Slotta, J. D. (2012). Scaffolding a knowledge community for high school physics. In *Proceedings of the 12th International Conference of the Learning Sciences (ICLS) 2012*. International Society of the Learning Sciences.

van Es, E. A. and Sherin, M. G. (2002). Learning to notice: Scaffolding new teachers' interpretations of classroom interactions. *Journal of Technology and Teacher Education*, 10(4):571–596.

van Leeuwen, A. (2015). Learning analytics to support teachers during synchronous cscl: balancing between overview and overload. *Journal of Learning Analytics*, 2(2):138–162.

van Leeuwen, A., Rummel, N., and van Gog, T. (2019). What information should cscl teacher dashboards provide to help teachers interpret cscl situations? *International Journal of Computer-Supported Collaborative Learning*, pages 1–29.

VanLehn, K. (2013). Model construction as a learning activity: a design space and review. *Interactive Learning Environments*, 21(4):371–413.

VanLehn, K., Wetzel, J., Grover, S., and v. d. Sande, B. (2017). Learning how to construct models of dynamic systems: An initial evaluation of the dragoon intelligent tutoring system. *IEEE Transactions on Learning Technologies*, 10(2):154–167.

Vatrapu, R., Teplovs, C., Fujita, N., and Bull, S. (2011). Towards visual analytics for teachers' dynamic diagnostic pedagogical decision-making. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, LAK '11, page 93–98, New York, NY, USA. Association for Computing Machinery.

Verbert, K., Duval, E., Klerkx, J., Govaerts, S., and Santos, J. L. (2013). Learning analytics dashboard applications. *American Behavioral Scientist*, 57(10):1500–1509.

Vieira, C., Hathaway Goldstein, M., Purzer, , and Magana, A. J. (2016). Using learning analytics to characterize student experimentation strategies in the context of engineering design. *Journal of Learning Analytics*, 3:291–317.

Voyiatzaki, E. and Avouris, N. (2014). Support for the teacher in technology-enhanced collaborative classroom. *Education and Information Technologies*, 19(1):129–154.

Walkoe, J., Wilkerson, M., and Elby, A. (2017). Technology-mediated teacher noticing: A goal for classroom practice, tool design, and professional development. In *Proceedings of the 12th International Conference on Computer Supported Collaborative Learning (CSCL) 2017*. International Society of the Learning Sciences.

Wang, H.-H., Moore, T. J., Roehrig, G. H., and Park, M. S. (2011). Stem integration: Teacher perceptions and practice. *Journal of Pre-College Engineering Education Research (J-PEER)*, 1(2).

Wang, W., Zhang, C., Stahlbauer, A., Fraser, G., and Price, T. (2021). Snapcheck: Automated testing for snap programs. In *Proceedings of the 2021 Conference on Innovation Technology in Computer Science Education*, ITiCSE 2021. Association for Computing Machinery.

Warren, B. and Rosebery, A. S. (1995). "this question is just too, too easy!" perspectives from the classroom on accountability in science. Technical report, Washington, DC.

Watkins, J., McCormick, M., Wendell, K. B., Spencer, K., Milto, E., Portsmore, M., and Hammer, D. (2018). Data-based conjectures for supporting responsive teaching in engineering design with elementary teachers. *Science Education*, 102(3):548–570.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., and Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1):127–147.

Weller, D., Bott, T., Caballero, M., and Irving, P. (2021). Developing a learning goal framework for computational thinking in computationally integrated physics classrooms.

Wen, C.-T., Liu, C.-C., Chang, H.-Y., Chang, C.-J., Chang, M.-H., Fan Chiang, S.-H., Yang, C.-W., and Hwang, F.-K. (2020). Students' guided inquiry with simulation and its relation to school science achievement and scientific literacy. *Computers Education*, 149:103830.

Wendell, K. B. and Rogers, C. (2013). Engineering design-based science, science content performance, and science attitudes in elementary school. *Journal of Engineering Education*, 102(4):513–540.

