COMPUTER SCIENCE LESSON STUDY: BUILDING COMPUTING SKILLS

AMONG ELEMENTARY SCHOOL TEACHERS

BY

THOMAS R. NEWMAN

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

EDUCATION

UNIVERSITY OF RHODE ISLAND

AND

RHODE ISLAND COLLEGE

2017

ProQuest Number: 10260505

ProQuest 10260505

DOCTOR OF PHILOSOPHY DISSERTATION

OF

THOMAS NEWMAN

APPROVED:

Dissertation Committee:

Major Professor   Kathy Peno

Peter Adamy

Anne Goodrow

Kathryn Sanders

Donald Halquist
Dean, Feinstein School of Education - RIC

Nasser H. Zawia
Dean, The Graduate School - URI

UNIVERSITY OF RHODE ISLAND
AND
RHODE ISLAND COLLEGE
2017

ABSTRACT

The lack of diversity in the technology workforce in the United States has proven to be a stubborn problem, resisitng even the most well-funded reform efforts. With the absence of computer science education in the mainstream K-12 curriculum, only a narrow band of students at public schools go on to careers in technology. The problem persists because computer-science reforms focus primarily on a small percentage of high school students rather than the majority of K-12 students, despite evidence that computer science can help early learners develop valuable thinking, problem-solving and social skills. The purpose of this research is to examine how elementary school teachers use a collaborative lesson study process of professional development, *Computer Science Lesson Study*, to acquire computer science content knowledge and teaching skills.

This qualitative action research study investigates how elementary school teachers, working with a computer-science professor from a local university, worked collaboratively over a twelve-week timeframe to teach computer science and computer programming lessons to over one hundred students in third-grade classes. The study took place in an urban elementary school serving students from minority groups underrepresented in the technology workforce. The findings indicate that *Computer Science Lesson Study* provided a high-quality professional development approach for the introduction of computer science to the elementary school curriculum.

ACKNOWLEDGMENTS

I am truly grateful to my family, and many friends and colleagues who have supported me throughout my graduate studies. The professors and students in the URI/RIC PhD program have made the last four years very special. I would like to begin by thanking my major professor, Kathy Peno, and committee members Peter Adamy, Anne Goodrow and Kate Sanders for their encouragement and guidance over these years. I would also like to thank chairperson Joan Peckham for being there to contribute her extensive knowledge of computer science education and chair the proceedings. Finally, a special thanks to the school principal, teachers and students who worked on this project with me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

.

CHAPTER 1

Introduction

Innovations in computers and technology over the past twenty years have transformed our world. Computer technology plays a primary role in virtually all aspects of our lives including security, finance and health care, yet computer science is underrepresented in the K-12 curriculum (Google, 2015; Wilson, Stephenson, Sudol, & Stehlik, 2010). Despite an increasing demand for more graduates trained in computer science (CS), colleges and universities are failing to produce sufficient numbers of graduates ready for the estimated 1 million job vacancies in technological fields by the year 2020 (Code.Org,2016; EEOC, 2016; Google; 2015).

In response to growing pressure from industry and parents a bipartisan federal CS education reform, Computer Science for All (CS4All), was announced by then President Obama on January 30, 2016. President Obama acknowledged CS as a foundational skill that all students will need for success in the 21st-century and observed that 9 out of 10 parents see it as imperative to their child's education. The CS4All initiative provides leadership and funding for states to design and implement CS education programs in public schools (Harsha, 2016). At a local level, Rhode Island Governor Gina Raimondo announced the Computer Science for Rhode Island (CS4RI) initiative that hopes to make computer science education available in all schools from K through 12, by December 2017 (Tempera, 2016).

Statement Of The Problem

Increasing access to CS among public school students is a stubborn problem in need of solutions that work at many levels. Achieving the objectives of CS4RI(2016) will require overcoming significant barriers, including a critical shortage of teachers who are qualified to teach CS, the lack of understanding among school administrators of what CS education entails, and a curriculum already packed with ongoing initiatives.

Because CS is unfamiliar to most teachers, schools are unable to provide CS education. Because schools are busy with initiatives focused on raising student achievement scores in math and reading, little time is available for teachers to attend CS seminars and workshops. Without a program of CS professional development (PD) that fits within the time and resources available, schools will be unable to provide students with the fundamental skills that they need to understand computer technology. This study examines *Computer Science Lesson Study* a teacher PD activity designed to equip teachers with the content and pedagogical skills they need to teach computer science. *Computer Science Lesson Study* is designed to provide a sustainable PD approach for elementary schools to bring computer science to the classroom.

Connection to my Professional Context

I began my career as a computer science teacher in 1987 as a CS professor teaching evening courses in computer programming to adult learners. At that time I had been working for three years as a computer programmer and systems developer

for a rapidly expanding software company involved in the medical industry. I became a CS teacher because our business was not able to hire enough programmers to keep up with the software development workload. In pursuing my professional development through the URI/RIC Ph.D. in Education program I hope to contribute my experience in industry and technology education to efforts to expand CS into the mainstream curriculum in the United States. As a CS instructor working in a local university, I have found that very few students in my classes have had any previous CS or programming experience. In my view effective CS reform will result from research aimed at all stages of the K-12 pipeline and key questions such as: how we help our students understand CS and prepare them for success in the 21st-century; how we can help students to become innovators and inventors; how CS, as a literacy skill, deepens students' understanding in other subject areas.

