8-2021

Understanding Teacher Sense-Making Discourse During Collaborative Professional Development of an Expansively-Framed Computer Science Curriculum

Courtney Stephens
Utah State University

UNDERSTANDING TEACHER SENSE-MAKING DISCOURSE DURING COLLABORATIVE PROFESSIONAL DEVELOPMENT OF AN EXPANSIVELY-FRAMED COMPUTER SCIENCE CURRICULUM

by

Courtney Stephens

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Instructional Technology and Learning Sciences

Approved:

Mimi Recker, Ph.D. Major Professor	Jody Clarke-Midura, Ed.D. Committee Member
Hillary Swanson, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Interim Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2021

ABSTRACT

Understanding Teacher Sense-making Discourse During Collaborative Professional Development of an Expansively-framed Computer Science Curriculum

by

Courtney Stephens, Master of Science

Utah State University, 2021

Major professor: Dr Mimi Recker
Department: Instructional Technology & Learning Science

Elementary school teachers are increasingly asked to teach computer science (CS) curricula with which they have little familiarity. To help teachers become familiar with the CS content, professional development (PD) is needed—specifically, collaborative PD that encourages teacher reflection, modeling, and collective participation. This thesis study uses the findings from a study of elementary teachers participating in a collaborative PD developed to accompany a novel computer science instructional unit. This seven-lesson unit used an expansive framing model to introduce students to programming concepts by having students first play an "unplugged" tabletop board game and then create game levels in Scratch. The PD sessions were structured as a set of participatory routines where the previous week's lesson was reviewed, the upcoming lesson was modeled, and then adaptations to it were discussed. Analyses of teacher discourse during PD revealed three kinds of sense-making episodes (suggestions, reflections, and connections). Analyses of these episodes show that a majority of suggestion episodes, as well as many connection episodes, were used during subsequent classroom implementations of the curriculum, indicating teachers' reliance on each other in the collaborative PD to teach the CS content, despite their collective lack of experience. Finally, analyses also showed that connection and suggestion episodes were frequently grounded in the board game, matching the intent of the instructional approach for supporting teacher learning and showing how teachers learned from the expansive framing model as well.

(98 pages)

PUBLIC ABSTRACT

Understanding Teacher Sense-making Discourse During Collaborative Professional Development of an Expansively-framed Computer Science Curriculum

Courtney Stephens

Elementary school teachers are being increasingly asked to teach computer science—something that most teacher certification programs do not prepare them for. In an attempt to study how elementary teachers learn to teach computer science, I analyzed the ways that teachers behaved during a professional development accompanying the implementation of a fifth-grade computer science curriculum. My findings suggest that teachers benefit from professional development that encourages collaboration and active participation in teachers through discussion and modeling. Furthermore, my findings suggest that teachers benefit from using curriculum that deliberately connects new concepts to content that they are already familiar and comfortable with—a model known as expansive framing. By encouraging active teacher participation in professional development and by using curriculum that relates to teachers' existing content knowledge, we may be able to help elementary teachers prepare to teach computer science with more confidence and accuracy.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation grant DRL-1837224. The opinions expressed herein are those of the authors and do not necessarily reflect those of the funding organization. We thank our partner educators, librarian, and students.

I give special thanks to my family, friends, and colleagues for their encouragement, moral support, and patience as I worked my way from the initial proposal writing to this final document. I could not have done it without all of you.

Courtney Stephens

CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

Chapter I: Introduction

In the United States, there is an increasing demand, as shown through new state standards and emerging school district programs, to incorporate computer science (CS) curricula into elementary school settings. As such, elementary school teachers who have limited experience with computer science need support developing pedagogical techniques and content knowledge to teach the CS curriculum that is frequently now expected of them.

Professional development (PD) or in-service teacher training has typically been used as the means of providing that support, given the limitations in pre-service teacher education and the recent nature of these changes. However, evidence is mixed in terms of how effectively PD results in observable and sustainable change in classroom practice (e.g., Ball & Cohen, 1999; Borko, 2004). In response to this mixed evidence, new approaches to PD have been developed under the general term of "collaborative PD." This PD format centers around inviting teachers to be active participants in both their own learning and the curricular design process. Examples of such PD include engaging teachers in collaboratively designing curricula (co-design: Peel et al., 2020; Voogt et al., 2015) and modeling new curricula (Goode et al., 2014).

To engage teachers in gaining agency in the curricular process and to encourage active sense-making, what counts as productive talk in the context of collaborative PD must be identified (Lefstein et al., 2020). In this thesis, I identify several aspects of productive pedagogical talk that appear to support teacher agency and sense-making in ways that influence classroom practice (Walkoe & Luna, 2020).

A previous study developed a fifth-grade curriculum centered around using an expansive framing approach to teach CS concepts (Lee et al., 2020). As defined by Engle et al. (2012), expansive framing is a theoretical model that draws upon a situated learning account of transfer. Transfer occurs when two or more contexts are re-framed for learners so that they are seen as instances of the same concepts or ideas. The curriculum's expansive framing approach introduced programming concepts by first having students play a computing-rich tabletop board game and then create and program their own board game levels in the block-based programming language, Scratch. The seven-week curriculum was designed to frame a familiar unplugged context (board games) as a computing rich space to support students in learning computing concepts. The project worked with a team of fifth-grade teachers at one school who had varying but altogether limited experience in CS and who participated in the project by attending weekly PD meetings and then implementing the lessons in their classroom.

While the aim of the project was initially to study student CS learning using board games as a frame and to increase student intrinsic interest in CS, the data from the implementation of this CS curriculum can also be used to answer questions about productive PD in the context of teachers learning CS. The teachers in this context needed to develop familiarity with the necessary CS content and pedagogical practices and were given the opportunity to do so in a collaborative PD setting that built in time for reflections and modeling to help the teachers both learn the key computing concepts as well as to help them adapt the implementation of the curriculum to their classroom contexts.

The purpose of this thesis study is to examine how teachers leverage the collaborative PD approach and their peers' contributions in this setting to increase their CS pedagogical and content knowledge. I also examine the extent that they gain agency in the context of the curriculum, which is demonstrated through their implementations of the curriculum in their classrooms. Using the data from this CS curricular implementation, I focus on the pedagogical talk that occurs in the PD setting, as well as the ways that the teachers' CS learning benefits from the expansive framing in this context. Focusing on pedagogical talk allows me to identify how teachers engaged in collaborative sense-making and how contributions of that sense-making can be traced to their implementation in the classroom as evidence of teacher curricular agency.

Research Questions

The thesis study is guided by the following research questions:

1. How do teachers discuss new CS content and pedagogy in a collaborative professional development setting?
2. What types of teacher discourse about new curricula that occur during collaborative PD impact teachers' subsequent classroom implementations?
3. How does an expansively-framed curriculum support *teacher l*earning?

Chapter II: Literature Review

Introduction

Reiterating the purpose of this research, my goal was to analyze teacher discourse during collaborative PD to ascertain how teachers leverage peer collaboration around an expansively-framed curriculum to help them make sense of CS content and implement

CS curriculum. The purpose of this literature review is to delve deeper into the research relating to teacher collaboration, collaborative PD, and expansive framing as tools for teacher learning, especially in regard to computer science curriculum. As such, this literature review will be divided into sections regarding expansive framing, collaborative PD, and teacher learning models using PD. The objectives for this literature review are as follows:

- To describe the current state of the research on the role of discourse in collaborative PD on teacher learning and agency in curricular implementation.
- To describe the current state of the research on the role of expansive framing in teacher sense-making and curricular implementation, particularly in regard to CS content and curricula.
- To discuss the issues, strengths, and weaknesses in previous research.
- To draw conclusions based on this information from which the research questions and strategies for this study were formulated.

Article Selection Criteria

Google Scholar and Utah State University online library resources were used to locate peer-reviewed studies that were published between 1999 and 2020, with a primary emphasis on those that relate to expansive framing, collaborative PD, and their association with computer science curriculum and teacher learning. A variety of search terms were used both singularly and in combination, including, but not limited to expansive framing, collaborative professional development, teacher learning, teacher sense-making, computer science, elementary school, and teacher discourse.

In addition to the keyword search, additional articles were selected based on their value in defining teacher learning in PD settings. These articles were identified through the recommendation of other researchers. In the end, 24 articles were identified and selected for inclusion. Those that were included primarily focused on the role and learning of teachers *and* on computer science education, though some studies with a broader STEM focus were included to provide more depth.

Summary of the Literature

A literature review was conducted on 24 articles using the qualitative analysis software MAXQDA. Articles were coded for discussion of key themes, particularly expansive framing and collaborative PD. This section of the thesis will present the results of this review.

Expansive Framing

The theoretical model underlying the CS curricular unit on which this research is based draws upon a situated account of transfer, called expansive framing (Engle at al., 2012). The model of expansive framing is founded on the idea that making frequent connections between the context of learning and the context of transfer helps learners create a context to assist in knowledge transfer. When students understand the larger, encompassing context surrounding both the learning context and the transfer context, learning is promoted (Engle et al., 2012). Engle et al. identified several specific types of connections that could be made using an expansive framing model, such as *connecting* settings to cue prior knowledge, helping learners to *understand* how skills and practice in one setting can be useful in future settings, and *authoring* and *creating* in new contexts

(Engle et al., 2012). These last two methods are particularly applicable to the computer science curriculum used in this study wherein the students author and create new board game levels in the block-based programming language Scratch.

Expansive framing has been applied by Grover et al. in the context of computational thinking curriculum and assessments (e.g., Grover et al., 2014), as well as by other researchers to design generally appealing curriculum (e.g., Hickey et al., 2020). The model has also been utilized in other STEM curricula, such as a high school biology curriculum enacted by Lam et al. and a one-to-one high school biology tutoring system implemented and studied by Engle et al. (Engle et al., 2011; Lam et al., 2014).

The curriculum from which I draw my research was designed with expansive framing in mind to encourage the use of an unplugged learning environment in a rural school without one-to-one computer access. In particular, the researchers responsible for this curriculum introduced the concept of Expansively-framed Unplugged to describe the curriculum in question and the practice of using an unplugged context to frame a computing curriculum that will eventually be represented in a digital space, such as block-based coding languages like Scratch (Lee & Vincent, 2019). Unplugged activities are canonically defined by Bell et al. to "involve problem solving to achieve a goal, and in the process [deal] with fundamental concepts from Computer Science" (Bell et al., 2009). These activities do not require the use of a computer and are designed to engage students in computational thinking (CT) and demonstrate the value of CT even to those who are not interested in studying computer science further (Bell et al., 2009).

In Engle et al.'s description of expansive framing, the authors mention that it may be beneficial for learners if they are provided with specific contexts in which to apply the content they are learning (Engle et al., 2012). With this in mind, I am interested in exploring how the context in which teachers have been presented with the content—that is, in a PD designed to prepare teachers to apply the content in their classroom implementations—will provide a richer learning experience and allow them to take advantage of the expansive framing context more effectively. There is currently a dearth of research addressing the benefits of expansive framing approaches in curricular design in terms of impacts on teacher learning; thus, this research may help to fill that gap.

Collaborative Professional Development

PD has been used extensively to support teachers in learning new curricular approaches. However, research has suggested that traditional PD approaches may not be effective at encouraging teacher learning and causing lasting changes in classroom implementations (e.g., Ball & Cohen, 1999; Borko, 2004). In response to this need for more effective PD, new collaborative approaches to PD have been developed. This collaborative PD format centers around inviting teachers to be active participants in their own learning and the curricular design process (Borko, 2004).

Examples of collaborative PD include engaging teachers in co-design as Peel at al. (2020) suggested. In Peel et al.'s work, a science teacher participated in a Design Based Implementation Research project implementing CT curriculum into her classroom. The results of the study indicated that the teacher showed increased confidence and understanding regarding the CT content after participating in the co-design process, which led to increased implementation of the content in her classroom (Peel et al., 2020). In a similar vein, Voogt et al. (2015) made the point that co-design PD can be seen through the theoretical lenses of a situated perspective, teacher agency, and the cyclical nature of learning and design. Voogt et al. show that when collaborative PD are analyzed through these theoretical lenses, co-design is an effective tool to encourage teacher learning, the creation of effective curricula, and teacher agency in implementation choice (Voogt et al., 2015).

In addition to PD that focuses on co-design, collaborative PD also often focuses on encouraging teachers to *reflect* on their practice, as is seen in Sherin's (2007) research. In this study, Sherin uses video clubs to research the way that such PD practices impact teacher's professional vision, or their ability to reflect on their professional (i.e., pedagogical) choices. Sherin found that such reflection was increased in this PD setting and that it was beneficial in encouraging teachers to make sense of their pedagogical choices and the content (Sherin, 2007).

Collaborative PD may also focus on *modeling* new curricula, as seen in Goode et al.'s research from 2014. In this study, modeling is found to be a useful tool in teacher's exploration of and development of CS knowledge and CS content implementations (Goode et al., 2014). PDs of this format allow and encourage teacher learning as they are closely linked to teachers' everyday experiences and challenges (Putnam & Borko, 2000). Due to this evidence, the PD design chosen for the curriculum on which this study is based was one of co-design that included dedicated time for reflection and modeling.

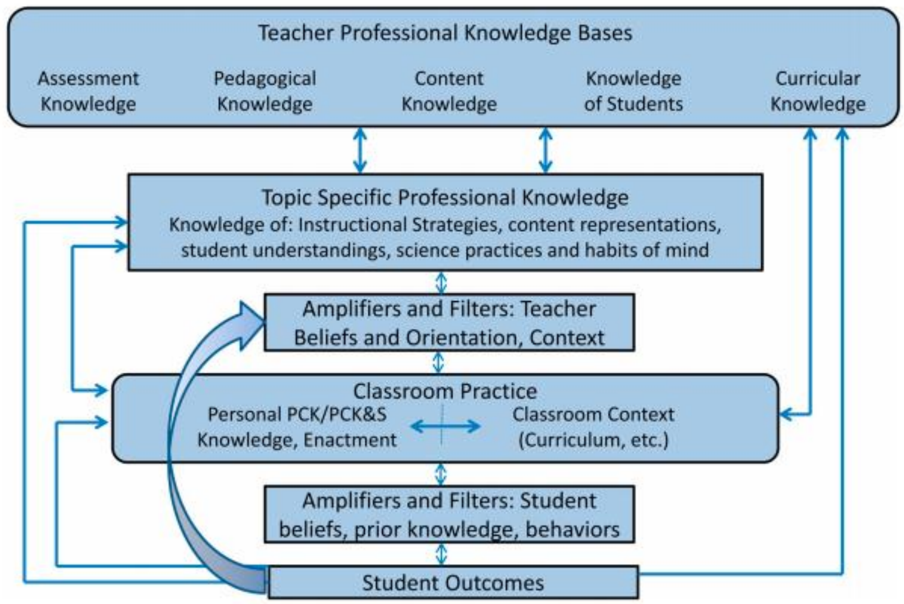
Modeling Teacher Learning in PD

Several authors have used diagrams to represent teacher learning through PD and to model the process of learning and its impact on the classroom. Many of these models reference back to and build off of the objectives of PD as established by Richardson (1996), which are that PD should attempt to foster change in teacher's knowledge, beliefs, and attitudes, as these components have a strong correlation to what teachers do in the classroom and thus influence student learning (see models by Firestone et al., 2020; Fishman et al., 2003; Gess-Newsome et al., 2019; Guskey & Huberman, 2005).

Guskey and Huberman suggest a model of teacher learning that is linear in nature; that is, PD impacts a teacher's classroom practices, which influence changes in student learning outcomes, and which finally changes a teacher's beliefs and attitudes (Guskey & Huberman, 2005). However, many other researchers suggest a more interactive and cyclical model of teacher learning in PD. Gess-Newsom et al.'s cyclical model incorporates the major claim made by Guskey and Huberman, which is that changes in student outcomes are what influence changes in teacher professional knowledge and beliefs (see Figure 1). However, their model does not consider how teacher PD factors into teacher professional knowledge (Gess-Newsome et al., 2019).

Figure 1

Model of teacher learning in PD setting



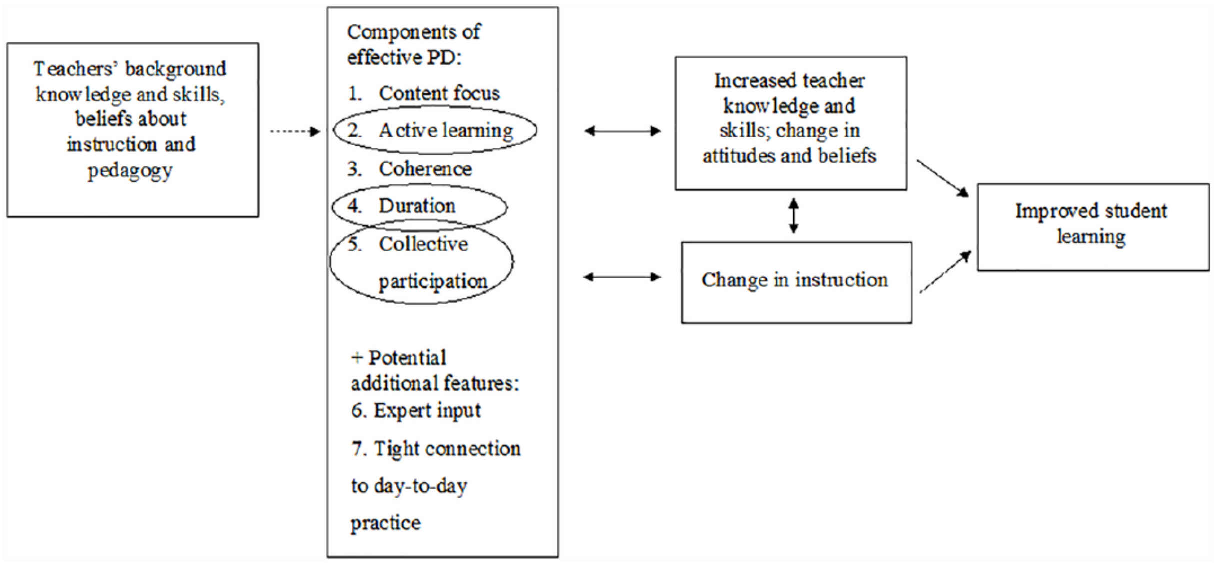
Note. Adapted from Gess-Newsome et al. (2019)

Note that for the sake of understanding Figure 1, "PCK" is pedagogical content knowledge, whereas "PCK&S" refers to the three internal constructs of PCK as proposed by Gess-Newsome: content knowledge (PCK-CK), pedagogical knowledge (PCK-PK), and contextual knowledge (PCK-CxK). Furthermore, amplifiers and filters are forces that influence a teacher's motivations to implement content and practices from the PD. These include the listed examples of teacher beliefs and orientation, context, as well as student beliefs, prior knowledge, and behaviors (Gess-Newsome et al., 2019).