Wendell, Kristen Bethke, W. J. . J. A. W. (2016). Noticing, assessing, and responding to students' engineering: Exploring a responsive teaching approach to engineering design. In *Proceedings of the 123rd American Society for Engineering Education Annual Conference*, pages 26–29. Association Society for Engineering Education.

Werner, L., Denner, J., Campe, S., and Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, SIGCSE '12, page 215–220, New York, NY, USA. Association for Computing Machinery.

Werner, L., McDowell, C., and Denner, J. (2013). A first step in learning analytics: Pre-processing low-level alice logging data of middle school students. *Journal of Educational Data Mining*, 5(2):11–37.

Wilensky, U. and Resnick, M. (1999). Thinking in levels: A dynamic systems approach to making sense of the world. *Journal of Science Education and technology*, 8(1):3–19.

Wiley, K. J., Dimitriadis, Y., Bradford, A., and Linn, M. C. (2020). From theory to action: Developing and evaluating learning analytics for learning design. In *Proceedings of the Tenth International Conference on Learning Analytics Knowledge*, LAK '20, page 569–578, New York, NY, USA. Association for Computing Machinery.

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

Windschitl, M., Thompson, J., Braaten, M., and Stroupe, D. (2012). Proposing a core set of instructional practices and tools for teachers of science. *Science Education*, 96(5):878–903.

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

Winne, P. and Baker, R. (2013). The potentials of educational data mining for researching metacognition, motivation and self-regulated learning. *Journal of Educational Data Mining*, 5(1):1–8.

Winne, P. H. and Hadwin, A. F. (2013). nstudy: Tracing and supporting self-regulated learning in the internet. In *International handbook of metacognition and learning technologies*, pages 293–308. Springer.

Wise, A. and Vytasek, J. (2017). Learning Analytics Implementation Design. In Lang, C., Siemens, G., Wise, A. F., and Gaševic, D., editors, *The Handbook of Learning Analytics*, pages 151–160. Society for Learning Analytics Research (SoLAR), Alberta, Canada, 1 edition.

Witten, D. M. and Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726. PMID: 20811510.

Wu, L., Looi, C.-K., Liu, L., and How, M.-L. (2018). Understanding and developing in-service teachers' perceptions towards teaching in computational thinking: Two studies. In *Proceedings of the 26th International Conference on Computers in Education*, pages 735–742. Asia-Pacific Society for Computers in Education.

Wu, S., Peel, A., Bain, C., Anton, G., Horn, M., and Wilensky, U. (2020). Workshops and co-design can help teachers integrate computational thinking into their k-12 stem classes. *Proceedings of International Conference on Computational Thinking Education 2020*.

Xhakaj, F., Aleven, V., and McLaren, B. M. (2017). Effects of a teacher dashboard for an intelligent tutoring system on teacher knowledge, lesson planning, lessons and student learning. In Lavoué, É., Drachsler, H., Verbert, K., Broisin, J., and Pérez-Sanagustín, M., editors, *Data Driven Approaches in Digital Education*, pages 315–329, Cham. Springer International Publishing.

Xie, C., Zhang, Z., Nourian, S., Pallant, A., and Hazzard, E. (2014). Time series analysis method for assessing engineering design processes using a cad tool. *International Journal of Engineering Education*, 30:218–230.

Xing, W., Li, C., Chen, G., Huang, X., Chao, J., Massicotte, J., and Xie, C. (2021). Automatic assessment of students' engineering design performance using a bayesian network model. *Journal of Educational Computing Research*, 59(2):230–256.

Xu, S. and Chee, Y. S. (2003). Transformation-based diagnosis of student programs for programming tutoring systems. *IEEE Transactions on Software Engineering*, 29(4):360–384.

Yadav, A., Gretter, S., Hambrusch, S., and Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education*, 26(4):235–254.