CS in K-12: Perceptions and Expectations

Over the past three decades computer science has struggled to make inroads into the mainstream curriculum in the United States. Students often consider CS boring and tedious (Taub, Armoni, & Ben-Ari, 2012). Computing researchers blame the shortage of qualified CS teachers on confusion over what computer science is, and misconceptions among students (Prottsman, 2014). In 2011 the Computer Science Teachers Association (CSTA) began to address the confusion with the release of the *CSTA K-12 Computer Science Standards* (CSTA, 2011). The standards provide a blueprint for implementing CS programs which include the fundamental five strands of computer science: computational thinking, collaboration, computing practice and

programming, and community, global, and ethical impacts. Since their release the CSTA standards have been used to inform computer science initiatives around the world. In addition to providing a blueprint for CS K-12 education the CSTA standards stress the importance of teaching the fundamentals of computer science from the time children start school (CSTA, 2011). While the CSTA K-12 Standards have made a significant contribution by providing a clear definition of CS and 21st-century skills, implementation barriers to CS reform have continued to delay the introduction of CS in most schools in the United States.

Recent efforts to breach the chasm between computer science and the mainstream K-12 curriculum are generating publicity and increasing awareness (Code.Org, 2016). In the United States parents, teachers and students are expressing a growing interest in computer science, as reflected in the numbers of students participating in the "Hour of Code" sponsored by Code.Org. The success of "Hour of Code" shines a bright light on the CS education reform landscape. According to Code.Org (2016), since its release in 2014, the K-8 courses from Code.Org have been used in more than 31,000 classrooms worldwide. Lesson materials available through Code.Org are designed to emphasize fundamental CS skills outlined in the CSTA K-12 Standards, such as computational thinking and algorithm development; community, global and ethical impacts; computer programming, pair programming, and computer engineering and communications. Despite the sustained effort to raise awareness through the "Hour of Code" CS continues to occupy a marginal place in the K-12 curriculum in the United States.

What are 21st Century Skills?

Many voices are calling for schools to teach skills that prepare students for success in the 21st-century. While the influence of technology on our lives is driving concerns that students acquire new and different sets of skills, questions arise of what exactly these 21st-century skills are and exactly how they are to be taught. Looking for answers to these questions reveals a robust discussion that began to develop in the mid to late 1990's. Fortunately, work to clarify what 21st-century skills consist of has had the attention of an international group of computing educators, researchers and CS advocates, aided by organizations such as the Computer Science Teaching Association (CSTA) and the National Science Foundation (NSF). Since 2012 publications of the CSTA have, for the most part, addressed the critical need for a precise definition of CS and a CS K-12 curriculum framework. While the question of identifying 21st-century skills is becoming clearer now, that of how to teach CS in K-12 is still unanswered. In general, 21st-century skills include computational thinking and problem-solving; promotion of creative and innovative ability; communication, collaboration, and computer programming at some level. Computational thinking, collaboration and problem-solving skills overlap with other disciplines. Computer programming is unique to CS. Trilling and Fadel (2009) observed that for teachers to be effective in promoting 21st-century skills, they must refocus their ideas and concepts and rebalance their time between being the "sage on a stage," presenting content, and serving as a "guide on the side," supporting students' research and discovery. As one teacher put it, "I had to unlearn the idea that teaching was about my content; I had to learn it was about their thinking and their skills" (Trilling & Fadel, 2009, p.39).

الم‍ن‍ا‍رة للاستشارات

www.manaraa.com

CS Teacher Shortages

University graduates who possess CS knowledge usually find their way into industry rather than teaching professions. This means that the supply of teachers with CS knowledge is severely limited (ABCTE, 2015). For K-12 public schools hiring an adequate supply of teachers to implement CS initiatives is simply not possible. According to the American Board for Certification of Teacher Excellence (ABCTE, 2015), the shortage of teachers qualified to teach computer science, as well as all STEM subjects, is affecting student learning. Restrictive licensure requirements in most states and the high turnover rate for teachers also contribute to the shortage of computer science and STEM teachers (Ericson et al., 2008).

In 2014 a report prepared by an international group of computing researchers assessed the state of CS education reform across all regional and national boundaries (Hubweiser, Armoni, Giannakos, & Mittermeir, 2014). The report, based on very detailed case studies and research prepared by members of the working group, compared CS reforms in Austria, Bavaria, France, Greece, Israel, Lithuania, Sweden, New Zealand and the United States. The researchers found a wide variety of K-12 school systems and significant differences between the context and conception of CS regarding organizational issues, learning objectives, teaching methods and other important aspects. Working to reconcile CS across international boundaries, the researchers developed the *Darmstadt Model* of categories and themes to facilitate international research in CS education. In addition to providing a basis of comparison

6

of CS across a variety of educational systems, the researchers identified critical factors in successful CS reform.