In Fishman et al.'s model and Firestone et al.'s model, changes in teacher beliefs are indeed influenced by the PD, but these changes also impact the PD itself (Firestone et al., 2020; Fishman et al., 2003). Fishman et al. specifically claim through their model that teacher knowledge, beliefs, and attitudes will eventually impact how PD is designed—an idealistic view that suggests the value of PD that is responsive to the teachers (Fishman et al., 2003). Firestone et al.'s model, which is displayed in Figure 2, suggests that changes in teacher knowledge, beliefs, and attitudes more directly impact the active learning process, which is central in these changes taking place in the first place (Firestone et al., 2020).

Figure 2

Model of Teacher Learning as Presented by Firestone at al. (2020)



These various models collectively argue for the value of a collaborative PD model that is responsive to teacher knowledge, attitudes, and beliefs. Furthermore, this collection of models universally recognizes how changes that occur in the classroom as a result of PD—or more directly, as a result of changes in teachers—can also be an influence on how PD and teacher learning proceed.

One area that was rarely expanded on in these models was the type of activities that occur in effective collaborative PD to *lead* to these changes. In fact, Firestone et al.'s model was the only one to include the components of effective PD, which it lists as content focus, active learning, coherence, duration, and collective participation (Firestone et al, 2020). Thus, in this study, I will expand on these existing models to further represent the components of effective collaborative PD.

Extending Collaborative PD and Expansive Framing Research

Collaborative PD and the value of its implementation have been described in many research studies (e.g., Kartal et al., 2019; Steeg & Lambson, 2015; van Oostveen, 2017), including in studies that specifically describe the impact such PD has on teacher understanding of computer science curricula (e.g., Rich et al., 2017). However, the *process* of teacher learning over the course of collaborative PD has been less studied (Walkoe & Luna, 2020). Specifically, research is needed to address how teachers discursively engage with collaborative PD, how teachers make connections to their prior knowledge, and how they take ownership of the new ideas they generate in ways that influence subsequent classroom implementations of curriculum (Walkoe & Luna, 2020). Furthermore, in a study of informal teacher interactions, Horn (2010) found that collaborative and collegial conversations had the ability to support teacher sense-making; however, the study was limited to informal settings and therefore could be extended and deepened to address how teacher conversation in formal collaborative settings impacts teacher learning.

My thesis study addresses these gaps. Specifically, this study examines the role of discourse in a collaborative PD setting with an emphasis on the role that discourse plays in impacting classroom implementations and sense-making. Furthermore, as mentioned previously, due to the current lack of research addressing the benefits of expansive framing instructional approaches in terms of impacts on teacher learning, it is hoped that this research will help to examine how teachers benefit from a curriculum designed using an expansive framing model.

Chapter III: Methodology

For this project, I am using existing data from a prior research study that developed an expansively-framed computer science curriculum and implemented it in two iterative cycles in local elementary schools. The study in question was headed by Dr. Victor Lee, Dr. Mimi Recker, and Dr. Jody Clarke-Midura. A more detailed description of the participants and curriculum will follow. Table 1 outlines the data sources and analyses for each of my stated research questions.

Table 1

Data Sources and Analysis for Research Questions

Research Questions	Data Sources	Analysis
How do teachers discuss new CS content and pedagogy in a collaborative professional development setting?	Audio and transcript files of the PD sessions	Coding of PD transcripts at an utterance and episodic level to identify conversational patterns
What types of teacher discourse about new curricula that occur during collaborative PD impact teachers' subsequent classroom implementations?	• Video and transcript files of classroom implementations • Audio and transcript files of the PD • Teacher interviews	• Coding classroom data to identify changes teachers make • Sorting of changes to identify those that came from the PD and those that did not • Comparing PD data to classroom data to see what translated to the classroom

How does an expansively-framed curriculum support *teacher l*earning?	• Audio and transcript files of the PD • Video and transcript files of classroom implementations • Teacher interviews	• Using episodic and utterance level coding PD transcripts to identify use of the expansive framing model • Use of the above coding to identify all teacher sense-making conversation • Coding of teacher curricular implementations for use of the expansive framing

Setting and Participants

The data for this research is a subset of data drawn from the second iteration of this CS curriculum implementation. This implementation occurred at a rural elementary school in the Intermountain West. The school district in which this elementary school was located has shown significant buy-in with the project and a great desire to emphasize CS curriculum in general in the coming years; the first cycle of the project also occurred in this district.

There were three fifth-grade teachers and one school librarian involved in this study, all of whom met together regularly to attend the PD sessions. The teachers also had described previously working together in collaborative environments. Each teacher and librarian implemented the curriculum independently in their classroom (or in the library) to a class of approximately 25 students each. The teachers first conducted a 20-minute preparatory lesson in their classroom, following which the librarian facilitated a hands-on period of student activity. In the library activity, students played the board game and its Scratch instantiation for the first few weeks. Students then used their time in the library to create and program their own levels of the game in Scratch (see Table 2). For the purpose of this study, I will only be focusing on the implementations of the classroom teachers due to both the more abundant collaborative data that exists between them and the similarities in their implementations of the curriculum. However, the interactions of the librarian in the PD sessions will also be considered.

The teachers' level of experience in the classroom and with CS content differed greatly. Two of the teachers were veteran teachers with limited CS experience, although the older of the two was more vocal about her lack of experience and the related anxieties she felt. The third teacher was a first-year teacher who had some experience with CS and with leading computer-science related educational experiences for students in other contexts.

Curricular Materials

The content used for this CS curriculum was created by a team of researchers with the goal of using board games—what is hopefully a relevant and interesting context to students—to enhance student learning of basic CS concepts. The board game used was titled *//CODE: On the Brink*, published by ThinkFun. In the game, players program a robot to navigate a two-dimensional puzzle. The game consisted of levels, with each level using a different puzzle board and allowing players to create new program combinations. As the levels progressed, new and more complicated procedures were introduced for players to use in their code, although the structure of the levels remained consistent.

Figure 3

CODE:// On the Brink *game*

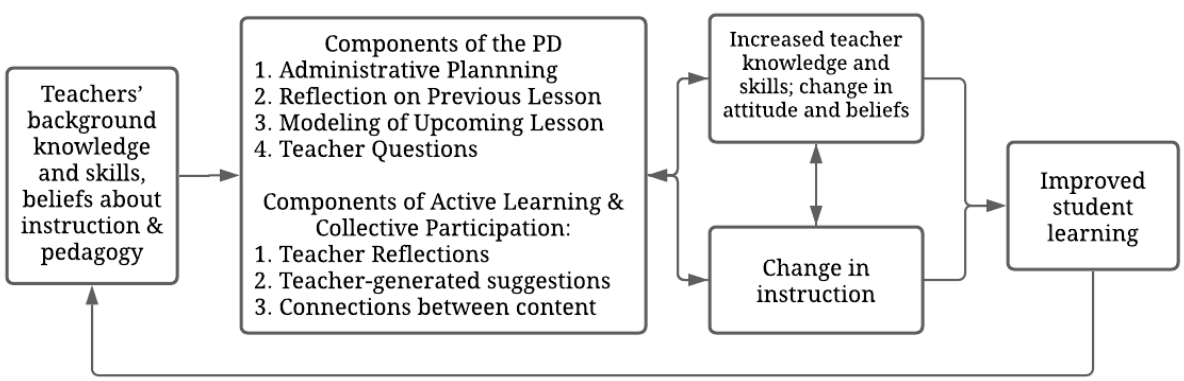
The curriculum consisted of seven lessons, occurring over the classroom and library, which were to be taught over the course of seven weeks, one lesson a week. The classroom portion was to be a preparatory teacher-led lesson taking approximately 20 minutes, while the library portion included hands-on activity playing the game in both its physical (board game) implementation and Scratch, as well as creating and programming student levels of the game in Scratch. The library portion was to take another 30 minutes. Each week, the teachers and school librarian met with at least one researcher for a PD session. These PD sessions were organized with four major parts—a time to address administrative needs, a time to reflect on the past week's lesson, a time to model the upcoming lesson and to evaluate and make adaptations to the lesson, and a final time period to discuss any standing questions teachers may have had about the curriculum or

CS content. In the final four weeks of the PD, one teacher would act as the instructor during the modeling portion, while the other participants would play the role of students.

Figure 4 provides a model of our collaborative PD format, drawing on the models created by Firestone et al. (see Figure 1) and Gess-Newsome et al. (see Figure 2). This model uses the general structure and categories as provided by Firestone et al., while adding the component that improved student learning can and does impact the teacher's background knowledge, skills, and beliefs about instruction and pedagogy—an idea presented by Gess-Newsome et al. (2019). Furthermore, this model adds to the questions that remained in Firestone et al.'s model by providing components of PD that described active learning, an area that Firestone et al. described as underdeveloped (Firestone et al., 2020). Note that as opposed to Firestone et al.'s generalized list of the components of effective PD, I have included a list of the components of our PD as it actually occurred, listed in the chronological order in which they occur.

Figure 4

Format of Collaborative PD Used in this Study



Note. Based on Firestone et al. (2020); Gess-Newsome et al. (2019)

Each of these "Components of Active Learning and Collective Participation" is drawn from the literature that was presented previously. Specifically, reflection was influenced by Sherin, who noted the value of reflection in teacher learning, while suggestions are a result of the value of co-design, as seen in the work of Peel et al. and Voogt et al. (Peel et al., 2020; Sherin, 2007; Voogt et al., 2015). Finally, the inclusion of "connections between content" is influenced by the expansive framing model.

For a more complete view of what occurred each week during the curriculum and the PD, refer to Table 2. This table includes a description of each lesson and its main learning goals, both in the classroom and library portions, as well as a description of the focus of the associated PD.

Table 2

Description of PD and lesson content and schedule

Week	PD Content (45 mins)	Setting	Description	CS Concepts
1	Played the board game and discussed strategies for presenting it.	Classroom	Introduce computer programming and the board game.	Defining programming
		Library	Students play the board game in pairs. (Levels 1-10)	
2	Discussed the rules of the game in terms of conditionals and played the game in Scratch.	Classroom	Review board game rules and mechanics. Introduce scratch by playing the game in scratch.	Conditionals
		Library	Students play the Scratch instantiation of the board game (Levels 10-20) in pairs and receive a review of conditionals.	

3	Discussed debugging strategies to solve game levels and how to introduce new procedures.	Classroom	Review Scratch environment, introduce CS concepts, and demonstrate step-by-step programing.	Procedures, algorithms, and abstraction
		Library	Students play the scratch instantiation of the board game (Levels 20-30) in pairs.	
4	Discussed the definition of conditionals and how to build them in Scratch.	Classroom	Demonstrate how to build new conditionals into the game in preparation for students building their own levels.	Conditionals, debugging, simulation
		Library	Students build paper prototypes of their own board game levels in pairs.	
5	Discussed how to alter the Scratch stage and sprite programming.	Classroom	Review how Scratch reads code and model how to transfer the paper prototypes to Scratch.	Debugging, simulation
		Library	Students transfer paper prototypes to Scratch.	
6	Discussed how to build new procedures in Scratch.	Classroom	Demonstrate how to build multiple types of procedures to add new game mechanics that were introduced in paper prototypes.	Procedures, abstraction, debugging
		Library	Continue transferring paper prototypes to Scratch and test/debug levels.	
7	Discussed CS concepts as they related to the game and how to share student levels. Also discussed their feedback for the project.	Classroom	Teachers review CS constructs learned in the project by illustrating where they appeared in the board game and Scratch. Play classmates game levels in Scratch.	Conditionals, abstraction, procedures, and debugging
		Library	Final post survey	

Data Sources

For this study, I used two data sources to capture the full picture of the trajectory of the teachers' experiences through the PD and their own classroom implementations across the length of the curriculum. These data sources were audio files and transcripts of the PD sessions and video files and transcripts of the seven classroom implementations for each teacher.

Professional Development Transcripts

My first source of data is transcripts made from audio recordings of the PD sessions from the implementation of the CS curriculum. This data was collected by the attending researchers using audio recorders and then transcribed by myself using Otter transcriptions services. For this project, researchers met weekly with the participating teachers and librarian for a period of seven weeks during the implementation of the CS curriculum. Coding teacher discourse during these PDs provides insight into the educators' CS understanding, pedagogical approaches, and perspectives on the curriculum, among other observations.

The coding for these PD transcripts took place in four stages, with two focusing on an utterance level of coding and two utilizing an episodic lens. I first coded teacher discourse at an utterance level for the type of discursive statement being made, such as a question, suggestion, or reflection, among others. These inductively determined codes allowed me to identify the structure of conversation occurring in the PD and develop a detailed picture of what kind of comments teachers are making and what the purpose of those comments is. I then coded teacher discourse at an utterance level for what the topic of the previously coded discursive statements was. An example of this would be coding a question as “Pedagogical” due to it being a question about a pedagogical topic. Only lines identified in the first pass of coding were considered for the second pass, thereby ensuring that each discursive statement is given a topic and each topical statement is classified by what role it played in the discourse. A more thorough description of the coding scheme I utilized can be found in Appendix A.

The third stage of the coding was to group the utterances into discursive statements framed by a unifying topic or theme. Each episode delineated a major topic of conversation in the PD, such as a conversation about how to teach new board game procedures to students in an active manner. The types of episodes developed included suggestion episodes, reflection episodes, and connections episodes, each developed around a type of discourse as identified in the first pass of coding. Additional episodes were identified inductively as analysis continued, with episodes focusing on how teachers collaboratively discuss the CS content, pedagogical approaches, the lesson plans, experiences in the classroom during the implementation of this project, and the expansive framing. The fourth and final stage of coding involved analyzing these episodes specifically for key features, such as triggering events that cause the episode to occur, types of teacher participation within the episode, or unifying topic of the episode. The purpose of coding again at an episodic level after utterance-level coding has occurred was to help categorize teacher discourse in more meaningful and observational components, which could be used to provide a better classification of the ways that teachers collaboratively learn and plan in PD settings.

Classroom Implementation Transcripts

My second source of data was transcripts and video recordings of teacher implementation of the CS curriculum in their classrooms. This data was collected by the attending researchers using video camera recorders, and then transcribed by myself and other researchers using Otter transcriptions services. Recordings and transcripts of each of the three teachers' implementations of all seven weeks of the CS curriculum were included, for a total of 21 approximately 20-minute lessons. Due to the nature of the librarian's participation, which did not include substantial instructor-led activity, video and audio recordings of the students' participation in the library were not included. By coding and analyzing teacher classroom implementations, I gained insight into the extent and nature of individual educators' use of the PD collaborative conversation and expansive framing to teach the CS concepts and adapt the curriculum in their classroom.

The coding for the teacher classroom implementation data occurred in connection with the analysis that occurred with the PD data. As episodes with the potential for implementation were identified in the PD, I coded the classroom implementations to determine whether these episodes were manifested in the teachers' in-class actions.

Coding also included identification of instances wherein teachers referred back to connection episodes that were made in the PD. This consisted of searching the transcripts that followed a connection episode's occurrence for times when the same subject matter was discussed in the classroom. When found, these discussions would be compared to the connection episode and to other times teachers had discussed the same concept prior to the connection episode's occurrence. For example, if teachers made a connection episode comparing procedures and algorithms in the fifth PD, I read through the transcripts for lessons five, six, and seven for each teacher to see if they talked about procedures or algorithms in those lessons. If such instances occurred, I compared them to the connection episode to see if the discussion in the classroom was similar to what occurred in the PD, and also compared the discussion in the classroom to any discussions about the same topic that occurred in the classroom prior to the connection episode being made. This allowed me to determine whether the connection episode had any impact on how teachers were discussing these topics in the classroom.

Analysis

To analyze the data for this research, I first coded the PD transcript data, as described previously. The process of coding PD data at an utterance level was conducted in tandem with the assistance of an additional researcher to ensure reliability and guarantee that no portion of the transcript was overlooked. The additional researcher worked with me on this coding process for the first two PD sessions, after which saturation of utterance types had been established. I then grouped utterances into episodes and coded at an episodic level, as described previously. I performed this coding using a thematic analysis approach, as established by Braun and Clarke (2006). After completing this level of coding, I compared my results with another independent coder to generate reliability data using Cohen’s Kappa values. To do so, I developed a coding scheme with definitions and examples and trained the independent coder to code the data, after which he independently coded all data. These Cohen’s Kappa values are included in Table 3 below.

Table 3

Cohen's Kappa Values Related to Sense-Making Episodes

	Cohen's Kappa Values	
Type of Episode	Episode Triggering Events	Episode Topic
Suggestion Episodes	0.78	0.81
Reflection Episodes	0.75	0.77
Connections Episodes	0.80	0.95

Upon identifying episodes with the potential for classroom implementation, I coded the classroom transcript data to identify the implementation of these episodes from the PD in the classroom. This was done using a thematic approach. Finding these episodes and identifying their transfer into the classroom helped me to answer my first research question regarding peer collaboration and its effect on classroom implementation. This occurred particularly as I identified the nature of collaborative episodes that were implemented.