Yaşar, , Baker, D., Robinson-Kurpius, S., Krause, S., and Roberts, C. (2006). Development of a survey to assess k-12 teachers' perceptions of engineers and familiarity with teaching design, engineering, and technology. *Journal of Engineering Education*, 95(3):205–216.

Yin, Y., Hadad, R., Tang, X., and Lin, Q. (2020). Improving and assessing computational thinking in maker activities: the integration with physics and engineering learning. *Journal of Science Education and Technology*, 29(2):189–214.

Zhang, N. (2020). *Supporting the Integrated Learning of Science, Engineering, and Computational Thinking in an Open-ended Learning Environment*. PhD thesis, Vanderbilt University.

Zhang, N., Biswas, G., Chiu, J. L., and McElhaney, K. W. (2019). Analyzing students' design solutions in an ngss-aligned earth sciences curriculum. In Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., and Luckin, R., editors, *Artificial Intelligence in Education*, pages 532–543, Cham. Springer International Publishing.

Zhang, N., Biswas, G., and Dong, Y. (2017). Characterizing students' learning behaviors using unsupervised learning methods. In *International Conference on Artificial Intelligence in Education*, pages 430–441. Springer.

Zhang, N., Biswas, G., and Hutchins, N. (2021). Measuring and analyzing students' strategic learning behaviors in open-ended learning environments. *International Journal of Artificial Intelligence in Education*.

Zhang, N., Biswas, G., McElhaney, K. W., Basu, S., McBride, E., and Chiu, J. L. (2020). Studying the interactions between science, engineering, and computational thinking in a learning-by-modeling environment. In *International Conference on Artificial Intelligence in Education*, pages 598—-609. Springer.

Zhang, N., Hutchins, N., and Biswas, G. (2022). *Towards a Deeper Understanding of K-12 Students' CT and Engineering Design Processes*.

Appendix A

List of Publications

Journals (Peer-Reviewed)

1. Zhang, N., Biswas, G., & Hutchins, N.M. (2021). The Role of Strategies in Student Learning: Measuring and Analyzing Strategic Learning Behaviors. International Journal of Artificial Intelligence in Education

2. Emara, M., Hutchins, N.M., Grover, S., Snyder, C., & Biswas, G. (2021). Examining Students' Regulation of Collaborative, Computational, Problem-Solving Processes in Open-Ended Learning Environments. Journal of Learning Analytics.

3. Hutchins, N.M., Biswas, G., Zhang, N., Snyder, C., Ledeczi, A., & Maroti, M. (2020). Domain-Specific Modeling Languages in Computer-Based Learning Environments: A Systematic Approach to Support Science Learning through Computational Modeling. International Journal of Artificial Intelligence in Education, 30(1), 537–580.

4. Hutchins, N.M., Biswas, G., Maroti, M., Ledeczi, A., Grover, S., Wolf, R., Blair, K.P., Chin, D., Conlin, L., Basu, S., & McElhaney, K. (2019). C2STEM: A System for Synergistic Learning of Physics and Computational Thinking. Journal of Science Education and Technology.

5. Vadaparampil S.T., Hutchins N.M., & Quinn G.P. (2013). Reproductive health in the adolescent and young adult cancer patient: an innovative training program for oncology nurses. J Cancer Ed, 28(1), 197–208

Conference Publications (Peer-Reviewed)

1. Cochran, K., Cohn, C., Hutchins, N.M., Biswas, G., Hastings, P. (in press). Improving Automated Evaluation of Formative Assessments with Text Data Augmentation. To appear in Proceedings of 23rd International Conference on Artificial Intelligence in Education.

2. Snyder, C., Hutchins, N.M., Biswas, G., Narasimham, G., Emara, M., Yett, B. (2022). Instructor facilitation of STEM+CT discourse: engaging, prompting, and guiding students' computational modeling in physics. In Chinn, C., Tan, E., Chan, C., Kali, Y. (Eds.). Proceedings of the 16th International Conference of the Learning Sciences - ICLS 2022. Hiroshima, Japan: International Society of the Learning Sciences.