Trends in Technology & Education

Over the past five years computer advocacy groups and policymakers have promoted CS in K-12 yet focused resources primarily on students nearing the end of the educational pipeline so that they might move into the technology workforce upon graduation (CINC, 2012; Code.Org, 2016; CS4RI, 2016). For example, in their annual report for 2012, the advocacy group Computing In The Core (CINC, 2012) expressed support for member Microsoft in its efforts to develop and finalize its National Talent Strategy. Microsoft's strategy outlines a plan that is focused primarily on increasing the number of workers in the technology industry by expanding CS in high school and increasing the number of foreign technology workers by easing visa restrictions. (Microsoft, 2012). By focusing a national CS strategy on Microsoft's and the computing industry's immediate workforce needs, the CINC hoped to improve the nation's STEM education infrastructure and support student success in postsecondary pursuits.

CS reform initiatives over the past two decades have not been able to reverse the trends that show women and minorities underrepresented in the technology workforce (EEOC, 2016; Google, 2015). In 2016 the Equal Employment Opportunity Commission (EEOC, 2016) compared employment patterns in the high-tech and private industries. EEOC reports that high-tech employs a larger share of whites, Asian Americans and men and a smaller share of women, African Americans and

7

Hispanics than does private industry. Figure 1 shows the lack of diversity within the high technology workforce, as reported by the EEOC (2016).
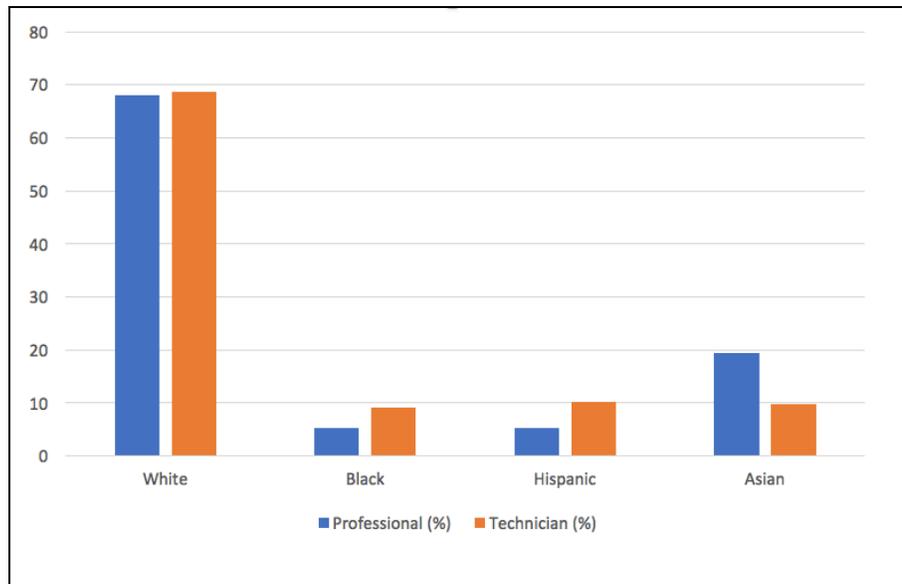
Figure 1. **Professionals & Technicians in high-tech by race and ethnicity**
Source: Equal Employment Opportunity Commission, Employer Information Reports (EEO-1 Single, Headquarters, and Establishment Reports, 2014)

The EEOC (2016) describes *Professionals* and *Technicians* as categories that make up approximately 54% of the jobs in the technology sector, as compared with 25% in all industries combined nationally. Professional occupations include software engineers, computer programmers, web developers and database administrators. Technical occupations include electro-mechanical, electronics, electrical engineering, medical records and healthcare technicians. Jobs in the technology sector tend to provide higher pay, better benefits and are more resilient to economic downturns. EEOC (2016) notes that employment in computer science and engineering is growing at twice the rate of the national average and have a strong potential for continued growth.

The positive employment outlook and continued growth of the technology sector is accompanied by concerns that the US educational system is failing to supply and adequate number of workers to support the expansion and development of new products (EEOC, 2016). EEOC (2016) reports that 80 percent of women in the technology workforce say they love their jobs but 32 percent feel stalled and are likely to quit within a year. Williams (2015) identifies five biases that push women out of technology jobs including: having to prove themselves over and over, pressure to play a traditional feminine role, having their commitment to the job questioned after having children, having to compete with women collegues for "the woman spot" and isolation. The high turnover of women tech workers not only contributes to the overall shortage of workers but also to the underrepresentation of women in leadership roles. Figure 2 shows how gender bias in the technology workforce results in the underrepresentation of women in leadership positions.