Finally, I coded for teachers' use of expansive framing within their implementations to answer my third research question. To do this, I used a Concept Coding approach as defined by Saldaña (2015) in *The Coding Manual for Qualitative Researchers* (pg. 119), looking for any reference in the classroom transcripts that could be aligned with the expansive framing approach built into the project.

For this coding project, Otter transcription software was used to produce transcripts of the audio and video recordings. I also decided to use MAXQDA as a platform for storing the codes. This allowed me to organize the data and change code names as my understanding of the data evolved. Furthermore, it allowed me to build and revise my code book as the analysis developed, as well as use built-in memos and color coding for my coding analysis. In addition, I also used word processing such as Microsoft Word and Google Docs and spreadsheet software such as Microsoft Excel and Google Sheets for further organization of data and to share and aggregate data with other researchers.

Chapter IV: Results

I will organize the results for this thesis according to the research questions around which this study was framed. The first research question to be discussed is *How do teachers discuss new CS content and pedagogy in a collaborative PD setting?*

RQ1: Teacher Discussions in Collaborative PD

As established in the methods section of this thesis, three elementary teachers and one school librarian participated in seven collaborative professional PD sessions to accompany this CS curriculum. Each session lasted approximately 45 minutes and included a structure wherein teachers would review the previous week's lesson and experiences, model the upcoming lesson, and ask any questions or make any suggestions that they had about the lesson. During the four final PD weeks, teachers took turns leading the modeling portion as the instructor while the other teachers and librarian would fill the role of students. Each teacher filled the role of instructor at least once.

During these PDs, teachers directed much of the conversation that occurred. As described in the Chapter III "Analysis" Section, I used a thematic analysis approach to inductively build a code book describing the topic and discursive purpose of teacher conversation at an utterance level. As this process was undertaken, I discovered that many of the teachers' comments could be construed as administrative—that is, comments where teachers are organizing meetings or talking about other obligatory but otherwise non-educational topics. These comments, as well as off-topic comments made that were unrelated to the PD, were excluded from the analysis. An example of administrative content would be when teachers planned the next meeting place for the PD sessions; an example of off-topic conversation would be a discussion about how teaching the day after Halloween is difficult and the day should be a school holiday.

Once administrative and off-topic comments were excluded from the analysis, I classified statements by what role they played in a conversation (e.g., asking a question, modeling) and what they were talking about at the time (e.g., CS content or pedagogy). Thus, a statement might be a question about CS content or a modeling of a pedagogical practice.

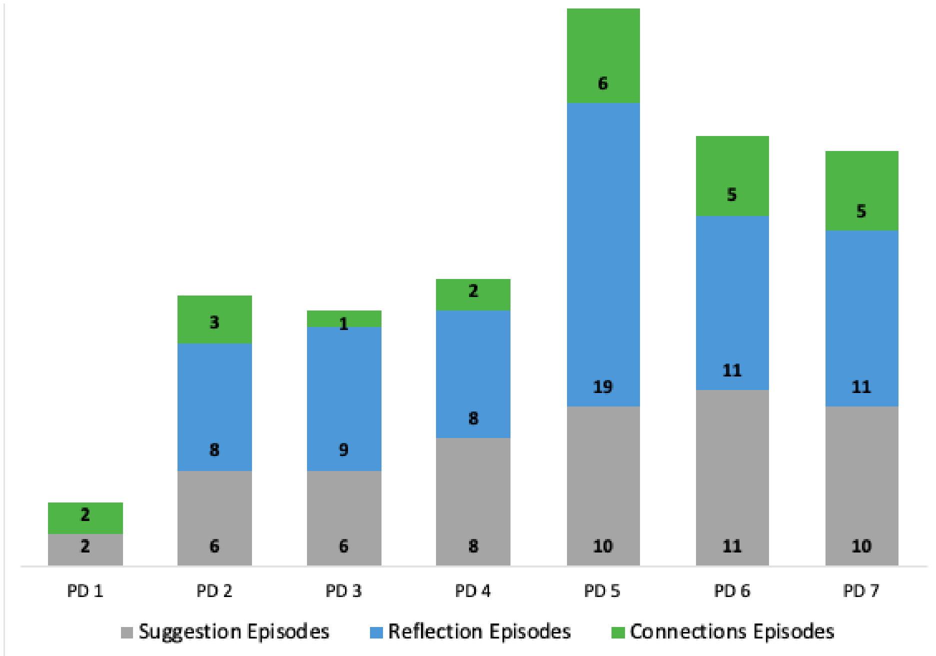
As this process was concluded, I determined that the primary types of statements that teachers made were suggestions, connections between various content topics, and reflections. Each of these types of utterances occurred 149 times, 80 times, and 216 times, respectively.

These utterances were grouped into episodes as described in the Chapter III "Analysis" Section. I characterized these episodes based on the triggering events that caused them to occur and their topic. The reasoning behind the use of these episodes was to delineate occurrences of teacher participation in the PD and to classify how that participating was occurring in one of three key and meaningfully different ways. By tracking the occurrence of these episodes, I determined how teachers' participation was changing over the course of the PD, what triggered these episodes, and what topics they focused on in the episodes.

Based on the number of episodes made, I found that teacher participation in the 45-minute PD sessions increased over the course of the seven-week period. As can be seen in Figure 5, there was a 60% growth in the number of sense-making episodes that occurred during the final three PD sessions (N=88 episodes) compared to the first four (N=55 episodes). The number of connections that teachers made grew particularly, with twice as many connection episodes occurring in the last three weeks (N=16 connection episodes) as occurred in the first four weeks (N=8 connection episodes). Note that those final three weeks were occasions where the teachers, rather than the researchers, were modeling the lesson. This also occurred in the fourth week, as previously described; however, only two of the four participants were in attendance during the fourth week of PD, whereas all participants were there for the final three weeks.

Figure 5

Frequency of Sense-Making Episodes by PD



Structure of a Sense-Making Episode

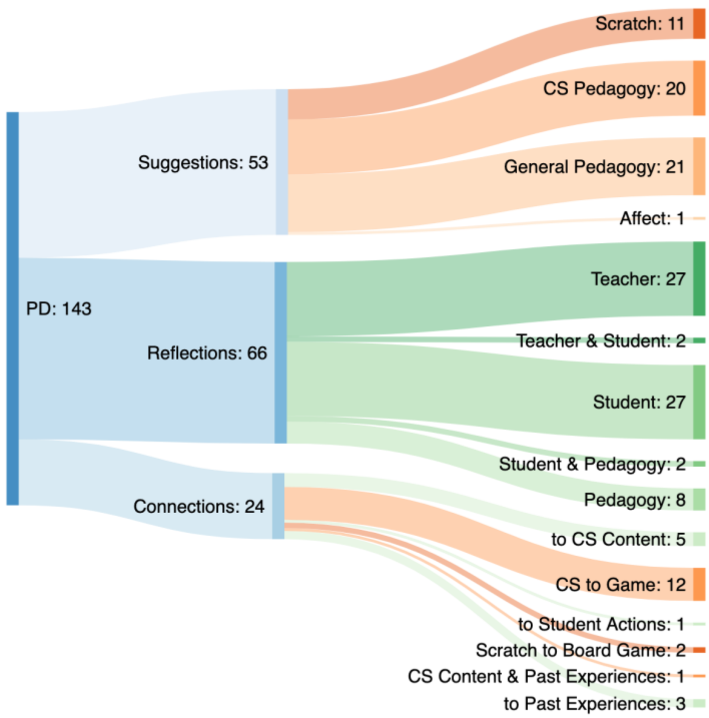
I coded each sense-making episode to consist of 1) a *triggering event*, or an utterance which instigates the sense-making conversation of which the episode is comprised; and 2) a *topic*, or the overall theme of the sense-making episode. An example of a triggering event might be a question asked by either a teacher or a researcher. The topic of an episode might be CS pedagogy or students' affect.

The types of triggering events and topics differ for each type of sense-making episode. This is due to the inductive nature through which these episodes were coded and the different nature of each type of episode. For example, a suggestion for an upcoming lesson might be triggered by a reflection on a past lesson. However, a reflection is not going to be triggered by itself.

Figure 6 provides a Sankey diagram of the frequency of all sense-making episodes that occurred in the PD and the frequency of their topics. Note that each type of sense-making episode corresponds to a certain set of topics—there are no shared topics between any two types of sense-making episodes. This is simply due to the fact that suggestions, reflections, and connections typically address different types of content and would thus naturally have different associated episode topics. As each episode type is discussed in further detail, the triggering events and topics most common to that type of episode will also be discussed.

Figure 6

Sankey Diagram of the three Sense-Making Episodes and their topics



Suggestion Episodes

Suggestion episodes occurred as teachers recommended changes to the lesson plans of the curriculum. These episodes centered around at least one suggestion utterance, as described in Chapter III. While some of the episodes contained only one suggestion made by a single teacher, many contained multiple suggestions made by multiple teachers as they engaged in conversation about a particular topic, such as how to teach the concept of a computing procedure to students. The interactive nature of these multi-suggestion episodes was actively encouraged by the collaborative activities designed into the PD.

Throughout the seven PDs, teachers generated a total of 53 suggestions episodes (see Figure 5). In total, teachers made 41% more suggestion episodes in the final three weeks of the PD, when the format of the PD was more teacher-led and teacher-centered.

Suggestion Episode Examples

An example of a suggestion is included below. In this example, Maria, one of the teachers, is modeling the upcoming lesson plan as though she were talking to her students. The librarian, Julie, makes a comment reflecting on what students have said to her in the past and how they would therefore likely respond. Taking this into account, Maria then incorporates the suggestion seamlessly into her modeling, recommending that the teacher have students demonstrate their personal definitions of a procedure to show the importance of precision when defining computational procedures.

> **Maria:** So, if you made your own card, like I know, Macey [a student], you did a hop forward. I don't know what hop forward means. I'm the sprite, so you're gonna have to tell me. So, we're gonna have to make our own procedures right here for what it means to hop forward.
> **Julie (Librarian):** "Well, it means just hop forward," is what they [the students] said to me.
> **Maria:** Okay, Macey, what does the hop—hop forward—mean to you. Think it in your head. Jensen, what does it mean? Ok, what does it mean, Macey? She tells me. Jacob [a student], tell me. Oh, so like this, or like this? Which one? Who's right? How do I know?
> **Debbie:** Okay, that works.
> **Teresa:** Good job.

Another example of a suggestion episode occurred in the sixth PD session. In this episode, Debbie reflects on an issue she had seen in her class regarding how difficult it was to program the Scratch sprite's location so that it was exactly centered in the game board squares (a necessary step to ensure the Sprite did not get off course by beginning in an incorrect position by a small amount). This reflection was followed by suggestions she made to change the size of the sprite to fix this problem.

> **Debbie:** So—so I found that with my class too. Once your sprite moves, one of your movements, if even—if your mouse is just a little bit off… The minute it touches a side, it stops.
> **Maria:** So, they're gonna have to get it really good on their very first one?
> **Teresa:** Right.
> **Debbie:** Can you make the mouse…Was there a way that we can make—
> **Maria:** —smaller?
> **Debbie:** The sprite smaller?
> **Researcher:** Yeah.
> **Debbie:** So, then it doesn't happen as much. Maybe that would help the kids.
> **Maria:** Yeah.
> **Debbie:** If they knew... because his tail gets in the way when he does a turn like on that last time.
> **Teresa:** Yeah.
> **Maria:** Oh, there you go.
> **Teresa:** Oh, perfect.
> **Debbie:** Yeah, see, so now it would be less likely to hit. Okay.

One final example of a suggestion episode occurred in the second PD session, when the teachers were discussing what conditionals were and whether to call the CS concept an "event" or a "condition." This suggestion was triggered by the researcher modeling the lesson for a moment before the teachers suggested ways that it could be improved.

> **Researcher 1:** […] So, programming is like giving instructions. And you guys have learned that the term "events and results." Well, they are the same as "conditions and results." So, for example, if the robot is standing on the blue color, then he's going to do the blue block.
> **Maria:** Do you want us just to call it "conditions" from the beginning?
> […]
> **Researcher 1:** We've gone back and forth on this so much—of "events" versus "conditions."
> **Maria:** What is it called in, like, technical terms—or is it both?
> **Researcher 1:** It's—
> **Debbie:** Neither.
> **Researcher 2:** It's a conditional, right?
> **Researcher 1:** It's a conditional. It would be a conditional.
> […]

Maria: If we're going to call it 10 minutes later, a conditional, then we might as well—
Researcher 1: Just call it conditionals.
Teresa: Right, because they're really…they're both foreign words to them right now. And so, so you might as well call them—
Researcher: Yeah, then we can call them conditionals from the beginning. I think that's totally fine.
Maria: Like, "Here's a condition. What's the condition outside? It's raining, that's a condition." So, you could—
Researcher: And if it's condition that I'm… "If it's raining, I'm going to use an umbrella."
Debbie: I like that.

This suggestion episode included a more detailed discussion of CS content and also was implemented in the classroom by at least one teacher.

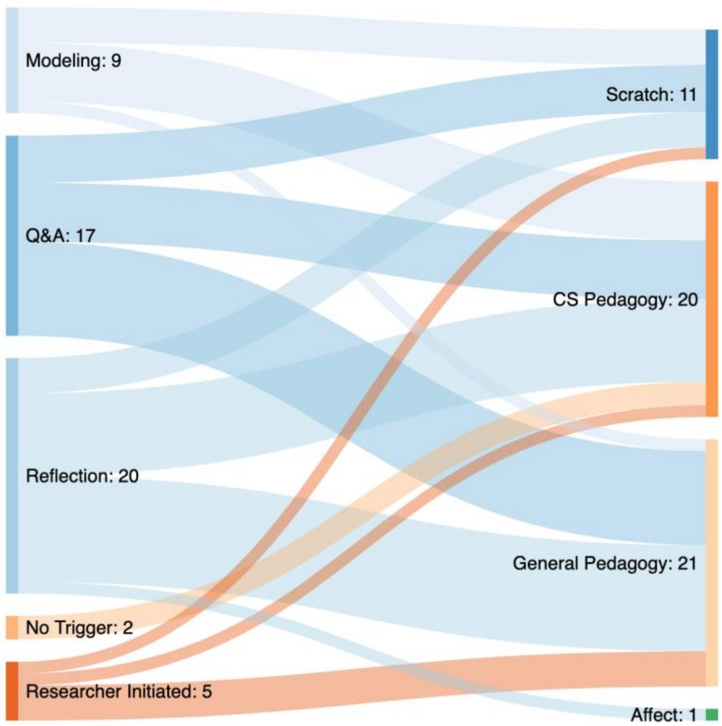
Suggestion Episode Structure

Suggestion episodes were triggered by questions, reflections, modeling activities, and researcher comments. Two suggestions occurred in such a way that there was no trigger, (Both occurred when a teacher suddenly changed the subject to discuss an unrelated suggestion.) The Sankey Diagram in Figure 7 shows the frequency of suggestions topics (on the right side), as well as the corresponding frequency of trigger types (on the left side). As seen in Figure 7, peer questions and peer reflections triggered a majority of suggestion episodes, accounting for 37 of the 53, or approximately 70 percent. This is indicative of the value of teacher conversation and peer collaboration in developing the autonomy to suggest making changes to the lessons. Modeling, a teacher-centric activity, triggered another 9 of the suggestion episodes.

Suggestion episodes covered a variety of topics—the use of Scratch, general pedagogy, CS-specific pedagogy, and improving student and teacher affect. Of note is that 41 suggestion episodes were considered pedagogical in nature, with 20 being related to CS pedagogy particularly.

Figure 7

Sankey Diagram of Suggestion Episode Triggers & Topics



Note. Trigger frequency is shown on the left and the frequency of suggestion episode topics is on the right.

In a later section of this thesis, I will describe how suggestions were used in classroom implementations of the lessons by the three classroom teachers.

Reflection Episodes

Reflection episodes occurred as teachers reflected on past lessons and experiences in the classroom. These usually occurred in a time period set aside at the beginning of the PD session in which teachers were asked to reflect actively on how the curriculum implementation was going. While some of the episodes contained only one reflection made by a single teacher, many contained multiple reflections made by multiple teachers as they engaged in conversation about a particular topic, such as how a particular step in a lesson went.

Throughout the seven PDs, teachers generated a total of 66 reflection episodes (see Figure 5). As can be seen, the number of reflection episodes that occurred any given week was roughly the same, ranging from 8 to 11, except for during week 4, in which 18 reflection episodes occurred. As is to be expected, no reflection episodes occurred during the first PD, as that session occurred prior to the first lesson and therefore did not include anything teachers could reflect about.

Reflection Episode Examples

An example of a reflection episode is included below. In this example, teachers discuss how they were nervous about teaching the first lesson but found that their students caught onto the material quickly and were able to understand the rules of the game easily. The general consensus from this reflection episode example is that the students were doing well and that the lesson was successful.

> **Julie [Librarian]:** They must have got it because when they came in here, they already knew. I didn't even have to do any explanation.
> **Debbie:** And I was worried I was gonna mess it up. And they literally, like, my kids were, like, on it. They were like, “Do this, do that.”
> **Maria:** They wanted to tell me before, but I did the first level, like all by myself, they were like, “Oh, we already know.”
> **Debbie:** And mine worked out perfect. I had a kid that made a mistake with the robot on the second one. And I was like, “Perfect. You went right into what I need.” And I was like, “Oh, look, my robot just went off the board.” He's like, “Oh, yeah, you need to turn left first.”
> **Teresa:** You're stuck there forever now.
> **Debbie:** Yeah. And I was like, “Hey thanks. I'm glad you made the mistake for me.”

Another example of a reflection episode occurred in the fifth PD session. In this episode, Debbie, one of the teachers, offers her opinion about how girls seem to be doing well in and enjoying the curriculum. The other teachers share their insights about how the pairs of students have been working together and offer up suggestions about how problems they have noticed and are reflecting on could be addressed.