3. Basu, S., McElhaney, K., Rachmatullah, A., Hutchins, N.M., Biswas, G., Chiu, J. (2022). Promoting Computational Thinking Through Science-Engineering Integration Using Computational Modeling. In Chinn, C., Tan, E., Chan, C., Kali, Y. (Eds.). Proceedings of the 16th International Conference of the Learning Sciences - ICLS 2022. Hiroshima, Japan: International Society of the Learning Sciences.

4. Hutchins, N.M. Biswas, G. (2022). Teacher Noticing and Response to Students' Computational and Engineering Design Strategies. Presented at the American Educational Research Association 2022 Symposium on AI and the Future of STEM Instruction: Designing New Models to Automate Feedback to Teachers.

5. Snyder, C., Narasimham, G., Hutchins, N.M., Biswas, G., Yett, B. (2022). Examining how prior knowledge impacts students' discussions and knowledge construction during computational model building. In Proceedings of the American Educational Research Association Annual Meeting.

6. Hutchins, N.M., Basu, S., McElhaney, K., Chiu, J., Fick, S., Zhang, N., Biswas, G. (2021). Coherence across conceptual and computational representations of students' scientific models. In E. de Vries, J. Ahn, Y. Hod (Eds.), 15th International Conference of the Learning Sciences – ICLS 2021 (pp. 330-337). International Society of the Learning Sciences.

7. Hutchins, N.M., Snyder, C., Emara, M., Grover, S., Biswas, G. (2021). Analyzing debugging processes during collaborative, computational modeling in science. In C. Hmelo-Silver, B. de Wever, J. Oshima (Eds.), 14th International Conference on Computer-Supported Collaborative Learning – CSCL 2021 (pp. 221–224). International Society of the Learning Sciences.

8. Hutchins, N., Biswas, G., Wolf, R., Chin, D., Grover, S., & Blair, K. (2020). Computational thinking in support of learning and transfer. In Proceedings of the International Conference of the Learning Sciences (ICLS), Nashville, TN, USA.

9. Conlin, L., Hutchins, N., Grover, S., & Biswas, G. (2020). "Doing Physics" And "Doing Code": Students' Framing During Computational Modeling in Physics. In Proceedings of the International Conference of the Learning Sciences (ICLS), Nashville, TN, USA.

10. Emara, M., Grover, S., Hutchins, N., Biswas, G., & Snyder, C. (2020). Examining Students' Debugging and Regulation Processes During Collaborative Computational Modeling in Science. In Proceedings of the International Conference of the Learning Sciences (ICLS), Nashville, TN, USA.

11. Snyder, C., Hutchins, N., Biswas, G., Emara, M., Yett, B., & Mishra, S. (2020). Understanding Collaborative Question Posing During Computational Modeling in Science. In Proceedings of the International Conference on Artificial Intelligence in Education, Ifrane, Morocco.

12. Snyder, C., Hutchins, N., Biswas, G., Mishra, S., & Emara, M. (2020). Exploring Synergistic Learning Processes through Collaborative Learner-to-Learner Questioning. In Proceedings of the International Conference of the Learning Sciences (ICLS), Nashville, TN, USA.

13. Yett, B., Hutchins, N., Snyder, C., Zhang, N., Mishra, S., & Biswas, G. (2020). Evaluating Student Learning in a Synchronous, Collaborative Programming Environment through Log-Based Analysis of Projects. In Proceedings of the International Conference on Artificial Intelligence in Education, Ifrane, Morocco.

14. Yett, B., Hutchins, N., Stein, G., Zare, H., Snyder, C., Biswas, G., Metelko, M., & Ledeczi, A. (2020). A Hands-On Cybersecurity Curriculum Using a Robotics Platform. In Proceedings of the Special Interest Group on Computer Science Education (SIGCSE) Annual Meeting, Portland, USA.

15. Yett, B., Snyder, C., Hutchins, N.M., & Biswas, G. (2020). Exploring the Relationship Between Collaborative Discourse, Programming Actions, and Cybersecurity and Computational Thinking Knowledge. In Proceedings of IEEE TALE.