Figure 2. **Women in leadership positions and tech jobs in US High Tech**
Source: Equal Employment Opportunity Commission, Employer Information Reports (EEO-1 Single, Headquarters and Establishment Reports, 2014)

9
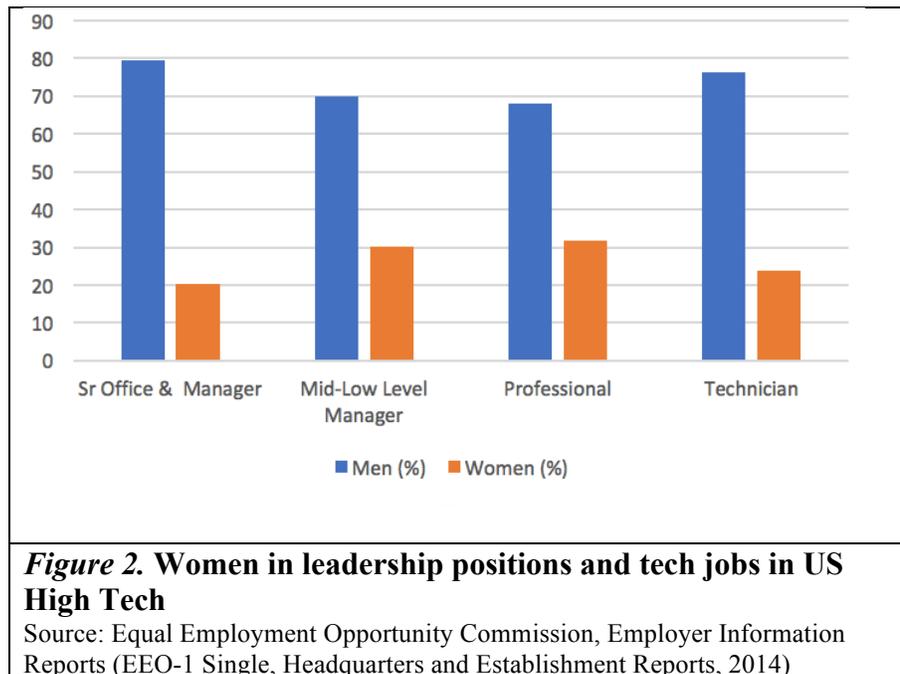
One of the most troubling findings, per the EEOC (2016), is that qualified female and minority workers often choose to leave the technology workforce because they do not feel welcome in the culture of the workplace. Therefore, just increasing the number of students interested in technology careers is not enough. Solving the diversity problem in the technology industry will require new ideas and different approaches. This study proposes a CS PD activity that will provide CS to students at elementary school. Expanding the number of students who are introduced to CS early in their education will, over time, increase the number of post-secondary students entering the technology workforce.

Changing the strategy for CS teacher PD will not only help spark interest in CS among more students but may also help them develop skills useful in other subjects. According to the National Science Foundation (NSF, 2016), although the percentage of fourth, eighth and twelfth-grade students achieving proficiency or higher on National Assessment of Educational Progress (NAEP) mathematics assessments increased between 2000 and 2013, those percentages stayed well below the 50% mark. The NSF also reports that, overall, students from disadvantaged backgrounds continue to lag behind their more advantaged peers, with these disparities starting at kindergarten (NSF, 2016). Rather than remaining just another educational reform initiative, teaching CS at early elementary school encourages collaboration, perseverance and problem solving that can help students develop social skills and analytical strategies, such as connecting mathematical and artistic concepts (Clements, 2002). Because learning CS at an early age can engage students in skills that overlap

with math, science and reading, providing CS to all students in primary grades might begin to close the achievement gap between US students and their international peers.

There are an increasing number of computing researchers who argue for an increased focus on early learners. Some say that it is even more important to focus CS reform on elementary schools because learning CS helps students develop cognitive skills. For example, Yongpradit (2014) argues that coding puzzles and games help elementary school students express their imagination and build creativity, collaboration, persistence and thinking skills. Bers (2010) used educational robotic kits to provide students with a new generation of "manipulatives" that build on the tradition of Montessori and Froebel. Bers (2010) suggested that projects with robotic kits help students develop both a basic understanding of the fusion of electronics, software, and mechanical structures as well as a deeper understanding of number, size, and shape. Bers (2010) also said that through class projects in robotics, students develop in-depth knowledge that may provide a gateway to learning applied mathematical concepts.

The lack of CS in schools raises issues of access and privilege. According to Goode and Margolis (2011), the narrow band of students who go through K-12 and move on to careers in technology typically have families that provide them with computers and software throughout their lives. Until schools in the US begin to teach CS, starting at elementary school, the persistent lack of diversity among the technology workforce is likely to remain a stubborn problem.