> **Debbie:** I think it's opening it up to girls too, like, I know that I've heard that the girls aren't into this [CS] sometimes as much. I've heard that from different people that, you know, the job, whatever, like, girls aren't into it as much. And I've noticed a lot of my girls have enjoyed it too, I think, or, like, are really getting it and seeming to enjoy it. So.
> **Julie [Librarian]:** It is interesting in the library though, when you have boy-girl pairs, it does feel like a lot of the times the boys run the pair, not in…not in a domineering way. It's just an observation. I've just been…it's been interesting to see that.
> **Teresa:** It would be really neat to just have girl pairs, because I've heard when girls pair with each other in science and math, they do a lot better than if you pair them with a boy.
> **Julie:** That might be something to consider if you do this again.
> **Researcher:** Yeah, yeah.
> **Julie:** When you're pairing kids up.
> **Debbie:** Most boys just seem to be more dominant a lot of times, they just…just the…in general—
> **Maria:** Well girls learn to [be]—
> **Researcher:** Passive.
> **Debbie:** Yeah, yeah, exactly.
> **Researcher:** No, totally.

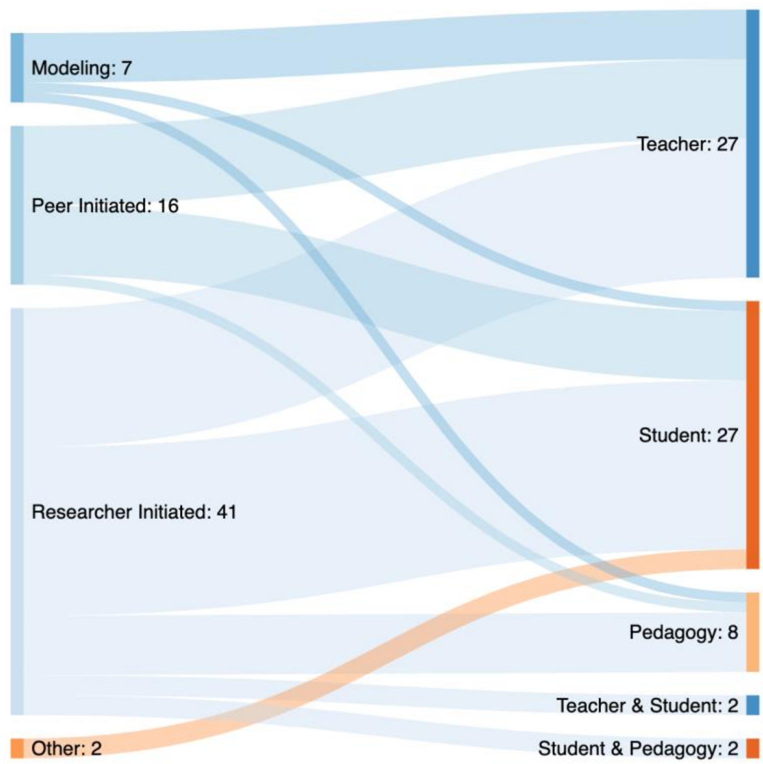
Reflection Episode Structure

Reflection episodes were triggered by modeling, suggestions, peer comments or questions, and researcher comments or questions. In addition, one reflection episode included no triggering event, as the reflection to begin the episode occurred unprompted and was unrelated to the previous conversation. The Sankey Diagram in Figure 8 shows the frequency of types of reflection episodes by topic (on the right), as well as the corresponding frequency of trigger types for each reflection topic (on the left side). As seen in Figure 8, researcher comments were the most common triggering event for reflection episodes, accounting for 41 of the 66 episodes. This is likely due to the structure of the collaborative PD session, where a researcher would often start the session by asking teachers how the previous week's lesson implementation had gone. After being prompted to reflect by the researcher, the teachers would then take turns talking about their week and discussing similar observations, issues, or successes that they had. This period of reflection would continue for several reflection episodes in a row, each focusing on a different conversational topic, but all prompted by the researcher's initial questions.

Reflection episodes covered a variety of topics—teacher-related reflections, student-related reflections, and pedagogical reflections. Teacher-related episodes were divided into three subcategories: episodes reflecting on teacher affect, teacher actions, and teacher understanding. Student-related episodes were also subdivided into three parallel categories: student affect, student actions, and student understanding. Of the 66 episodes, 31 were categorized as discussing student-related topics, while 29 involved teacher-related topics. Only ten focused on the pedagogy itself. Notice that some episodes covered multiple topics concurrently and are thus double coded. Overall, teachers spent more time reflecting on students than any other topic, and particularly focused on student understanding, which was discussed in 18 episodes. The next most common category was teacher action, which was discussed in 11 episodes.

Figure 8

Sankey Diagram of Reflection Episode Triggers & Topics



Note. Trigger frequency is shown on the left side and the topics (and frequency) of reflection episodes are on the right.

Connection Episodes

Connection episodes illustrate how teachers make connections between the CS content and other content. The connections focused on their newfound understanding regarding programming and the Scratch interface, and often drew on past experiences, prior understandings, and the board game used to frame the curriculum. These also were occasions of teacher learning, as teachers spoke out loud about the ways that they were understanding the material being taught.

Throughout the PDs, teachers generated a total of 24 connection episodes, (see Figure 5). While the number of connections made was consistently lower than other sense-making episodes, they increased drastically between weeks 4 and 5, roughly doubling for the remaining three weeks—again, weeks in which the PDs were more teacher-led.

Connection Episode Examples

An example connection episode is included below. In this example, teachers discuss what they think that an abstraction means and begin to differentiate between computational procedures and abstraction in a computing context. They connect the concepts to “Elevate,” the district-authored writing curriculum that they use with their students to provide an analogy to explain their thinking. Both teachers present at the time are fully involved in the conversation, which is representative of the typical connection episodes which tend to be longer and more in-depth conversations between all teachers.

> **Maria:** I mean, I don't think I knew the word “abstraction.”
> **Teresa:** The kind of all, I mean, is like, well, the procedure isn't a small abstraction.
> **Maria:** Yeah, maybe I was actually like, so procedure was like all the steps, right? And abstraction is like the act of doing it. Maybe I wasn't, I don't know if I taught that completely correctly.
> **Teresa:** You could have really big abstractions—
> **Maria:** Well, cuz like I said, like I gave a similar example about the sharpened pencil, like, when I say get out your Elevate stuff, we know that means get out this, this, this. Start your language sheets, like you don't have to say all those steps.
> **Teresa:** Right.
> **Maria:** So, I kind of said that, but—
> **Teresa:** Yeah, and the algorithms are all the instruction—
> **Maria:** And those are the procedures, the abstraction is like the act of making it smaller?
> **Researcher:** Yeah.

A second example of a connection episode occurs in PD 2, when the researcher is modeling a part of a lesson about computational events to the teachers. The teachers think about how their students would understand the concept of events and then use the fact that they think students wouldn't understand this topic well to develop a more comprehensive explanation for what a computational event is. They do this by connecting the term to the events in the board game as the students would understand it thus far.

> **Researcher**: So, level 15 would be right here. Okay. Right. And so, looking at level 15, we would say, "What are some of the events that you see?" And there's two ways to answer this, right? One of this we could say—actually, we'll just have you guys—what are the events that are on here?
> **Maria:** I think as a kid, I would be like, "Wait, what do you mean, like what—"
> **Julie:** Yeah, I wouldn't understand what the word "events" means either.
> […]
> **Debbie:** Well, if it's, like what you're saying with the games, are you meaning, like, when I land here, I need to turn left? Is that what you're wanting as an event?
> […]
> **Teresa:** Or this guy has to turn right or left before he—
> **Maria:** Yeah, they're gonna start telling you what to do, I think.
> […]
> **Debbie:** And is that considered an event?
> **Researcher:** Yeah, I mean, that would be an event, and probably what I would consider is that if you're on red, do the red cards.
> **Maria:** So, landing on red is, or being on red, is the event.
> **Researcher:** Yeah. So being on red is the event.

This example shows how teachers leaned on the board game to make some of the connections that they drew on in the PD sessions. In later sections of the findings, I will go into greater detail about how the board game benefitted the teachers' learning.

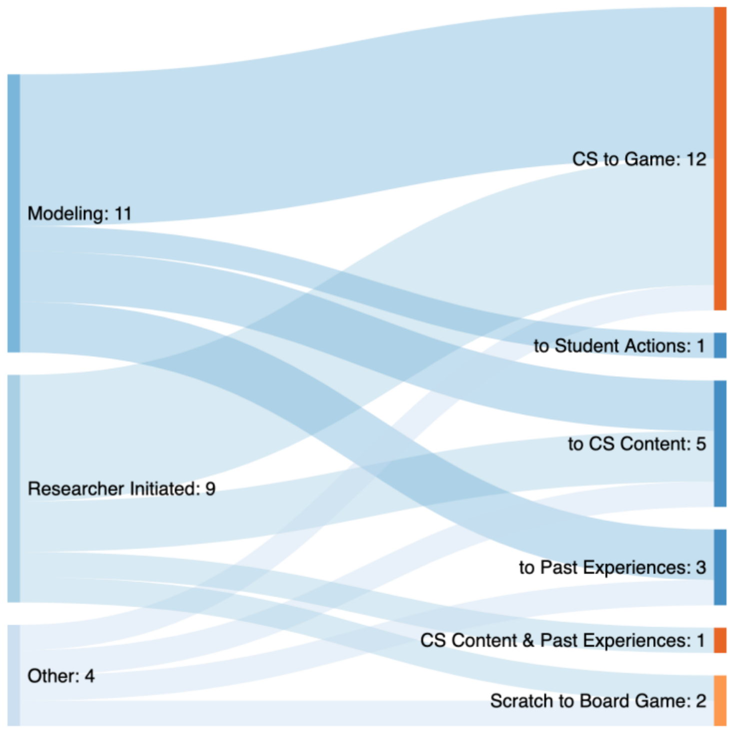
Connection Episode Structure

Connection episodes were triggered by researcher comments or planned connection events in the lesson plans, question and answer sessions, reflection episodes, and modeling activities. The other topics for connection episodes were connections of CS content to other CS content, connections related to student actions, connections between the tabletop game and its Scratch version, and connections to teachers' past experiences.

The Sankey Diagram in Figure 9 shows the frequency of topics of connection episodes (on the right), as well as the corresponding frequency of trigger types for each connection topic (on the left). Modeling triggered 12 of the 24 episodes. This suggests the value of the curriculum in encouraging connections. As can be seen, exactly half of the episodes that teachers generated involved comparing CS content to the board game, suggesting the value of the expansive framing approach being used. Many of the remaining episodes were analogies meant to help relate the content to every-day occurrences, like the weather or classroom procedures.

Figure 9

Sankey Diagram of Connection Episode Triggers & Topics



Note. Trigger frequency is shown on the left and frequency of connection episode topics is on the right.

RQ2: Discourse during PD and Classroom Implementation

The second research question I address in this thesis is: *What types of teacher discourse about new curricula that occur during collaborative PD impact teachers' subsequent classroom implementations?*

While the discussions in the PD have value in showing how teachers make sense of the CS curriculum, there is additional value in examining how they impact what teachers do in the classroom and thus ultimately impacting students. For this reason, I analyzed how the sense-making episodes in the PD translated to classroom implementation choices.

In this context, an episode “translating” to the classroom occurs when the content of a suggestion or connection sense-making episode was used by at least one teacher in a later classroom implementation of the CS curriculum. This means that they chose to use the suggestions being made in the PD as they taught in the classroom, or they referenced a connection that had been made in the PD when teaching.

It should be noted that reflection sense-making episodes did not translate into the classroom, because reflection episodes focused on things that had already happened in the classroom, rather than on things that could be implemented in the future. While some of the reflection utterances made triggered suggestion episodes, the reflections themselves cannot be implemented in the classroom. Instead, I will focus on how suggestion and connection sense-making episodes translated into the classroom.

Classroom Implementation of Suggestion Episodes

I found that of the 53 suggestion episodes that occurred during the PD, 33 of the suggestions (62%) had the potential to be implemented by teachers in future lessons. This potential for implementation occurred when an episode was directed at the teachers, not the researchers, and applied to an upcoming lesson rather than a previous lesson. The other 20 episodes included suggestions that referred to previous lessons or required significant changes that the teachers could not implement within the time constraints. In addition, some of these suggestions were directed at the researchers and were not things that could be done by the teachers within the scope of this iteration of the project (such as adding new lessons). The Sankey Diagram in Figure 10 shows this breakdown.

Figure 10

Sankey Diagram Showing the Usability of Suggestion Episodes

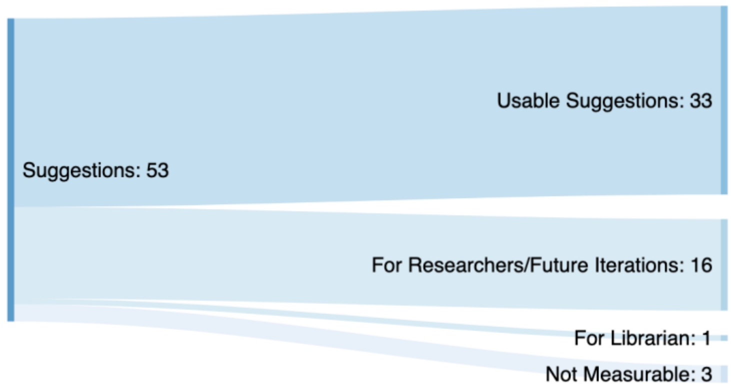


Figure 11

Frequency of How Usable Suggestion Episodes were Implemented (or Not) By Teachers

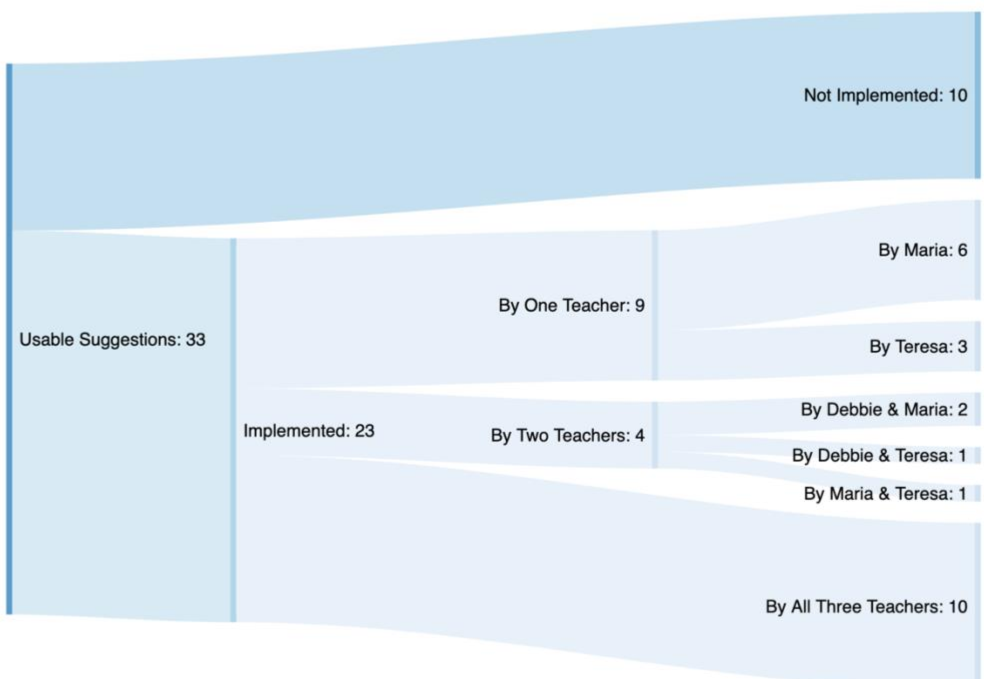


Figure 11 provides a visual representation (Sankey diagram) of how suggestions were taken up by individual teachers, out of all the suggestions that were usable. Of the 33 suggestions that teachers could implement, 23 (about 70 percent) were implemented by at least one teacher. Fourteen of these were implemented by two of the three teachers, and 10 were implemented by all three teachers. There were 9 of the 23 suggestions that were implemented by a single teacher only (6 by Maria, and 3 by Teresa). Thus, it is clear that the teachers did not have the same criteria about which suggestions should be implemented in their classroom, but rather chose individually what they felt would be most effective. Table 4 displays this data succinctly.

Table 4

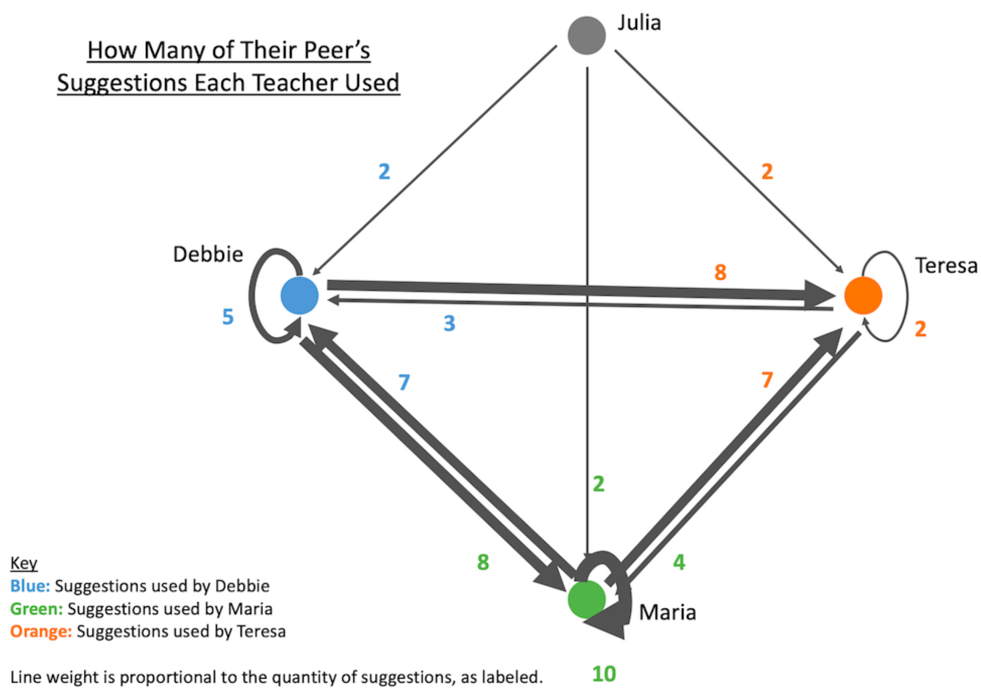
How Suggestions Were Implemented by Teachers

Suggestions Implemented in Class	Quantity Implemented	Percentage Implemented
Implemented by at least one teacher	23 (of 33)	70%
Implemented by at least two teachers	14 (of 33)	42%
Implemented by all three teachers	10 (of 33)	30%
Implemented by only one teacher	9 (of 33)	27%

Each teacher contributed to the creation of at least one suggestion episode. However, while the teachers all gave suggestions during the PD, they did not always implement their own suggestions. In fact, each teacher used many of their peers' suggestions. Figure 12 shows which teacher made the suggestions that were implemented by each of the teachers, with the direction of the arrow indicating how a suggestion moved from the teacher who made it to the teacher who used it (circular arrows represent a teacher taking her own suggestion). All teachers relied on suggestions made by the others, as well as suggestions made by the librarian. Furthermore, while some teachers had more of their suggestions get used than others, at least half of each participant's (including the librarian's) suggestions were implemented by someone in their classroom. Table 5 shows how many of each participant's suggestions were implemented by a teacher.