16. Yett, B., Snyder, C., Zhang, N., Hutchins, N.M., Mishra, S., & Biswas, G. (2020). Using Log and Discourse Analysis to Improve Understanding of Collaborative Programming. In Proceedings of the International Conference on Computers in Education (ICCE). **Best Student Paper Award**

17. Hutchins, N., Shi, C., & Biswas, G. (2019) A High School Computational Modeling Approach to Studying the Effects of Climate Change on Coral Reefs. In Proceedings of the American Educational Research Association Annual Meeting, Toronto, Canada.

18. Hutchins, N., Biswas, G., Grover, S., Basu, S., & Snyder, C. (2019). A systematic approach for analyzing students' computational modeling processes in C2STEM. In Proceedings of the 20th International Conference on Artificial Intelligence in Education (AIED 2019), pp. 116-121, Chicago, USA.

19. Grover, S., Hutchins, N., Biswas, G., Snyder, C., & Emara, M. (2019) Examining Synergistic Learning of Physics and Computational Thinking through Collaborative Problem Solving in Computational Modeling. In Proceedings of the American Educational Research Association Annual Meeting, Toronto, Canada.

20. Lédeczi, Á, Maroti, M., Zare, H., Yett, B., Hutchins, N, Volgyesi, P., Broll, B, Darrah, T., Metelko, M., Smith, M., Biswas, G., & Koutsoukos, X. Teaching Cybersecurity with Networked Robots. (2019). In Proceedings of the Special Interest Group on Computer Science Education (SIGCSE) Annual Meeting, Minneapolis, MN.

21. Snyder, C., Hutchins, N., Biswas, G., & Grover, S. (2019). Understanding Students' Model Building Strategies through Discourse Analysis. In Proceedings of the 20th International Conference on Artificial Intelligence in Education (AIED 2019), 263-268, Chicago, USA.

22. Snyder, C., Hutchins, N., Biswas, G., Emara, M., Grover, S., & Conlin, L. (2019). Analyzing Students' Synergistic Learning Processes in Physics and CT by Collaborative Discourse Analysis. In Proceedings of the International Conference on Computer Supported Collaborative Learning, Lyon, France.

23. Hutchins, N., Biswas, G., Conlin, L., Emara, M., Grover, S., Basu, S., & McElhaney, K. (2018). Studying Synergistic Learning of Physics and Computational Thinking in a Learning by Modeling Environment. In Yang, J. C. et al. (Eds.). In Proceedings of the 26th International Conference on Computers in Education (ICCE), Manila, Philippines, pp. 153-162. **Best Student Paper Award, Nominated Best Paper Award**

24. Hutchins, N., Darrah, T., Zare, H., & Biswas, G. (2018). A DSML for a Robotics Environment to Support Synergistic Learning of CT and Geometry. Kong, S. C., Sheldon, J., & Li, K. Y.. (Eds.). In Proceedings of International Conference on Computational Thinking Education 2018. Hong Kong, 77-82.

25. Hutchins, N., Biswas, G., Maroti, M., Ledeczi, A., & Broll, B. (2018). A design-based approach to a classroom-centered OELE. In Proceedings of the 19th International Conference on Artificial Intelligence in Education (AIED), London, 155-159

26. Darrah, T., Hutchins, N., & Biswas, G. (2018). Design and Development of a Low-Cost Open-Source Robotics Education Platform. In Proceedings of the 50th International Symposium on Robotics, Munich, Germany.

27. Hutchins, N., Zhang, N., & Biswas, G. (2017). The Role Gender Differences in Computational Thinking Confidence Levels Plays in STEM Applications. Kong, S. C., Sheldon, J., & Li, K. Y.. (Eds.). In Proceedings of International Conference on Computational Thinking Education 2017. Hong Kong, 33-38.