11

Overview Of Research Design

Computing pioneer Seymour Papert (1980) proposed that CS in primary education has the power to open students' thinking to powerful ideas such as "finding the beauty in abstract things" (p. 10), or "mastering the art of deliberately thinking like a computer" (p. 27). The capacity of young children to grasp advanced concepts is also a central theme for Bruner (1960), who supported the social constructivist view, that social interaction is essential to the development of new knowledge. According to Bruner (1960) children of any age are capable of grasping the underlying structure of how one thing is related to another, allowing them to relate many other things in similar ways. A proponent of constructivist approaches to teaching, Bruner and his colleagues were instrumental in transforming K-12 science and mathematics education in the US. During the cold war era policymakers and industry called on the US education system to produce more students qualified for technology careers. In September 1959 a group of thirty-five scientists, scholars and educators met in Woods Hole, on Cape Cod, Massachusetts to discuss a long-term strategy to improve science education in the United States (Bruner, 1960). The meeting was not intended to develop a crash course, or a strategy to generate more students ready for jobs in industry. Rather, the group was concerned with imparting a sense of the substance and method of science through a new curriculum. Bruner and his colleagues were focused on a longer-term strategy that would increase the quality of the science taught in schools and align it with the longer-term needs of industry. Today we see a similar gap between the advanced technologies used in industry and that taught in schools. The design of this research rests, in part, on the assumption that to increase the number and

12

diversity of technology workers will require a long term strategy involving collaboration between CS experts and teachers in urban elementary schools.

Action research is a method that has been in use in the social and medical sciences since the mid-twentieth century. Because it is grounded in practical action and aimed at solving an immediate problem, action research is used extensively for scholarly investigation of topics in information and technology (Baskerville, 1999). CS Lesson Study is a form of action research designed to engage teachers and school administrators in inquiry-based classroom research focused on CS. Ultimately CS Lesson Study aims to engage teachers in a collaborative process focused on instructional improvement in CS.

The action research design based on lesson study is appropriate because teachers participate as co-researchers in developing and refining the CS Lesson Study process itself as well as the CS curriculum. Urban public school districts work with limited budgets and need assistance in finding high-quality materials. The resources and materials used in this study were carefully chosen to ensure that they would be both efficient and readily available at no cost to teachers or school district. Whenever possible the methods, procedures and terminology used throughout this study were adapted from educational, professional development resources with which teachers may already be familiar. For example, this research draws from lesson study practices used by teachers in Japan, Europe, the UK and the US to build skills in math and science (Dudley, 2011; Lewis & Perry, 2014). Similarly, rather than develop new materials, the teachers in this study used software and computer science resources

13

available through Code.Org, the DevTech research group at Tufts University, and the Lifelong Kindergarten group at the MIT Media Lab.

Carr and Kemmis (1986) describe action research as a form of self-reflective inquiry undertaken by participants in social situations to improve their practices, their understanding of these, and the situations in which they are carried out. In this study the researcher participated in the role of CS instructor to help teachers develop their understanding of the key fundamentals of computer science and how to implement a lesson study process using a research lesson based on Scratch Jr.

Scratch Jr is a programming language that is designed to put young children in control of computers, using them to experiment, explore and express themselves. Scratch Jr was conceived and developed by Marina Bers and Mitch Resnick in collaboration with the Lifelong Kindergarten research group at the MIT Media Lab and the DevTech research group at Tufts University (Bers, & Resnick, 2016).

CS Lesson Study is designed to help teachers build the content knowledge and pedagogical skills they need to teach CS. It facilitates a balanced transition from building CS content knowledge to CS teaching and learning. Figure 3 provides a conceptual diagram showing the transition from CS knowledge to CS Pedagogy over the course of a CS Lesson Study project.

14
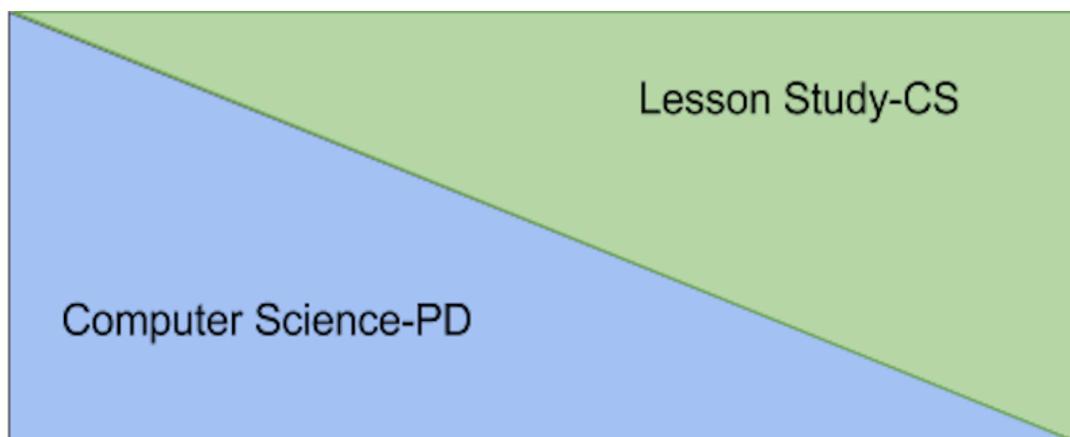
8/7— CS PD Sessions ---------------------- 10/4 – CS Lesson Study Sessions -------------11/2

Figure 3. **Conceptual diagram – CS Lesson Study**

Key: In this diagram the lower (blue) section represents the initial focus on CS skills. The upper (green) section shows that lesson study becomes the main focus once teachers have acquired fundamental CS skills.