Figure 12

How Teachers Used Each Other's Suggestions



Note. Circular arrows represent a teacher taking her own suggestion.

Table 5

How Teachers Made and Used Suggestions

Participant	Suggestions Implemented Out of Made	Percentage Implemented
Maria	10 out of 16	62.50%
Teresa	4 out of 7	57.14%
Debbie	11 out of 15	73.33%
Julie (Librarian)	2 out of 4	50%

One example of a CS pedagogical suggestion that was implemented by multiple teachers was an episode that occurred in PD session 6, when Maria suggested that the teachers define “calling a procedure” before teaching students how to build procedures in Scratch. The following excerpt is the episode in which this suggestion takes place.

> **Researcher:** Those are all procedures that we have defined. Now, what they need to do is—because some of them have a “move forward three.” Well, I didn't build that procedure, because that's not in the game.
> **Debbie:** Right.
> **Researcher:** So, they need to build that procedure and they need to call it.
> **Maria:** So, I think for the lesson plan, I think you should just have the call part first before I say, “We're going to build a new procedure called ‘move back two squares.’”
> **Researcher:** Okay.
> **Maria:** Cuz then you, like, go back to that like you didn't really—
> **Researcher:** Yep.
> **Maria:** You know what I mean?
> **Teresa:** So, do the call first, and then…
> **Researcher:** Awesome.
> **Debbie:** That's a good idea, so put that before...
> **Researcher:** Yeah.

This altered the given lesson plan, which introduced the term “call” partway through the lesson. In the classrooms, Teresa and Maria both made this change and presented this definition before showing students how to build procedures. Debbie did not, though she did define “calling” a procedure near the beginning of her lesson. The following excerpt shows how Teresa implemented this suggestion by explaining what it means to call a procedure.

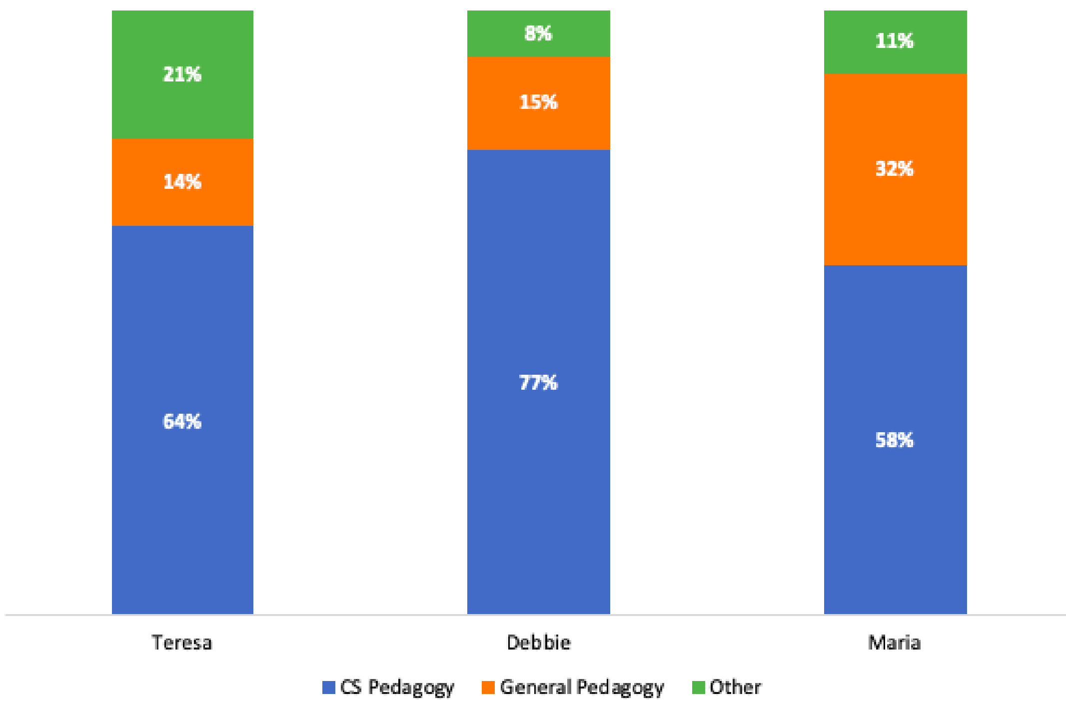
> **Teresa:** But what happens if we want to have something that's not on there? It means we're going to have to define a procedure. And then we can call it.
>
> So, I'm going to be talking about calling procedures today. And it's like, we want this program to do something. So, we're going to give a call on the phone. It's like, "Hey, I need you to do something for me. Can you do that?" And the program says, "Oh, sure. Just a second. I'll be right back." It goes and gets it, pulled it up, does the procedure and then our sprite can move again. So that's calling the procedure.
>
> Alright, so we're going to look at some things, and we're going to go to…we're going to try something new, it's called—I got them over there—move forward three. So, we want to define a new procedure.

Another suggestion that was implemented by multiple teachers was to have students physically stand up and move their bodies to demonstrate new game procedures during lesson 3. This was implemented by all three teachers in a slightly different way: Maria had one student stand and demonstrate the new procedure, Teresa had all the students do it simultaneously, and Debbie had students individually demonstrate procedures before having the whole class do it together. The implementation of this idea that was not preexisting in the lesson plan, and the varying ways in which teachers took ownership of the pedagogical choice shows agency in classroom practice as well as involvement in the PD by implementing it in the classroom. It also shows evidence of teacher learning by their ability to apply and adapt the content from the PD.

Nature of Implemented Suggestions. Figure 13 shows the percentage of topics from the suggestions each teacher implemented in their classroom. Suggestions regarding CS-specific pedagogy were the most commonly implemented type of suggestion for all three teachers, with more than 50% of each teacher's used suggestions focusing on that topic. The lowest percentage of CS Pedagogy suggestions used by any teacher was 58% by Maria, who claimed the most comfort with CS at the beginning of the project.

Figure 13

Percentages of the Topics of Suggestions used by Teachers

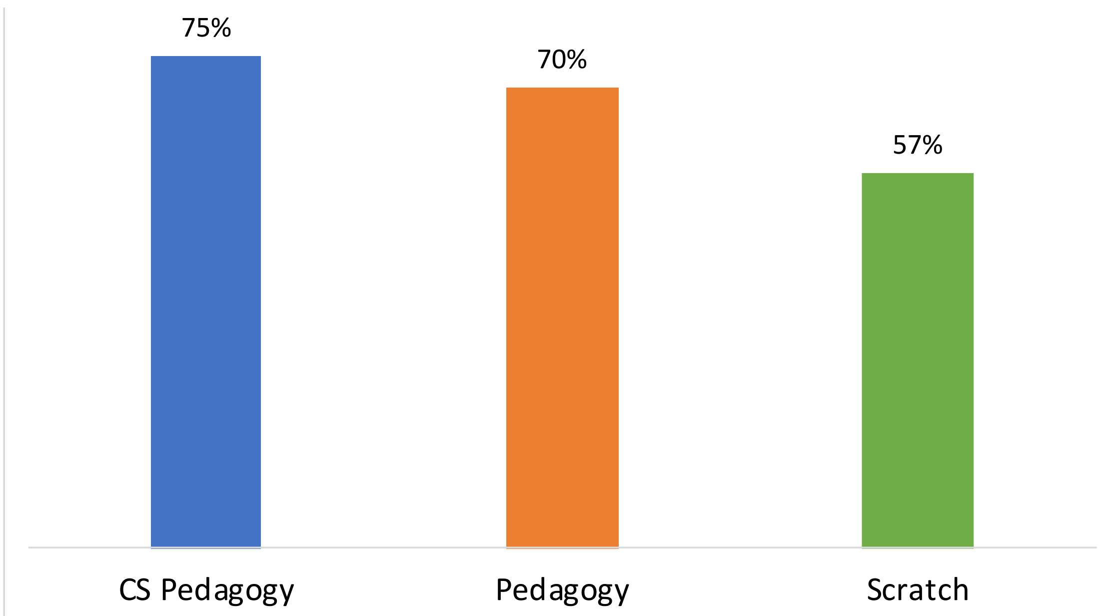


Another measure I used for analyzing suggestions implementation was examining what percentage of *available* suggestions were taken up out of each topic. Figure 14 shows what percentage of available suggestions was implemented by at least one teacher. CS Pedagogy suggestions showed the highest percentage of implementations. There were

16 possible suggestion episodes that were categorized as being about CS Pedagogy; 12 of those 16 CS Pedagogical suggestions, or 75%, were implemented by at least one teacher. In other words, teachers took advantage of 12 out of 16 possible CS pedagogy suggestions and chose to use them in their classroom. They also used a large percentage of the available suggestions about general pedagogy.

Figure 14

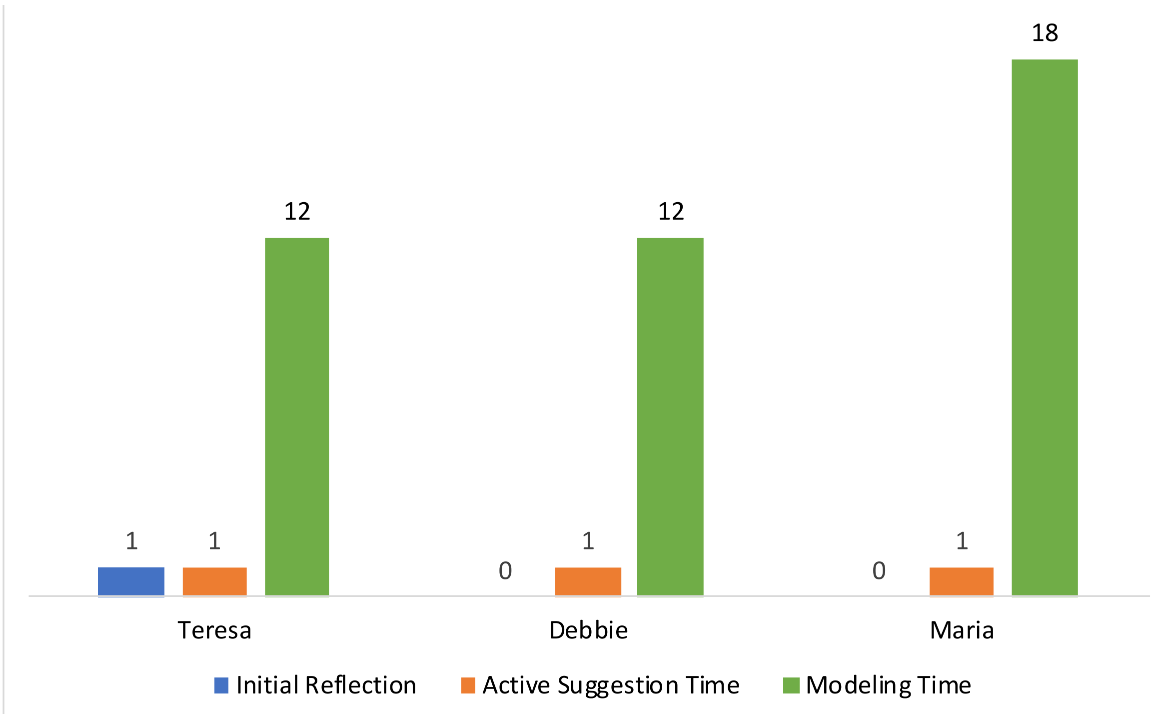
Percentage of Each Type of Suggestion Topic Implemented by a Teacher



Teachers also were more likely to use suggestions that occurred during the lesson modeling component of the PD. In fact, of the suggestions implemented, 91% came from the modeling portion of the PD or the conversation immediately following the modeling, where teachers were still discussing their plans for the upcoming lesson. This represents 72% of the usable suggestions made during this time period (21 out of 29). This is substantially more than those made during the time periods where researchers actively sought suggestions from teachers; in those times, only 33% of suggestions were implemented by at least one teacher. Refer to Figure 15 to see how many suggestions each teacher drew from the main components of the PD session—the initial reflection period, the modeling period, and the time when researchers were actively asking the teachers to make suggestions about the lessons.

Figure 15

Frequency of Suggestions Teachers Implemented from Each section of the PD



Classroom Implementation of Connection Episodes

In addition to implementing suggestions, teachers' connections episodes also influenced what they said and did in the classroom. While implementation of suggestions entails teachers taking the suggestions that were made during the PD and implementing them in the way that they teach the CS lessons, the implementation of connection episodes is more subtle, since teachers seldom explicitly made the exact same connections in the classroom as they did in the PD. Implementing connections in the classroom involved teachers using *some* of the same analogies that they made during PD connection episodes. However, more of the implementation of connection episodes took the form of teachers refining their use of computational terms as a result of their growing understanding of these terms.

One example of how teachers used their connection episodes to affect their teaching in the classroom was in their use of the CS term “procedures.” The following quotes show early ways that each teacher described and defined the term “procedures” in their classroom. Each of these quotes comes from a lesson prior to any formal discussion of the term “procedures” within PDs.

In Teresa’s quote, which took place during the first lesson, she describes a procedure as a set of cards used to move the game robot. For context, it should be noted that the game used in this curriculum, *Code:// On the Brink*, is played by moving a robot on a two-dimensional grid using cards that declare a type of move (such as “move forward”). The cards individually are analogous to the concept of computational procedures, but not as Teresa suggests:

> **Teresa:** Now, I have two cards on my blue. I have a "Move Forward" and an "X". Every time you have two movements, it makes one procedure. That means you cannot just do half a procedure; you always have to have two things in there.

In Maria’s excerpt below, which took place in the third lesson, she refers to classroom procedures to describe what a computational procedure is. Her focus in her description of the term is that a procedure is defined and that they tell the computer what to do. For context, “Elevate” is the name of the school’s writing curriculum.

> **Maria:** Alright? Yeah, so “move forward” means we're going to move 60 steps on the computer. And we're going to wait one second, right? This is called a

procedure. If I tell you the procedures for getting out Elevate stuff, what do we do first when we get out Elevate stuff? Abby?

Abby (student): Get out all our Elevate stuff.

Maria: Ok, we get out all our Elevate stuff, then what do we do?

Abby (student): Start on our language sheet?

Maria: Start our language sheet, and then we go into the lesson. Those are the procedures. But if I tell you get ready for Elevate, I don't have to say, "Get out your book, get out your other book, get out your other book, get your language sheet, get your pencil, write it down." I just say, "Get ready for Elevate," and you know right what to do. So that's kind of like this. It's the procedures. I say move forward. We already told the computer, right, what to do.

In Debbie's quote, which takes place in the third lesson as well, she defines the concept of procedural thinking as thinking step by step through the program. She makes a connection that a procedure is using step-by-step instructions, though appears to brush over this concept quickly.

Debbie: Ok, procedures using a step-by-step instruction. So today, it's really important that you guys are also thinking in step by step. Thinking, "Okay, when I'm trying to solve coding or one of the games or the boards, you need to think it step by step."

While these definitions are not necessarily wrong, they are incomplete or relatively simple. The teachers spend many PD sessions after these lessons discussing and clarifying the definition of a computational procedure, particularly in comparison to algorithms. One detailed connection episode takes place in the fifth PD session. The conversation begins as the teachers try to distinguish between algorithms and procedures in the Scratch code.

Debbie: What is the correct term for these again, these are our "procedures"?

Researcher 1: So, this is—

Teresa: Algorithms.

Researcher 1: Yeah.

Debbie: This would be an algorithm.

Maria: Yeah, tell me the differences between algorithms. I feel like I don't know what an algorithm was.

After the teachers discuss the definitions with a researcher, they begin to restate the ideas in their own words. They base their connections in this moment around the Scratch interface that they are familiar with.

Researcher 1: Algorithm is a fancy word for a whole bunch of steps.

Debbie: So, what's procedures?

Researcher 1: A procedure, we're—and this is a big focus today—is […] these red blocks that do a particular thing.

Debbie: Ohh. […] So, this is the algorithm to make them all work?

Researcher 1: Yeah.

Debbie: A procedure is once you're back at the top and it says, “do red card, move forward—"

Maria: And then you see in the main computer program, the red blocks—

Debbie: Ok.

Research 2: That's where we call that procedure.

The teachers then start to relate the content to other ideas that they are familiar with. In this part of the episode, they are making connections to the CS content through analogies to mathematics. They also continue to use the Scratch code to help them define the computational terms. The researchers use this time to provide additional analogies to help the teachers understand the terms.