28. Quinn GP, Hutchins N, Nelson A, & Vadaparampil ST. (2012). Working with expert panels to develop a training program for oncology nursing. In Proceedings of the International Cancer Education Conference, Ann Arbor, MI.

Book Chapters (Peer-Reviewed)

1. Biswas, G., Zhang, N., & Hutchins, N. (2022). Leveraging Learning Analytics for a Deeper Understanding of Students' Engineering Design Processes in K-12 Science Classrooms. In A. Amirhossein and colleagues(Ed.). *Artificial Intelligence in STEM Education: The Paradigmatic Shifts in Research, Education, and Technology*. Taylor & Francis.

Conference Presentations and Symposia (Peer-Reviewed)

1. Kafai, Y., Biswas, G., Hutchins, N., Snyder, C., Brennan, K., Haduong, P., et al. (2020). Turning Bugs into Learning Opportunities: Understanding Debugging Processes, Perspectives, and Pedagogies. In Proceedings of the International Conference of the Learning Sciences (ICLS), Nashville, TN, USA.

2. Dorsey, C., Haavind, S., Hutchins, N., & Levin, M. (2019). Computational Thinking in STEM from Preschool to High School: Research and Practice. In Proceedings of International Society for Technology in Education Annual Meeting, Philadelphia, PA, USA.

3. Biswas, G., Hutchins, N., Lédeczi, Á., Grover, S., & Basu, S. (2019). Integrating Computational Modeling in K-12 STEM Classrooms. In Proceedings of the Special Interest Group on Computer Science Education (SIGCSE) Annual Meeting, Minneapolis, USA.

4. Hoppe, U., Looi, C., Biswas, G., & Hutchins, N. (2018). Introduction to Computational Thinking. In Proceedings of the International Conference on Computers in Education, Manila, Philippines.

Appendix B

Completed Studies

Location	Student Population	Domain	n
Tennessee	High performing high school student program at Vanderbilt, elective participation	PHY	13
Tennessee	Elective participation during computer science camp held at ISIS	PHY	10
Tennessee	Higher performing school; 30% minority students (higher than state average), honors physics class	PHY	90
Tennessee	High performing high school student program at Vanderbilt, collaboration study	PHY	26
California	75% students from low income families, 75percent underrepresented minorities	PHY	40
Tennessee	Minority population is 86%, lower proficiency in Math and Reading than state average	PHY	90
Illinois	High performing middle school	PHY & MB	40
California	70% students from low income families, 80percent underrepresented minorities	PHY	480
Massachusetts	Linguistically and economically diverse public charter high school	PHY	50
Illinois	High performing middle school	PHY & MB	40
Tennessee	60% minority population	MB	120
Tennessee	50% minority, performs at about state average	GEN	110
Tennessee	High performing middle school	ES	99
Tennessee	High performing high school student program at Vanderbilt, collaboration study	PHY	26