CS Lesson Study consists of two sections, CS PD Sessions and CS Lesson Study Sessions. Section I - CS PD Sessions, consisted of six (6) CS PD Sessions held online via Webex online meeting software. CS PD Sessions provide teachers with foundational knowledge in computer science, computer programming and lesson study. During weekly meetings the CS Instructor (the researcher in this study), teaches an introductory course in CS and computer programming. CS PD Sessions include readings, discussion, programming assignments and lesson planning. Throughout the 6-week CS PD Sessions teachers work together to prepare a CS research lesson. In Japanese lesson study, a research lesson is an actual classroom lesson plan that teachers develop and refine through a process of teaching, reflection, and re-teaching.

Section II - CS Lesson Study Sessions involved a collaborative lesson study process in which the teachers teach, revise and re-teach the CS research lesson to five classes of third-grade students. Lesson study is a collaborative form of PD used by

teachers to examine their practice by jointly planning a single research lesson on a topic. A lesson study cycle consists of four steps that teachers follow to improve teaching and student learning (Figure 4).
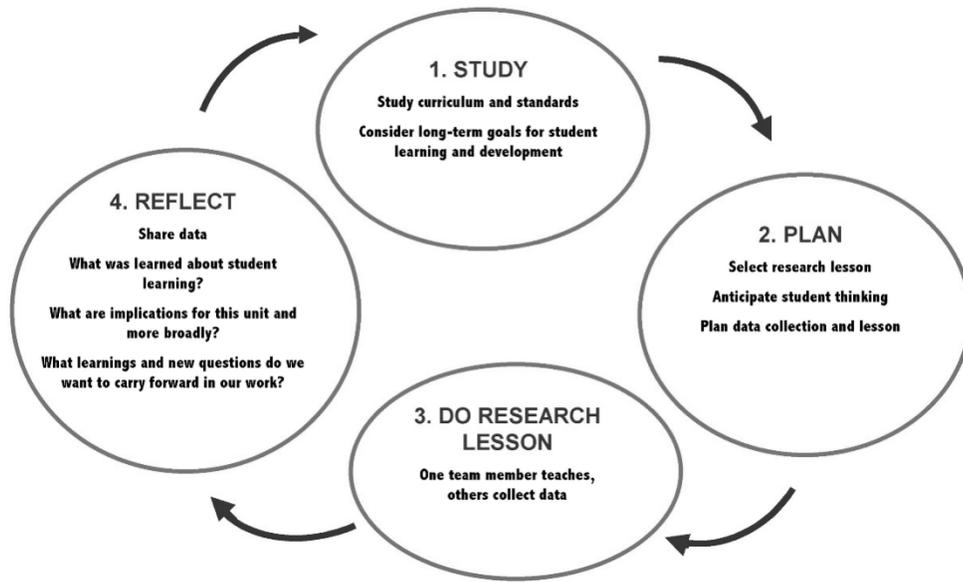
Figure 4. **The Lesson Study Cycle**
From: *Lesson Study Step-by-Step*, Lewis & Hurd, 2011

Lesson study provides teachers with a chance to work with their peers to improve teaching practice. Within a lesson study cycle the teachers begin by reviewing curriculum and student objectives. Next they plan the lesson and develop a list of anticipated questions that students might ask. In the third step one teacher teaches the lesson while others observe and collect data. In weekly *reflection*

16

meetings, lesson study step four, teachers discuss their students' progress and revise the research lesson.

In this study data collected during teaching sessions included audio recordings, lesson materials used by teachers, examples of students' Scratch Jr programs; coding sheets (pre-printed forms that students used to plan their algorithm); questions the students raised during the discrete parts of the lesson; obstacles that arose; and specific techniques that the teacher used to encourage understanding.

During lesson study teachers develop content and pedagogical skills that go beyond the immediate topic (Lewis, Perry, & Hurd, 2004). By providing a collegial forum in which teachers focus on student learning, lesson study addresses the students' long-term development, responds to their eagerness to learn, and ability to work with others. The reflection discussion, step four, is held shortly after the teaching session to share the data and discuss student learning before the next teaching session. The next section outlines the CS PD and CS Lesson Study sessions for this study.

Section I: CS PD Sessions

CS PD Sessions were held between August 2, 2016, and September 14, 2016. CS PD Sessions included presentations, discussion, and exercises on the following topics:

1. Computer architecture and instruction sets.

2. Computer history and significant events.

3. Binary numbering.

4. Algorithm development and program constructs.

5. Current and former computer-science education initiatives.

6. Computer Programming with Scratch Jr.

7. Computational Thinking

8. Lesson Study process and procedures.

Section II: CS Lesson Study Sessions

CS Lesson Study Sessions were arranged as five weekly lesson study cycles that took place from October 2, 2016 through Nov 1, 2016. Since the CS research lesson required two teaching hours, the lesson was taught in two one-hour sessions over two consecutive days (Classroom Teaching Session Day1, Day2). By the end of the project 104 students in five third-grade classrooms had learned about computer science and computer programming from their teacher and begun to develop coding skills through collaboration with fellow students.