Teresa: So, in school terms, addition—[…] carrying your ones—that's the algorithm. All the little steps that are in there, okay.

Researcher 1: Yep.

[…]

Maria: So, this is a procedure?

Researcher 1: Uh-huh. And that particular procedure has two steps. Right? But notice, the two steps are also red, which means they're also procedures, right? And so, we can go over to—

Debbie: So, would a procedure be like two plus two is four, every time you add that?

Researcher 1: That would be an algorithm […] because that's the steps. A procedure would be something like—let's go back to the cooking analogy.

Debbie: Okay.

[…]

Researcher 1: Yeah, a procedure would be the recipe. Cut your onions, sauté them, dah-da-dah. A procedure would be sauté onions. And if you wanted, you could unpack that and say, “That says go get the onion, slice it up into small bits, put it into the pan,” but you abstract away from those details to make a procedure called “sauté onions.”

Debbie: Gotcha. Okay. And the algorithm is all of it together.

Researcher 1: Yeah, the whole thing.

Teresa: It's all the instructions.

The conversation then continues in the vein of this new cooking analogy as a researcher explains how calling a procedure helps to reduce the amount of code that needs to be written by abstracting it in a procedure. Teachers use this to continue thinking about how procedures can define step-by-step instructions. The conversation then wraps back around to the math analogy and the Scratch interface as teachers confirm and finalize their thoughts on each of these connections.

Researcher 3: Another way to think about it is if you type up the instructions for sautéing onions, every time you do it, you want to have…you call the procedure. You don't have to write all that code.

Debbie: Gotcha. Okay, so that procedure lets me know that it's actually “get out the onion, chop it,” without me having to say it all. The algorithm is the whole entire thing of you doing it. Ok.

Maria: I guess that makes sense when you were thinking about long division or something. You do the algorithm. You do this, then you do this, then you do this, and when you get to this point, you do it again—

Debbie: That's the procedure, but then you do the algorithm—

Maria: That's the algorithm, because it's like, how many times does it go into the first number, then you minus.

Debbie: So, what could you consider the procedure in math?

Teresa: Long division.

Researcher 2: Yes.

Debbie: Oh, it's just the long division.

[…]

Researcher 2: All long division, and then you do—

Debbie: Under that, gotcha—

Teresa: All of the instructions.

Researcher 1: But just to go through this again. If you know…if you scroll up to the top, right, there's a procedure called […] blue cards. And then under it are two red blocks, and so Scratch uses color to say these are actually procedures too, and now you can scroll down and…let's go see them. There they are. Turn right.

Debbie: Oh, so procedures are all the red ones?

Researcher 2: Yes. […] once we define. So, there's two things we can do with procedures. We can define them, which tells them what they do, and then we can call them, which tells them to do those things.

Debbie: Okay.

This whole conversation lasted only a matter of minutes, but it was one of several that focused on these types of thoughts—notice how the researchers suggest that they "go back to the cooking analogy," which references a previous conversation based around this analogy.

These conversations influenced the way that teachers discussed procedures in the classroom. The following quote is taken from Teresa's classroom after the fifth PD, where the above conversation took place. Teresa provides only a short description of procedures in her classroom. However, her definition is far more accurate, and her explanation provides a more fundamental description of what a procedure does—providing step-by-step instructions.

Teresa: So, we've got all of our cards and these are the things that tell them what to do, right? So, we've got our algorithms, we've got our procedures. *Our procedures tell us what our steps are and what our movements are* [emphasis added].

Notice that Teresa also continues to reference the game as she provides her description of procedures, showing how she not only made sense of the term herself using the board game, but also how she benefited from the game as a teaching tool. This shows how the expansive framing model used in this curriculum benefitted teachers as both a learning and teaching tool, even if they were not aware of the model underlying it.

In Maria’s fifth lesson, she uses a similar strategy to what she did in the third lesson—focusing on an analogy to something the students would be familiar with from everyday classroom activities. However, in this case, instead of focusing on the analogy itself, her emphasis is on *how* the analogy is a procedure and how procedures are step-by-step instructions, as was clarified in their recent PD conversation.

> **Maria:** I just wanted to remind you these are called our procedures. So, if we have red cards, “do red cards,” if I'm a computer, you need to tell me what that means, right? So that's why we […] drag our cards right here. But if I'm a computer, do I know what turn right means? How do I tell myself what turn right means? Martin?
>
> **Martin (student):** You show on the code—the stuff that it knows.
>
> **Maria:** Okay, you have to tell it exactly, right? You have to give me the procedure for turning right. If I know the procedure for sharpening a pencil, I could tell Martin, "Martin, sharpen your pencil," and he would know what that meant, right? Because he's a smart fifth grader. But if Martin was a computer, and I said, "Go sharpen your pencil," he would kind of be stuck. He wouldn't be able to do it. I would have to say, "Stand up, turn around, walk five steps, grab your pencil, walk five more steps, put it in the pencil sharpener, grab the crank, turn it five times." I have to give him [something] specific.

While Debbie doesn’t redefine procedures in her fifth lesson, she does talk about them in the sixth. In this example, she describes specific procedures in the game and then defines them as procedures after discussing what value they provide in the program. This adds depth to her previous description of procedures as “step-by-step instructions,” which was already technically correct, if not rushed in the conversation wherein it was presented.

> **Debbie:** Alright, so for today what we're going to do is we're going to learn how to build new procedures. And then we're going to discuss why procedures are useful. […] If I want my sprite to know what a move forward is, how does that sprite know? Carson?
>
> **Carson (student):** Because you program it?
>
> **Debbie:** Okay, I programmed it. So, does that mean that I just use one of these red “move forward” cards?
>
> **Carson (student):** No, when you scroll down there's things that tell you.
>
> **Debbie:** Good, because this just says “move forward,” but I haven't defined it yet, right? So, if I go down below, it should have "move forward" defined for me just like Carson said, right there. It says, “move forward: move 60 steps and wait one second.” So, it knows when I say, “move forward,” I move 60 steps, which is one block, and then I wait one second. Okay, what if I wanted to do a “turn right”? How does it know, what do I do? Anyone else know? Bennett?
>
> **Bennett (student):** So, what you do for […] a turn right, you should move your mouse by 90 degrees and wait one second, then move forward.
>
> **Debbie:** Good, so he knows that it's been programmed in there, and it's called a procedure. So, we need to define any procedures that aren't already in the cards.

This series of quotes provides some evidence of how teachers benefitted from and implemented the connections episodes in their classroom. Similar events occurred with other terms and with other analogies and descriptions shared during the PD. However, there were also many connections that did not get directly implemented in the classroom. While this may seem like failure of the PD to translate to the classroom, these connections could instead be considered as moments where teachers went beyond the needs of the classroom to focus on their individual learning at a deeper level. Because these conversations provided opportunities for teachers to learn and grow collaboratively in their understanding of CS as learners, it provided a valuable benefit regardless of how directly it appeared in the classroom.

RQ3: How the Expansively-Framed Curriculum Impacted Teacher Learning

The third research question I address in this thesis is: *How does an expansively-framed curriculum support* teacher *learning?*

This question can be addressed through the coding scheme that was developed; there are 242 teacher utterances throughout the PD sessions coded as having the board game as a topic. By looking at how often teachers make comments that are coded as being about the game, I can see how often teachers talk about the frame in general during the PD. Furthermore, there are 44 utterances where teachers talk about the game in conjunction with CS content in general. Examples of these utterances include the following:

> **Debbie:** Well, if it's like what you're saying with the games, are you meaning, like, when I land here, I need to turn left—is that what you're wanting as an event? (PD 2)

> **Maria:** So, when we're defining a movement, we're actually defining a procedure, the steps. And once we've defined the procedure, then we can put it in as a card in our program. Okay, on your games, well, you've made some of your own cards that aren't right here. So, if you made your own card, we're going to have to build the procedures for that. Let's start with the— (PD 6)

> **Maria:** So, if it doesn't have abstraction, we have to say move forward, or we have to say, "Move 60. Wait. Move 60. Wait." Then the cards for the left. (PD 7)

However, beyond simply quantifying how often teachers speak about the game, I can look at how often teachers' connection episodes focus on the game and how it connects to the content. Recall that of the 24 connection episodes made, 12 of them connected the CS content to the board game and two of them connected the board game to Scratch. Thus, more than half of the connection episodes made focused on the board game, showing how often teachers relied on that as a source for their own learning.

An example of an episode where teachers connect the board game to CS content occurs in the fifth PD, where they discuss what conditionals and abstraction mean in terms of the game rules to touch on colored squares.

> **Debbie:** Okay. The condition is met…it does not… Okay, so I don't understand this part. Walk through the—check and check each condition to see if it is met. If the condition is met, it does each step, that is—oh, okay. So, does that mean that if I have here, if it says, "if touching color blue," then we'll go that. So, if you actually land on a blue, then that condition is met and then it will do the blue cards?
>
> **Researcher:** Yes. What happens is—
>
> **Teresa:** So, you're checking each of your things in there.
>
> **Maria:** What if we used a repeating block to define this?
>
> **Researcher 2:** Yes. Try ten all together. And then you could scroll up. It's up to you to show them what do blue cards is.
>
> **Debbie:** Okay? Because you can't see those there, right?
>
> **Researcher 2:** And then look at that. There's two procedures under them.
>
> **Debbie:** So that's [sic] means that it will also do those two because I touched that one, right? That condition?
>
> […]
>
> **Maria:** Then you could say we've abstracted all the way to this.
>
> **Researcher 2:** Yes.
>
> **Maria:** We hit start and it does the whole thing.
>
> **Researcher 2:** Yeah, so…so, there's three levels of abstraction here. So, it's hard. This is hard conceptually.

Additional examples of connections episodes that focus on the game have already been included as examples in this thesis, including the second example described in Chapter IV in the section defining connection episodes, where teachers make the connection between events and the cards in the board game (page 40), as well as the example in the section describing how connection episodes were implemented, where teachers defined procedures and differentiated between procedures and abstractions by talking about the Scratch instantiation of the game (page 53-54).

As can be seen, therefore, teachers often used the game to frame their understanding of the CS content that they were learning in the PD. During the final PD, teachers discussed how they felt about the project with the researchers for a few minutes. During this time, the teachers were asked what they would suggest to future teachers implementing this unit. They answered by saying how helpful the board game had been to them and their learning, and recommending that future teachers also rely on it.

> **Researcher:** Ok. What suggestions do you have for future teachers who teach or participate in this project?
>
> **Maria:** So, like, maybe if they, like, had…well, we did have the board to play and stuff.
>
> **Julie:** Yeah, I just felt…
>
> **Maria:** But, like, show. I did it, like, once, yeah, I didn't—
>
> **Teresa:** I didn't, yeah. I didn't use the game as much. I worked more on the computer than I did on the game. [Note—the computer refers to the Scratch iteration of the game.]
>
> **Julie:** Although I was surprised at how much I learned from the game. Like—
>
> **Debbie:** Yes. I think the game is…
>
> **Julie:** It made sense better in my head because of the game.
>
> **Researcher:** Yeah?
>
> **Teresa:** The first couple of weeks.
>
> **Julie:** Yeah.
>
> **Teresa:** To manipulate the board and do that.
>
> **Researcher:** So that was helpful?
>
> **Debbie:** Definitely.
>
> **Julie:** Uh huh.

This excerpt confirms what can be seen from these connection episodes—that the teachers' learning centered around the game, much as the design of the curriculum was intended to support students. This suggests the value of an expansive framing approach for teacher learning, in addition to the expected effect on student learning.

Chapter V: Conclusions

In review, this study addressed the following research questions:

1. How do teachers discuss new CS content and pedagogy in a collaborative professional development setting?
2. What types of teacher discourse about new curricula that occur during collaborative PD impact teachers' subsequent classroom implementations?
3. How does an expansively-framed curriculum support *teacher l*earning?

I found that teacher conversation during the collaborative PD could be described in terms of sense-making episodes, or conversations comprised of teacher utterances that showed how they made sense of the PD and curriculum. The three types of episodes that described teacher participation during the PD were defined as: suggestion episodes, reflection episodes, and connection episodes.

Suggestion episodes, or conversations around suggestions that teachers made for the CS curriculum, were centered around pedagogy 77% of the time, with roughly half of these pedagogical suggestions focusing on CS pedagogy specifically. This highlights that the teachers were engaged with the process of understanding and adapting pedagogy for the CS curriculum, despite the newness of the content for the teachers. Furthermore, about 70% of these episodes were triggered by peer question-and-answer periods and peer reflections. This is indicative of the value of teacher conversation and peer collaboration in developing the teacher agency to suggest making changes to the lessons. This claim is further supported by the fact that teacher modeling of the lesson triggered an additional 10 suggestions, providing evidence of the participation of teachers in an active and collaborative manner.

Reflection episodes, or conversations centered around reflections the teachers made, centered on student understanding in 27% of episodes, and students in general in 47% of episodes. These types of sense-making episodes were often triggered by researcher comments or questions due to the nature of the PDs. Time was explicitly built into the PDs to allow teachers to reflect; when prompted to do so, teachers would naturally take turns reflecting on their implementation experiences and providing feedback to each other or corroborating each other's experiences. This shows the value of including reflection time in a structured collaborative PD, as well as the value of taking the time during PD for teachers to discuss things with each other at their own pace and in their own way.

Connection episodes were defined as conversations centered around statements teachers made that showed how they understood or related to a topic. These discussions were often centered around how the board game connected to the CS content, showing how the teachers were supported by the expansive framing instructional approach to make sense of the content. Furthermore, half of the connection episodes were triggered by modeling—the most common type of triggering event. This again shows how modeling of lessons encouraged teachers to engage in sense-making activities during the times when they led the PD and had active roles.

I next looked at how teachers' sense-making in the PD impacted their classroom implementations. I found that a majority (23 out of 33) of usable suggestion episodes were implemented by at least one teacher in the classroom. Thus, the teachers used each other's suggestions, trading ideas regularly. This shows how the collaborative nature of teacher learning directly benefitted teachers and, subsequently, their students. In addition, more than half of the suggestions that each teacher implemented were about CS pedagogy, showing how the PD benefitted the teachers directly in the area that they were least confident in (the content itself). Furthermore, the modeling portion of the PD influenced teachers' classroom actions, as 91% of implemented suggestions came from that portion of the PD. This supports findings of Goode et al. (2014) where modeling is found to be a useful tool in teachers' exploration of and development of CS content knowledge.

Teachers also implemented connection episodes in their classroom as they used the connections that they made in the PD discussions to shape the way that they taught CS concepts. While not all of the connections that teachers made were implemented in the classroom, the value of these conversations still impacted the teachers. The connections that did not reach the classroom were examples of teachers expanding their understanding of the content area to enhance their own learning and to effectively teach the content in the classroom. If one considers the teacher learning models that were presented in the literature review, one must suppose that anything teachers learned in the PD through these sense-making conversations would at least indirectly impact what they do in the classroom. Thus, while students in these teachers' classrooms were directly impacted by the sense-making episodes that occurred in the PD, they were also indirectly impacted in positive ways by the teachers' learning.

Finally, the third research question focused on how teacher learning was supported by the expansive framing instructional approach. As can be seen by the fact that half of teachers' connection episodes connected CS content to the board game, teachers often relied on the expansively-framed curriculum to understand the content.

Teachers often talked about the board game throughout the seven PD sessions and became familiar with talking about CS concepts in terms of how they were represented by game rules and blocks in the Scratch environment. However, even more importantly, teachers stated in the final PD that they believed that the board game helped them learn, showing that they could clearly see the value for themselves (refer to page 61). By looking at what teachers talk about, it seems clear that the board game helped them learn; by listening to the teachers, that assumption can be confirmed.

Knowing that teachers benefitted from the expansively-framed curriculum is a valuable addition to the previous understanding that expansively-framed instruction could benefit students. By recognizing that teachers are learning how to teach CS at the same time that they themselves are learning CS content, it can be seen that they too are learners and would, therefore, reasonably benefit from the same tools that help students—in this case, a frame which relates the content to something with which they are already deeply familiar and interested.

Returning to the Literature

As was stated in Chapter II, existing research on collaborative PD has not deeply addressed the process of teacher learning over the course of collaborative PD. This study presents a description of teacher learning in collaborative PD as shown through the collaborative discussions that teachers engaged in. These discussions, characterized as three types of sense-making episodes, have been shown to impact the classroom implementation. Through evidence of changes in teachers' practices using connections episodes, this study shows evidence of teachers' learning. Through evidence of teacher engagement in the material as seen in suggestion episodes and their implementation, this study shows changes in teachers' content knowledge. This study thus shows how teachers discursively engage during collaborative PD, and how this engagement led to teacher learning. This study also confirms the findings of Horn (2010) regarding the value of peer conversation in teacher learning, extending it to a formal setting such as collaborative PD.

Furthermore, this study shows the value of the PD model developed through the combination of several teacher learning frameworks (e.g., Firestone et al., 2020; Gess-Newsome et al., 2019). Teachers increasingly engaged in the PD through discussion, as shown in Figure 5, which led to changes in teacher knowledge and classroom practice. In addition, the reflections that occurred in the PD provided evidence of how these classroom changes further influenced the PD, corroborated by how many of these reflections became the triggering events for suggestion and connection episodes in turn.

Finally, this research helps to build on existing expansive framing research by showing how a curricular approach which is known to be effective for students can also benefit teacher learning. This has been seen through the way that teachers made many connections centered on the board game, as well as through statements teachers made in which they directly reflect on and confirm the value of the board game as a tool for their own CS learning.

Limitations and Future Research

The research conducted in this thesis is based around a single implementation of a CS curriculum in a single, rural school by three teachers and one school librarian. Thus, there is a limited sample from which to draw analyses of these sense-making ideas. Furthermore, the nature of the research being conducted in the implementation was initially focused on student learning, rather than the actions or learning of the teachers. Thus, the only data that was gathered from PD sessions was audio recordings. While this provides us with a decent picture of what happened during that time, video data in future iterations of research would allow a better picture of the interactions occurring during PD, allowing for a more detailed interaction analysis to occur, as recommended by Walkoe and Luna (2020).