Appendix C

Example Teacher Simulation Discourse

Person	Quote
Teacher	Okay. So I'm gonna look at graphs because I like looking at graphs and I'm going to talk while I do this.
Teacher	Yes, please. Because then it will help me if I have questions.
Teacher	All right. Um, so correct. Incorrect. So they're not so great at remembering to initiate variables which is good because I they free I always forget that too.
Teacher	Equal to. Less Than. Interesting.
Teacher	Less than Oh, so that means like this. They didn't get to it like if there's only Yes, okay, great.
Teacher	So a lot of them just got stuck probably because they didn't initiate it.
Teacher	And then most of it got equal to so love that.
Teacher	That's interesting these concepts, success opportunity. pedological change curriculum edition.
Teacher	Um, so what's the difference between opportunity and pedological change and curriculum addition.
Researcher	So for curriculum addition those are new material to develop in the future. So hey, we need to change that next time we implement
Teacher	I understand, like a big change.
Researcher	Yeah. If you already know exactly what you'd want for a pedagogical change for next time, then I'd say pedagogical change.
Teacher	Right, I would say start class with review of initial variables. And I would pick students. I'll just do this. At the end, I'll just say pick students to lead slash drive.
Teacher	Okay, so I'll submit that.
Researcher	And then just close that open just in case you wanted another, like teachers want another answer.
Teacher	And I say opportunity. Don't review equal to yet. See how goes next day.
Teacher	Okay. And then behaviors. So who tried different values for the rainfall? No. total rainfall? Yes. So materials [needs work].
Teacher	So we need to say briefly tell them to test or briefly remind them to test different variables.
Teacher	And then students based on strategy.
Teacher	So diver, that's the Tinker, Trial by Fire strategists. Okay.
Teacher	So then I would say here I would just say have Blake and picking Blake's. I have a student named Blake who I love. He's like my favorite person this year. Have Blake show and talk through his code I don't know if it was his her code there. Because there are more kids who don't have genders now, which is great. Their code submit as an example, to help the divers.
Teacher	Now, these are notes. Oh.
Researcher	So this is all of the feedback that's automatically generated. So I have the success and opportunity from this group.
Teacher	Okay, great. Great.
Teacher	So I'm definitely not looking at that after after this first day.
Teacher	Okay. And then success is so great.
Teacher	And I've already like, called out Blake to present. And I already talked about that, great
Teacher	And [students that] contain errors in their models go to trial by fire.
Teacher	Okay, so I think I've addressed those two, by looking at the graphs.
Teacher	Okay. So then I'm going over to my strategies.
Teacher	Oh, my God, look at this. It's so wavy.

Continued on next page.

Person	Quote
Teacher	Yeah, there's a lot and then equal to I understand.
Researcher	Yeah. And I it's just they didn't really get to a yet.
Teacher	Okay, great. Okay. All right. Awesome. And then I'm just gonna click on this just for love. There's a lot here, sorry.
Researcher	I'm gonna try and dive deeper. And I tried color coding it to make it less, yeah.
Teacher	I'm not reading this yet. And I'm just nope, my brain says just great. Because I think this is great. You can put it as a side note, I am really tired. So you are getting me on like the peak of like, I have very little motivation to do anything. Kind of
Researcher	I love it. Yeah.
Teacher	I would be like this is this is I don't need all this information. This is information.
Teacher	But here, okay, so didn't equal to see how it goes next day, right?
Teacher	And then start class with the initial review of variables pick students to lead drive.
Teacher	So I would say then remind them to test different variables. So start with Blake.
Researcher	Oh, wait. Okay.
Teacher	Oh, that in respond. Like, should I like?
Researcher	Do you have any, like ordering here? What's most important of your?
Teacher	Yeah, I do like I would want so in my head, I was gonna, like, put a new reflection of like, make a plan Because I see that I would have like, Blake, kind of start by reviewing their code, right? And then I would say like, Hey, as Blake is reviewing their code, make sure you interject to, to review the initial variables, like make sure like you're enforcing that point in and out. And then as Blake is talking, like, I'm like, assuming Blake did test different variables to like, call that out as well. So I was gonna make like little interjection notes for myself.
Researcher	So what we'll do is at the bottom, for science, CT, engineering or strategy, right, would you say like, What is your response? So you can click on like, if it's, it's a strategy? Yeah.
Teacher	Yeah, strategy. So I want to Yeah, I definitely want to see that reflect. So I want to say, start with Blake, they're showing slash talking through their code [to the class] And then I want to say, make sure to highlight, initializing variables. And testing different materials. No, more than, like, seven minutes on this.
Researcher	Okay. Yep. So for the rest of that class, we do have that on paper, a debugging activity that they can do. Do you think want to spend that time on the debugging activity and then have them jump right back in (to the code)? Or no?
Teacher	I would want to start with this. Because I feel like we're just going to see the same mistakes, again, in the debugging activity, if they don't If they haven't, because obviously they don't. It's kind of one of the things they don't know what they don't know. So I think a lot probably at this point, and because I've done this in the past, a lot of them don't know that they're making mistakes