Theoretical Framework

This research views the process of teaching and learning through a social constructivist lens. Rather than primarily copying or receiving information from others, constructivists view learning as a process in which humans construct and reconstruct their own mental conceptions of the world through experience (Angelo, 2001). Social constructivists view the complex interactions and relationships within a society as the primary source of information that individuals use to construct knowledge and assign meaning directed at certain objects and experiences (Creswell, 2014). In social constructivist research, researchers focus on the context in which individuals live and work. Through observation, and interaction with participants,

social constructivist researchers seek to understand the processes that individuals use to develop subjective meanings based on historical and cultural norms (Creswell, 2014). Because CS is an unfamiliar topic for most teachers working in US elementary schools, this research study relies on prolonged observation and interaction between the participants and the researcher to develop an understanding of how teachers construct and reconstruct CS content and pedagogical knowledge. Social constructivist researchers rely, as much as possible, on the participants' view of the topic being studied. By listening to what members say and observing what they do, social constructivist researchers develop their own impression of the situation (Creswell, 2014). Because subjective meanings within a situation are not just imprinted on the individuals involved, but emerge through social interaction, researchers look for a complexity of views, categories, and ideas (Creswell, 2014).

Constructivist research, including lesson study, grounded theory and action research, is influenced by the work of pragmatist philosopher John Dewey. In their study, "The Discovery of Grounded Theory," Glaser and Strauss (1967) credit the work of Dewey with the theoretical underpinnings of grounded theory. Glaser and Strauss invented grounded theory to bring studied control to Dewey's theory of knowledge arising from social situations (Glaser, & Strauss, 1967). Glaser and Strauss (1967) explain that studied control of qualitative research meant adopting a scientific approach toward data collection and analysis. Glaser and Strauss explain that grounded theory coincides with Dewey's concept that "applied science" means applying scientific processes 'in' the research, rather than applying science 'to' the research. Application 'in' something, according to Dewey (1925) "signifies a more

19

extensive interaction of natural events with one another, an elimination of distance and obstacles; provision of opportunities for interactions that reveal potentialities previously hidden and that bring into existence new histories with new initiations and endings." (Dewey, 1925, pp. 161-162). Dewey (1925) viewed the application of scientific procedures in connection with existential affairs as a way to yield insight and understanding by "filling events with coherent and tested meanings" (p.163).

Dewey's influence on teaching and learning extended beyond the United States. His work with Japanese educators began with a visit to Japan in 1920. Dewey's lectures in Japan and China started a dialogue among educators that eventually led educators in Japan to develop lesson study as a practical approach to teaching (Wang, 2007). Following a series of lectures at the University of Tokyo, Dewey spent the next two years in Asia. After World War II Japanese educators refined the practice of lesson study that had begun in the 1920's (Baba, 2007). Over the next fifty years lesson study became the predominant form of teacher PD in Japan (Murata, Bofferding, Pothen, Taylor, & Wischnia, 2012). The success of lesson study, as a constructivist research method, depends primarily on the participation of teachers, rather than on the PD designers, developers, or content experts (Stigler, & Hiebert, 2016). Lesson study is unlike teacher PD approaches that focus on assessment of individual teachers. In this research, lesson study is intended as a vehicle that teachers will use to develop a shared understanding of CS and to establish a teacher-led program for CS within the school.

As a constructivist practice, lesson study improves teaching and fosters a professional education community while providing a learning structure based upon an

inquiry process (Lewis & Hurd, 2011). Lesson study is unlike forms of teacher PD that convey information through workshops and seminars attended by individual teachers. Because lesson study emphasizes an ongoing collegial relationship among teachers, focused on student learning, PD becomes an integral part of the professional learning culture within the school, rather than something provided by outside entities.

In addition to sowing the seeds for lesson study in Japan, Dewey influenced the development of action research. After fleeing from Berlin in 1933, Kurt Lewin developed action research as a way of helping Jewish refugees establish new lives in Palestine (Adelman, 1993). Sponsored by Eleanor Roosevelt and others, Lewin developed close ties with John Dewey, Edward Thorndike, Frank Boas and many other academics and philanthropists (Adelman, 1993). Dewey's view that teaching and learning are improved through a democratic, collaborative process, forms the theoretical basis for action research as a method of systematic inquiry used by groups of individuals as a means of resolving intractable problems (Adelman, 1993).

Dewey's ideas continue to have a strong influence on education reform. New technologies and advancing knowledge across a range of disciplines creates the need for change in educational systems. Increasingly we see alternative PD approaches becoming more common as teachers are unable to develop knowledge of rapidly advancing subjects through traditional PD (Elliot, 2016). In the UK Elliot (2016) describes a transformation of teacher PD based on Dewey's apprenticeship model of teacher education. According to Elliot (2016), the shift toward an apprenticeship

approach was needed as the traditional approach to teacher training had proven unworkable.