While some data was gathered regarding how sense-making episodes were triggered, less data was gathered about what happened after sense-making utterances were made within these episodes. A more detailed analysis of these trailing events, as they could be called, could allow researchers to gain a more detailed understanding of how teachers make decisions about what suggestions and ideas they will implement in their classrooms from those made available during the PD.

Conclusion

In conducting this thesis study, I set out to understand how teachers discuss CS content and pedagogy in a collaborative PD setting. I have found that teacher conversation in such a setting can be rich, and representative of sense-making and agency; teachers make connections between content, suggest changes that meet their needs and represent their understanding of the material, and reflect on the effects of these changes together. These conversations directly impact their classroom implementation and have the potential to have lasting impact on teacher CS knowledge. Furthermore, I found that when teachers are given an expansively-framed curriculum, they benefit from the connections it encourages them to make just as previous research would expect for students. This shows the value of expansive framing in impacting teacher learning, even if teachers do not recognize it as such.

With this greater understanding of teacher sense-making in collaborative PD, I hope to be able to conduct future research that will expand on how teachers learn, especially in content areas as unfamiliar to them as CS. By doing this, I hope to be able to benefit teachers in their pursuits of mastery of such an important curricular area, and to benefit researchers and content designers in understanding how teacher learning can be generalized to enhance other areas of study as well.

References

Ball, D. L., & Cohen, D. K. (1999). *Developing practice, developing practitioners: Toward a practice-based theory of professional education.* In L. Darling-Hammond, & G. Skyes (Eds.), Teaching as a learning profession: Handbook of policy and practice. San Francisco: Jossey-Bass.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, *13*(1), 20-29.

Borko, H. (2004). Professional Development and Teacher Learning: Mapping the Terrain. *Educational Researcher*, *33*(8), 3–15. https://doi.org/10.3102/0013189X033008003

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Engle, R. A., Lam, D. P., Meyer, X. S., & Nix, S. E. (2012a). How Does Expansive Framing Promote Transfer? Several Proposed Explanations and a Research Agenda for Investigating Them. *Educational Psychologist*, *47*(3), 215–231. https://doi.org/10.1080/00461520.2012.695678

Engle, R., Nguyen, P., & Mendelson, A. (2011). The influence of framing on transfer: initial evidence from a tutoring experiment. *Instructional Science*, 39(5), 603–628. https://doi-org.dist.lib.usu.edu/10.1007/s11251-010-9145-2

Firestone, A. R., Cruz, R. A., & Rodl, J. E. (2020). Teacher Study Groups: An Integrative Literature Synthesis. *Review of Educational Research*, 90(5), 675-709.

Fishman, B. J., Marx, R. W., Best, S., & Tal, R. T. (2003). Linking teacher and student learning to improve professional development in systemic reform. *Teaching and Teacher Education*, 19(6), 643-658. doi:10.1016/S0742-051X(03)00059-3

Gess-Newsome, J., Taylor, J. A., Carlson, J., Gardner, A. L., Wilson, C. D., & Stuhlsatz, M. A. M. (2019). Teacher pedagogical content knowledge, practice, and student achievement. *International Journal of Science Education*, *41*(7), 944–963. https://doi.org/10.1080/09500693.2016.1265158

Goode, J., Margolis, J., & Chapman, G. (2014). Curriculum is not enough: The educational theory and research foundation of the exploring computer science professional development model. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education - SIGCSE '14*, 493–498. https://doi.org/10.1145/2538862.2538948

Grover, S., Pea, R. and Cooper, S. (2014). Expansive Framing and Preparation for Future Learning in Middle-School Computer Science. In Proceedings of the 11th International Conference of the Learning Sciences (2014), Boulder, CO.

Hickey, D. T., Chartrand, G. T., & Andrews, C. D. (2020). Expansive framing as pragmatic theory for online and hybrid instructional design. *Educational Technology Research and Development*, *68*(2), 751 782. https://doi.org/10.1007/s11423-020-09759-4

Guskey, T. R., & Huberman, M. (1995). *Professional development in education: New paradigms and practices.* Teachers College Press, 1234 Amsterdam Avenue, New York, NY 10027 (paperback: ISBN-0-8077-3425-X; clothbound: ISBN-0-8077-3426-8).

Horn, I. S. (2010). *Teaching Replays, Teaching Rehearsals, and Re-Visions of Practice: Learning From Colleagues in a Mathematics Teacher Community*. 36.

Kartal, E. E., Dogan, N., Irez, S., Cakmakci, G., & Yalaki, Y. (n.d.). *A five-level design for evaluating professional development programs: Teaching and learning about nature of science*. 26.

Lam, D. P., Mendelson, A., Meyer, X. S., & Goldwasser, L. (2014). *Learner Alignment with Expansive Framing as a Driver of Transfer*. 8.

Lee, V. R., Poole, F., Clarke-Midura, J., Recker, M., & Rasmussen, M. (2020). Introducing Coding through Tabletop Board Games and Their Digital Instantiations across Elementary Classrooms and School Libraries. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 787–793. https://doi.org/10.1145/3328778.3366917

Lee, V. R., & Vincent, H. (2019). An Expansively-framed Unplugged Weaving Sequence Intended to Bear Computational Fruit of the Loom. *Proceedings of FabLearn 2019*, 124 127. https://doi.org/10.1145/3311890.3311907

Lefstein, A., Vedder-Weiss, D., & Segal, A. (2020). Relocating Research on Teacher Learning: Toward Pedagogically Productive Talk. *Educational Researcher*, *49*(5), 360–368. https://doi.org/10.3102/0013189X20922998

Peel, A., Dabholkar, S., Anton, G., Wu, S., Wilensky, U., & Horn, M. (2020). *A Case Study of Teacher Professional Growth Through Co-design and Implementation of Computationally Enriched Biology Units*. 8.

Putnam, R. T., & Borko, H. (2000). What do new views of knowledge and thinking have to say about research on teacher learning?. *Educational researcher*, *29*(1), 4-15.

Rich, P. J., Jones, B., Belikov, O., Yoshikawa, E., & Perkins, M. (2017). Computing and Engineering in Elementary School: The Effect of Year-long Training on Elementary Teacher Self-efficacy and Beliefs About Teaching Computing and Engineering. *International Journal of Computer Science Education in Schools*, *1*(1), 1. https://doi.org/10.21585/ijcses.v1i1.6

Richardson, V. (1996). The role of attitudes and beliefs in learning to teach. In J. Sikula, T. Buttery, & E. Guyton (Eds.), *Handbook of research on teacher education* (pp. 102–119). New York: Simon & Schuster Macmillan.

Saldana, J. (2015). *The Coding Manual for Qualitative Researchers*. Newcastle upon Tyne: Sage.

Steeg, S. M., & Lambson, D. (2015). Collaborative Professional Development: One School’s Story. *The Reading Teacher*, *68*(6), 473–478. https://doi.org/10.1002/trtr.1338

van Oostveen, Roland. (2017). *Purposeful action research: Reconsidering science and technology teacher professional development.* The College Quarterly. 20.

Voogt, J., Laferrière, T., Breuleux, A., Itow, R., Hickey, D. T., & McKenney, S. (2015). Collaborative design as a form of professional development. *Instructional Science*, 43, 259–282.

Walkoe, J. D. K., & Luna, M. J. (2020). What We are Missing in Studies of Teacher Learning: A Call for Microgenetic, Interactional Analyses to Examine Teacher Learning Processes. *Journal of the Learning Sciences*, *29*(2), 285–307. https://doi.org/10.1080/10508406.2019.1681998

Appendix

Coding Scheme: Stage 1 of PD Analysis

The following coding scheme was used to code for the discursive purpose of statements made by participating educators at an utterance level.

Table 6

Coding Scheme for Stage 1 of PD Analysis

Code	Definition	Example
Teacher Question	All teacher-asked questions; also includes statements expressing a lack of understanding that are treated like questions by participants, or statements that are described as questions by the speaker, regardless of how they are phrased grammatically.	"Question. One of my students is gluten free." "I'm just trying to figure out if I'm supposed to do two of them in a row or, I pick two out of there that it's going to make the sprite move where I want to."
Teacher Answer	A teacher's response to another teacher's question or a researcher's question; Must be within 4 lines of a question to be considered an answer (chosen because there are five contributors—four teachers/instructors and one researcher.)	"You have to go through all the things to get there first."
Reflection	Comments teachers make about what they did in their lessons in the CS curriculum, what students did, and what they thought about their teaching and actions. These statements specifically reference what occurred; statements about what ought to have been done are not reflections but rather suggestions.	"I put the board on the tray and put the cards right above, so they could see this is where they would—" "They must have got it because when they came in here, they already knew. I didn't even have to do any explanation." "I thought the lesson went well enough, I just thought it was too much."
Making Connections	A declarative sense-making statement that expresses how the participant understands or relates to the topic (particularly CS topic). Examples include a participant using one of the following for sense-making: personal experience past classroom experiences the expansive framing previous conversation in PDs Other CS topics	"Oh, it's a mines, it's like a Minecraft type thing--" "Like what we do in computers with Mrs. Hansen?" "So basically, you're saying if statements are events." "Oh, we could define get the peanut butter. Get the peanut butter means open it, get your knife, and spread it—" "But if I sit down to play the game, I need to be able to be like what's gonna have, I've got to put the cards in, that's what you're saying."
Modeling	Teachers modeling how they would teach in their classrooms by presenting to an imagined student audience; typically	"Okay kids, so if you want it to restart, you're going to need your mouse to know that it's got to go to this, or should I say sprite, I'm sorry, to

	occurs in the teacher as-learner section of the PD	negative 211 and 24. So I'm just going to type that in."
Suggestion	A comment made to suggest a pedagogical change to the lessons, administrative choice, or content revision to the lessons, PDs, etc.	"Do you know what you could do? Use a remote and zoom in." "So, you could say, that's because it's conditional because it's only going to work if this happens. You're not going to get grounded unless you hit your sister, if, but if your sister, then you're going to get grounded."
Technical Demonstration	A teacher demonstrating to others how to use a technological tool. This includes how to use the Scratch interface, but not how to program in Scratch.	"So, when I'm here, I just click here, right click—"
Expectation	Teachers expressing expectations about what will happen in the future lessons and how they or students will act, think, or behave in future lessons.	"They're going to get real confused, yeah." "And it will go much faster than this." "Yeah, cuz I thought, was like if we got on and did this once in a while, I could get better at it and feel more confident if I ever went back to teach it."

Coding Scheme: Stage 2 of PD Analysis

The following coding scheme was used to code for the topic or thematic purpose of statements made by participating educators at an utterance level.

Table 7

Coding Scheme for Stage 2 of PD Analysis

Code	Definition	Example
General Pedagogy	Related to how lessons were taught, teaching techniques, & theories; Content about how to use non-Scratch technology (Modeling of teaching techniques is covered under syntactical codes.)	"And we could probably put this under the doc cam?" "Yeah. So maybe we could do more of that "we do" like how you're doing with us like, Okay, what do you think now watch, I can test it. Oh, I made a mistake. I know I need to move that now."
CS Pedagogy	Related to how CS content specifically is taught or should be taught; subcategory of Pedagogy	"So, something we did with first and second grade last year, is we had them program the teacher. It helped really good. So, they had to get me from A to B. And so, they had to say turn. And I physically did it."
CS Content	Related to CS and CT concepts	"So basically, you're saying if statements are events." "Yeah, maybe I was actually like, so procedure was like all the steps, right? And abstraction is like the

		act of doing it. Maybe I wasn't, I don't know if I taught that completely correctly.”
Analogies	Teachers make analogies to describe CS content; a subcategory of CS content	“Like here's a condition. What's the condition outside it's raining, that's a condition so you could—”
Students	Teachers talk about how their students felt or acted or how they expect their students will feel or act	“So, I asked them, one day, those of you who said you didn't like this, why didn't you like it? And I'm just curious. And a couple of them were like, I didn't like having to work with a partner.” “They're going to be yelling no it's not going to work!”
Real-life Applications	Comments that are about how the content from this project can be applied to students' futures	“Okay, so like on a computer, what, what things will they be doing as an adult if they wanted this job? Like, I guess that's what—”
Past Experiences	moments where teachers discuss their or students' previous experiences with CS; typically used as orientating declarations, but does not have to be	“Like when the STEM bus came, we had to get through to that.” “I know they used scratch last year with Mrs. April, and kids just seem to wrap their heads around this type of stuff easier than I think adults do.”
Board Game	Moments when the teachers are playing the board game or talking about the board game	“Does the finish line count as white?” “Is it or? Is it, is it move forward or nothing?”
Scratch	Moments when the teachers are discussing the use of Scratch and/or actively using Scratch	“So, you clicked on my stuff, so you might have missed that, show the my stuff one.” “So, in our program if we have if-then statements checking to see if the sprite is on blue, red, or yellow, but not for green squares. To make an if-then statement for green squares, we need to go to the control tab, so over there and go and drag over the "If-Then" blocks.”
Self-Efficacy/Affect	Expresses teachers’ self-efficacy and affect, as well as times when teachers express generalized beliefs about CS, computer scientists, or other related topics	“I felt bad how I like ended up doing it, I was like I hope this is okay. But I didn't want to ask you like as we were doing the lesson so I'm like I'm just gonna go with it and he can tell me later if it didn't work.” “Okay, this is like a totally stereotypical thing, but I feel like my husband always makes fun of this and so it's okay, cuz he's, but like, he says people that are into coding aren't very social sometimes. So those people that are really good at this game might not want to have a partner.” “These look hard...! I'm not going to be able to do this.”

Table 8

Coding Book Defining the Types of Sense-Making Episodes

Code	Definition	Example
Suggestion Episode	An episode of teacher discussion that demonstrates teacher sense-making by illustrating how teachers voice recommended changes to the curriculum, as well as voice recommended pedagogical techniques and other forms of advice for their colleagues.	See Section Titled "Suggestion Episode Examples" (Page 35)
Reflection Episode	An episode of teacher discussion that demonstrates teacher sense-making through teachers' comments about the previous lessons' implementation and effectiveness; may also centered around the students as they participated in the lesson, as well as on the teacher themselves.	See Section Titled "Reflection Episode Examples" (Page 39)
Connection Episode	An episode of teacher discussion that demonstrates teacher sense-making by illustrating how they make learning connections between the CS content and their prior experiences and expertise.	See Section Titled "Connection Episode Examples" (Page 42)

Table 9

Coding Book Defining the Triggering Events for Suggestion Episodes

Code	Definition	Example
Question & Answer	Triggered when a teacher asks a question and one of their peers or researchers answers the question, which leads the group to make related suggestions. The suggestion may be the answer to the question or a response to the answer given.	M: How did you decide to partner that? Was it just random? Researcher: Jenny partnered them up the first day and then we just kept it with that. But I'm guessing it was random on her part because she said she wasn't really sure who, hindsight being 2020, now we know you guys have been reading partners. That probably would have been the ideal partners. M: Or like, yeah, maybe if we did, I don't know what like maybe our math partners, like precision partnering. So, we can kind of, not that that will go perfectly with this. But so, you don't get two lows or two highs you even just like, you kind of just get this half of the list— Researcher: You have people helping each other out.

		M: So that they're precisely partnered. T: Yeah. Rather than just—
Reflection	Triggered when a teacher reflects (makes an utterance that is coded as "reflection"). This reflection leads to related suggestion utterances being made.	D: So, so I found that with my class too, once your sprite moves one of your movements, if even, if your mouse is just a little bit off, the minute it touches a side, it stops. M: So, they're gonna have to get it really good on their very first one? T: Right. D: Can you make the mouse, was there a way that we can make— M: smaller? D: —The sprite smaller? Researcher: Yeah. D: So, then it doesn't happen as much. Maybe that would help the kids. M: Yeah. D: If they knew... because his tail gets in the way when he does a turn like on that last time. T: Yeah. M: Oh, there you go. T: Oh, perfect. D: Yeah, see, so now it would be less likely to hit. Okay.
Modeling	Triggered when a teacher models a lesson plan. The modeling leads to a related suggestion being made. The suggestion can be made through the modeling itself (demonstrated as modeling) or it can be made as an aside during the modeling activity.	D: Okay, and then I need a wait one second— M: And you know, I am going to put one wait at the end, just so that I can see it pause before it goes to the next.
Researcher Initiated	Triggered by a researcher asking a question or prompting the teachers to make suggestions. This is actively done—the researcher is explicitly looking for suggestions to be made.	Researcher: So, just, so thinking about the classroom implementation and this idea, you know, you could do a whole group, individual group, peer group, what do you think would work? D: I feel like we're doing like what we would do with like math or something like you, we, they need more instruction. Is that kind of what you guys are getting at? You feel like they need more like, practice? Researcher: No, no, we want, we're asking you. M: I mean, I think it'll be easier to do whole group because there's only one of me. And even if there's groups that are not, like, they're feeling pretty good about it, they're not going to like do the whole debugging, like, systematic way of figuring out, so I'm gonna have to do whole group or else— T: Right. I think modeling of all of those— D: But I like how you guys are wanting us to help them think through the process, because are they not doing that once they're doing it? Do you know or?

		Researcher: No, I think we just, I think they just need a little more time. D: Okay.
No Noticeable Trigger	The episode begins in such a way that nothing prior to the episode is related to the suggestion being made; therefore, nothing triggers it. My occur when a teacher interrupts an existing conversation with a new and unrelated suggestion or when a suggestion takes place first in the audio file and therefore no trigger can be identified due to data limitations.	M: Oh, I thought of something. When you said the board games thing, I had been thinking previously, before we started that we might spend like more than just like the one day on the board games. And I think that in terms of like getting them excited about it before you jumped to the computers, maybe like in the future, whatever. Like maybe spending a couple of weeks on just the board games, like and okay, let's play this game like can you do this challenging level? Like, can you do it or like race the teacher like who's smarter try to figure it out? I don't know, like a few times just doing that. So that's like fun to them that now we can do it, all of us can do it with the computers like. Researcher: I think that's a good point, it's available for check out in the library. M: Yeah.

Table 10

Coding Book Defining the Content Types for Suggestion Episodes

Code	Definition	Example
Scratch	The episode focuses on suggestions regarding how to use Scratch in the curriculum.	D: So, so I found that with my class too, once your sprite moves one of your movements, if even, if your mouse is just a little bit off, the minute it touches a side, it stops. M: So, they're gonna have to get it really good on their very first one? T: Right. D: Can you make the mouse, was there a way that we can make— M: smaller? D: —The sprite smaller? Researcher: Yeah. D: So, then it doesn't happen as much. Maybe that would help the kids. M: Yeah. D: If they knew... because his tail gets in the way when he does a turn like on that last time. T: Yeah. M: Oh, there you go. T: Oh, perfect. D: Yeah, see, so now it would be less likely to hit. Okay.
CS Pedagogy	The episode focuses on pedagogical suggestions,	T: Okay, so just conditionals. Researcher: So, it starts off with number two. Yeah, asking them— T: If on red, do red, then do red, if on blue, the do blues—

	specifically regarding the teaching of CS content.	M: So, you could say, that's because it's conditional because it's only going to work if this happens. You're not going to get grounded unless you hit your sister, if, but if your sister, then you're going to get grounded. T: Right. Researcher: Yep.
General Pedagogy	The episode focuses on pedagogical suggestions that are not specific to CS content, such as general practices.	Researcher: Which was my next, so you didn't actually have a driver. How did you guys pick your drivers? D: I picked my, so especially after, so at first we were like, okay, maybe we could reward someone. And it almost, once she kind of told me how it went with Griffin, I was like, no, I'm going to pick someone that I know is good enough and that will do it and I won't have any issues with. T: Well, he would have been, because he's done it before for me and when I give him a task, he's usually on so why he was so completely off that Monday... D: See and I, and, and I, and I guess that's what I was like, you know what, I'm going to pick someone that I know for sure it wouldn't happen no matter what, so. So, if you're going to have teachers do it in the future, I would say hey, just because of this, I would make sure it's one of your top students that maybe understands it plus, you know you're not going to need to— Researcher: Well and I think that like, like you walking it through with him beforehand probably definitely helped as well. D: And it didn't take long. I just quickly was like, Hey, here's kind of the things and once he knew he felt comfortable, he's like, got it, it was awesome.
Affect	The episode focuses on suggestions regarding how teachers can improve theirs or their students' experiences and affect in the project.	M: Well, I was getting observed too, right before, Trudy just came to observe me. Like she didn't tell me she was coming, she just came and did it. But then, like I was, I was fine, like I'm pretty used to people observing me so it wasn't that big of a deal, but then I was like, oh now you're going to film me. T: Yeah, you freaked me out with the with the video camera. J: You just have to pretend like he's not there— D: That's what I did, I just initially pretended like he wasn't there and didn't even look at him and then I was like, Okay, I can do that. M: Yeah, I just don't look at them at all.

Table 11

Coding Book Defining the Triggering Events for Reflection Episodes

Code	Definition	Example
Peer Initiated	A peer (teacher) makes a statement or asks a question that prompts another teacher	T: Right? And I think definitely, because I'm trying to think alright, if you just throw in colors randomly, is there even a solution and do you have to kind of pick your routes first before you put colors in?

	to make related reflection utterances.	J: So that's how they did it in the library. They had a starting point, and a starting direction, and then they would do a color and figure out what, and I mean so one step at a time so that it wasn't, it wasn't random, those kids— […]
Researcher Initiated	A researcher makes a statement or asks a question that prompts a teacher to make a related reflection utterance.	Researcher: So, what so, so, what can we have changed about this week? Maybe more time, but maybe not? M: Well, I mean, I don't know if I'd really want to take more time. I just had to go fast. D: I feel like mine, I feel like I had plenty of time with mine. I was able to still get my, I guess, I think if this was something we were really rolling out to where it was required, and this was something we put in schools, I feel like, I don't understand it enough that I would need more training. Because once it got into, and I don't know how Jenny's doing, I kind of almost have wanted to watch her part. Is she getting a lot of questions from the kids like once they get into it? […]
Modeling	An utterance coded as modeling occurs that prompts a teacher to make a related reflection utterance. That is, modeling activities encourage the teachers to reflect.	D: Yes, we put it in green. Okay, so next, do you notice this "for ever" loop? A forever loop will continually run all of those blocks below until we stop it. So, once you put this in the forever loop, which I kind of touched with, I don't know if you noticed with my classes last time, this week, is that it makes it that it goes forever. And if you click Start, it will go through the algorithm.
Suggestion-Based	An utterance coded as suggestion occurs that prompts a teacher to make a related reflection utterance.	[…] J: I would suggest that if you know people that do not work together that you let me know, because I'm not going to know. D: If you want for my class, they have partners. T: Yeah, so do mine. D: And it's a boy-girl, and I purposely put them with who they should be, and you can just say, hey, you're using Mrs. Bingham's partners that you use in your class. […] J: And I did notice, and I can't remember who it was. I can kinda remember where they were sitting. But there were a couple that I know are like lower-level thinkers too and they really struggled, because neither one of them were very smart. […]
No Noticeable Trigger	The reflection utterance that begins a reflection-based conversation (episode) occurs without any prompting. This may be due to the reflection occurring at	J: So, I asked them, one day, those of you who said you didn't like this, why didn't you like it? And I'm just curious. And a couple of them were like, I didn't like having to work with a partner. […]

	the beginning of the recording and therefore a lack of data to provide the trigger or the reflection occurring as an interruption to an existing conversation or as a non-sequitur.	J: But for a couple they're like, I know what I'm doing and you're just bugging me and— D: Oh, so they wanted to play the game on their own. J: They just wanted to go and do it on their own. And It was pretty funny, actually, because a couple of them that said that I was like, Yeah, I guess.

Table 12

Coding Book Defining the Content Types for Reflection Episodes

Code	Definition	Example
Teacher Affect	The reflections in the episode are primarily about how teachers felt in the context being reflected upon.	D: The second day of teaching it. Like I feel, it felt much better the second day I'm like, okay, it felt good. I definitely in the future you'll feel better once you practice it. Researcher: Yeah, that's, I think Jenny, she gets that iteration of doing it three times. D: Oh, yeah. That's I told her I'm like, I rocked it at your class. In my class I was like, [Researcher]. I'm forgetting something. [Researcher]. Help.
Teacher Actions	The reflections in the episode are primarily about teachers' actions. These are not framed as pedagogical choices, but rather general actions teachers took in the context being discussed.	[…] D: I felt bad how I like ended up doing it, I was like I hope this is okay. But I didn't want to ask you like as we were doing the lesson so I'm like I'm just gonna go with it and he can tell me later if it didn't work. T: And I was gonna jump in there and tell you but when I was going to run over there, you weren't in there and then I forgot about it. D: Well, and she ended up telling me about it, and so she was like here, borrow some of these magnets, and that's how I knew about your magnets. T: Yeah, okay. D: But she only gave me three— […]
Teacher Understanding	The reflections in the episode are primarily about what teachers understood in the context being reflected upon. This may be how they understood a concept or whether or not they understood it at all.	[…] T: Those. Well, I was reading through it at home trying to figure out, and my husband does programming at his job. And I said, Okay, you tell me what you think this means. And he's going, this looks like a master's thesis. You, they're, they're handing this to you and you know absolutely. And I said, I know, I know. I don't know what this means. So, you need to be put it in terms that, that was, that we, if we have to teach it, we need, either need the training to learn it. Or it needs to be explained in terms of, in layman's terms because that was like, and it took me about a half an hour to figure out how I could explain that to the kids, like the abstraction. I would have just said, okay, if I told you, sharpen your pencil, you know what that means. But the computer doesn't so, we would have done all of those teeny, little things to get this to over there,

		back to our desk, and have a sharp pencil. Okay, that's an abstraction. Right? So, but when I don't know any of that, and just reading that, that was like, huge for me, because I had to figure out what the heck you meant first.
Student Affect	The reflections in the episode are primarily about how students felt in the context being reflected upon. They may be about how students claim that they felt or how teachers believe students felt, even if these statements are not necessarily accurate.	J: So, I asked them, one day, those of you who said you didn't like this, why didn't you like it? And I'm just curious. And a couple of them were like, I didn't like having to work with a partner. Researcher: We […] were actually talking about that today, of the partners where I mean, like, I think for a lot of students the partners help and then for some students— J: But for a couple they're like, I know what I'm doing and you're just bugging me and— D: Oh, so they wanted to play the game on their own. J: They just wanted to go and do it on their own. And It was pretty funny, actually, because a couple of them that said that I was like, Yeah, I guess.
Student Actions	The reflections in the episode are primarily about how students acted in the context being reflected upon. It could be statements students made that are not specifically about their understanding, choices students made, or activities students did, among other types of activities.	M: Did, Jenny, did they mostly finish or— J: I think they all finished. Researcher: Yeah. J: I'm pretty sure they all did. I think there were kids that would have done more with more time, but, but they all finished, so yeah.
Student Understanding	The reflections in the episode are primarily about what students understood in the context being reflected upon. This may be how they understood a concept or whether or not they understood it at all.	J: They must have got it because when they came in here, they already knew. I didn't even have to do any explanation. D: And I was worried I was gonna mess it up. And they literally like, my kids were like on it. They were like, do this, do that. M: They wanted to tell me before, but I did the first level, like all by myself, they were like, Oh, we already know. D: And mine worked out perfect. I had a kid that made a mistake with the robot on the second one. And I was like, perfect. You went right into what I need. And I was like, Oh, look, my robot just went off the board. He's like, Oh, yeah, you need to turn left first. T: You're stuck there forever now. D: Yeah. And I was like, hey thanks. I'm glad you made the mistake for me.

Pedagogy	The reflections in the episode are primarily about pedagogical choices teachers made or the lesson plans themselves. Teachers may explicitly call these pedagogical choices, but do not have to.	Researcher: Yeah, they're all there. What did you guys think of the lesson plans last week? D: I liked it. […] D: And I tweaked mine a little bit too, just because I was like, oh, they're getting it. And so, they wanted to do the dancing one. But I was like, No, the diagonal one took the teachers... And it's like, well, how about we do the diagonal one because I had kids that did. So, we jumped to that one, just because it was a little bit more in detail how to do that. M: And that might have been good because we kind of ran out of time. D: And that was me, I was noticing the time was getting there. So, I'm like, Oh, I'm going to push them towards let's look at the diagonal, because that one for me was the harder concept one. And I just explained to them, I said with the dancing one, if I want me to turn this way, and they're like, Oh, you go 360. And I'm like, but what if I don't want to go full circle? They're like, Oh, it's only 180. So, they already kind of got that part, so I was like, let's go to the harder one, which was the diagonal, I thought. Researcher: Yeah.

Table 13

Coding Book Defining the Triggering Events for Connection Episodes

Code	Definition	Example
Researcher Initiated/Planned	A researcher makes a statement or asks a question that prompts the teachers to make a connection utterance. In addition, the lesson plan a teacher is modeling is pre-written to ask for them to make a connection (by researchers) and teachers therefore do so as prompted to by the researcher made lesson plan.	D: [reading the lesson plan] Alright. Ok, procedures? Where did we see procedures in the board game or on scratch? Who remembers? T: Those were all of our cards. M: I was curious, actually, did I use the word procedures the way you wanted me to in my lessons, you probably don't remember, but I used it the way I thought it would have been. Researcher: I think so. Yes. D: Because I think procedures were the one that were in the middle, right? […] T: All of our cards are procedures. Researcher: Yes, all of our cards are procedures. […] D: So, I could, so this one would be considered a procedure and then these cards also are procedures because they've already been defined.

		Researcher: Yep. D: Okay. Researcher: Perfect. You got it.
Question & Answer	A teacher asks a question and receives an answer that encourages teachers to make utterances coded as making connections. The connection may be the answer itself or a response to their peer's answers.	Researcher: So that part, whatever you see part start, that's kind of the main part of the program. And now what we want to do is add a part for the new green cards. T: Gotcha. And these are the algorithms, yes? Researcher: Yeah. T: Ok, I'm getting it. Researcher: Yep. So, what— M: Which one did you call the algorithm? T: The directions. […]
Reflection	An utterance coded as reflection occurs that prompts a teacher to make a related connection utterance.	D: I don't remember what the program is called, but our classes are already doing something that's a ton like this with Mrs. Hansen. Have you guys looked at, seen it at all? J: She's said, they're doing— D: It's not Scratch. J: No, last year we did, I think they're doing coding.org. D: No, it's something different, so I went in and I was like this looks a ton, so it's the same type of game that he's— M: Oh, it's a mines, it's like a Minecraft type thing— […]
Modeling	An utterance coded as modeling occurs that prompts a teacher to make a related connection utterance. Note that if the connection occurs because the lesson plan being modeling requests the connection be made it is "researcher initiated/planned." If the connection occurs as the teachers make an aside or add connections to their modeling, then this code applies instead.	M: Okay. How does the sprite know what move forward is, If I'm the sprite and you tell me to move forward, I don't know what that means, how do I know what that means? Somebody raise their hand. T: You've got to go to, no, you have to go to the algorithm, go down. D: Is it the algorithm one? T: Yeah, cuz that defines what a— M: Oh, right— T: There you go. M: Right here? D: Yeah. T: Because that tells it what to do for a turn right or a walk forward. Researcher: So, then the next one would be, another one would be like if you go off the path, then you lose, like the event would be moving off the path. Or if you have reached

		the edge, then you lose or if you reach finish then you win, would be examples of events. T: So basically, you're saying if statements are events. Researcher: Yes, is where we're going to go to […]

Table 14

Coding Book Defining the Content Types for Connection Episodes

Code	Definition	Example
CS content to other CS content	The connections in the episode are about how one CS concept relates to another CS concept.	Researcher: So, then the next one would be, another one would be like if you go off the path, then you lose, like the event would be moving off the path. Or if you have reached the edge, then you lose or if you reach finish then you win, would be examples of events. T: So basically, you're saying if statements are events. Researcher: Yes, is where we're going to go to […]
CS content to the Game	The connections in the episode are about how a CS concept relates to the board game, whether in its tabletop iteration or the Scratch iteration.	Researcher: So, level 15 would be right here. Okay. Right. And so, looking at level 15, we would say what are some of the events that you see? And there's two ways to answer this, right? One of this we could say, actually, we'll just have you guys, what are the events that are on here? M: I think as a kid, I would be like, wait what do you mean, like what— J: Yeah, I wouldn't understand what the word events means either. Researcher: Okay. So— D: Well, if it's, like what you're saying with the games, are you meaning like when I land here, I need to turn left, is that what you're wanting as an event? Researcher: Yeah. T: Or this guy has to turn right or left before he— M: Yeah, they're gonna start telling you what to do, I think. Researcher: Yeah. So, one option would be— D: And is that considered an event? Researcher: Yeah, I mean, that would be an event, and probably what I would consider is that if you're on red, do the red cards.

		M: So, landing on red is, or being on red is the event. Researcher: Yeah. So being on red is the event.
CS content to an Analogy	The connections in the episode are in the form of analogies that can be used to describe CS content.	M: I mean, I don't think I knew the word abstraction. T: The kind of all, I mean, is like, well, the procedure isn't a small abstraction. M: Yeah, maybe I was actually like, so procedure was like all the steps, right? And abstraction is like the act of doing it. Maybe I wasn't, I don't know if I taught that completely correctly. T: You could have really big abstractions— M: Well, cuz like I said, like I gave a similar example about the sharpened pencil like, when I say get out your Elevate stuff, we know that means get out this, this, this, start your language sheets, like you don't have to say all those steps. T: Right. M: So, I kind of said that, but— T: Yeah, and the algorithms are all the instruction M: And those are the procedures, the abstraction is like the act of making it smaller? Researcher: Yeah. So, I, if you want to jump in. I have my definition of it. But...
Student Actions	The connections in the episode are about how student actions relate to each other or to other contexts with which the teachers are familiar. The purpose of these connections is to understand how the students are acting and why they are making those choices.	M: We can name them whatever we want but we still have to tell the computer what it's going to mean. D: Yeah, so if they wanted to put in drift— Researcher: Yeah D: —that's obviously not going to tell it what it's going to do, but they can name the card drift. M: Okay, we're gonna name it something useful.
Scratch to the Board Game	The connections in the episode are about how the Scratch iteration of the board game compares to the tabletop iteration of the game.	Researcher: The one last thing that we have on here, and I don't think they're going to be ready for this. So, we might want to scratch it. So, if you go up to the top where we have our cards, so it, notice that these are empty here. Before them, for somebody else to play their game, we need to drag in the cards that— […] J: But if I sit down to play the game, I need to be able to be like what's gonna have, I've got to put the cards in, that's what you're saying. Researcher: Yeah, cuz you need to, if you just say that it could be any card, that's really hard. Right? D: Oh, so like you did for us. You've always put the list of like the eight cards that you're going to need.

		M: So, like these?
		Researcher: Yeah.
		D: Or like on the board game, it says for each level, you need two of these, okay.
Past Experiences	The connections in the episode are about how the curriculum compares to teachers' past experiences, whether personal or professional. The purpose of these connections is to contextualize what a teacher is seeing for the first time in terms of experiences with which they are familiar.	Researcher 1: And so, one thing to make sure that they, it's rotate only, like it's not like move forward and turn—
		Researcher 2: Your kids will do this. They'll do this. They'll do this a lot.
		D: Oh, gotcha. So, you can't move forward it's just a rotate.
		M: And they did do pretty good at that, because we had the STEM Bus people come and that's what it was, and they told each other what to do. And they had to like do it in loops and stuff.
		Researcher: What might be a good thing to help them see that is to actually do this for them. And do something like this.
		D: Okay.