Teachers who do not have the opportunity to practice teaching in classrooms, or learn from more experienced teachers, are unable to convey abstract concepts to their students. Elliot (2016) explains that Dewey's apprenticeship model links the development of a teacher's theoretical knowledge with that of teaching practice through their professional experience as learners in the classroom and beyond. The apprenticeship model rests on Dewey's view that teacher education consists of developing warranted beliefs about the relations between teaching and learning. Warranted beliefs are beliefs that teachers test through experimentation in classrooms. The apprenticeship model conceives of classrooms as a laboratories in which educational beliefs are tested. In turning to an alternative approach to teacher PD, educators in the UK realized that knowing how to teach and what to teach are entirely intertwined with relational outcomes (Elliot, 2016).

Research Questions

The goal of this investigation is to examine how teachers use a CS Lesson Study PD approach to build content and pedagogical skills in CS. Specifically, the purpose of this study is to determine whether CS Lesson Study PD provides a cost-effective method to provide computer science to all third-grade students and teachers at a large urban elementary school. The research questions are:

RQ1. How does CS Lesson Study influence CS instructional planning by elementary school teachers?

22

RQ2. How does CS Lesson Study influence CS instructional effectiveness among elementary school teachers?

RQ3. How does CS Lesson Study among teachers influence student CS learning?

Data Collection & Analysis

CS Lesson study involves the planning, teaching and reteaching of a single CS research lesson in each third-grade classroom. In CS Lesson Study Section I teachers participated in online CS PD Sessions designed to introduce lesson study and build content and pedagogical knowledge in CS. In CS Lesson Study Section II teachers worked collaboratively to follow the lesson study process that included classroom teaching sessions and lesson study group meetings. During a classroom teaching session one teacher volunteered to teach the lesson while the other members of the group acted as observers. Data collected throughout the study include:

1. Pre-study and post study interviews with participants.

2. Webex recordings of online CS PD meetings.

3. Audio recordings of classroom teaching sessions.

4. Audio recordings of Lesson Study meetings.

5. Email and other correspondence between the researcher and participants.

6. Lesson study plans and materials posted to a shared folder on Google Drive.

7. Examples of student work including Scratch Jr projects.

8. Researcher journal.

Grounded Theory With Action Research

Action research arises from the need to work together to address critical problems (Reason, & Bradbury, 2008). Within an action research project communities

of inquiry and action evolve as participants address significant questions and issues. In fields where there is little empirical research available, researchers use grounded theory methods within action research to promote analytical rigor and validity (Pauleen, & Yoong, 2004; Wastell, 2001). For this study the combination of action research and grounded theory provided two essential benefits. Firstly, regarding teaching CS to elementary school teachers, grounded theory methods helped the researcher to analyze incoming data, develop practical knowledge and implement course corrections to the ongoing CS PD and CS Lesson Study Sessions. Secondly, grounded theory procedures helped the researcher understand the situational realities of the elementary school system.

Both grounded theory and action research are iterative processes designed to help researchers develop new approaches based on data collected as a situation unfolds (Charmaz, 2014; Dick, 2003). According to Dick (2003), grounded theory methods improve action research by formalizing the theory-building process within the cyclic action research process. Previous research suggests that grounded theory analysis in educational action research can invigorate and engender a sense of control and ownership among participants involved in action research in school (Butterfield, 2009). Butterfield says the use of grounded theory as an inductive method in action research adds value because participants can monitor, evaluate and change the intervention strategy in the action research cycle. The process of data collection and analysis for this research is characterized by: simultaneous data collection and analysis throughout the research; constructing analytic codes and categories from the data; constant comparison of data at each stage; memo writing to elaborate categories and

develop properties and relationships; and advancing theory development during each step of data collection and analysis (Charmaz, 2014).

The researcher employed a two-stage coding process of initial and focused coding as described by Charmaz (2014). In stage one the researcher collected data and created codes simultaneously, or within a short period thereafter. The second stage, focused coding, was performed as the data were organized and brought into the NVivo software. Handling the large amount of data and analysis in the ongoing research presents challenges regarding interpretation of different types of data. In an action research study using grounded theory Dick (2003) suggests a deliberate approach to resolving ambiguities and clarifying emerging concepts. First the researcher identifies overlaps between any two data sets and then seeks disconfirming evidence in further data collection. Next, the researcher seeks explanation when the two data sets overlap but disagree. In this study ambiguities and questions identified during the analysis process, such as inaudible statements on audio recordings, were member checked with the teachers and discarded if unresolved. A conceptual model of action research with grounded theory, based on Yoong (1996), helps explain the relationship between the cycles of action research and the development of grounded theory (See Figure 5).

***Figure 5*. Conceptual diagram of Action Research with Grounded Theory**

Adapted from: Yoong (1996).
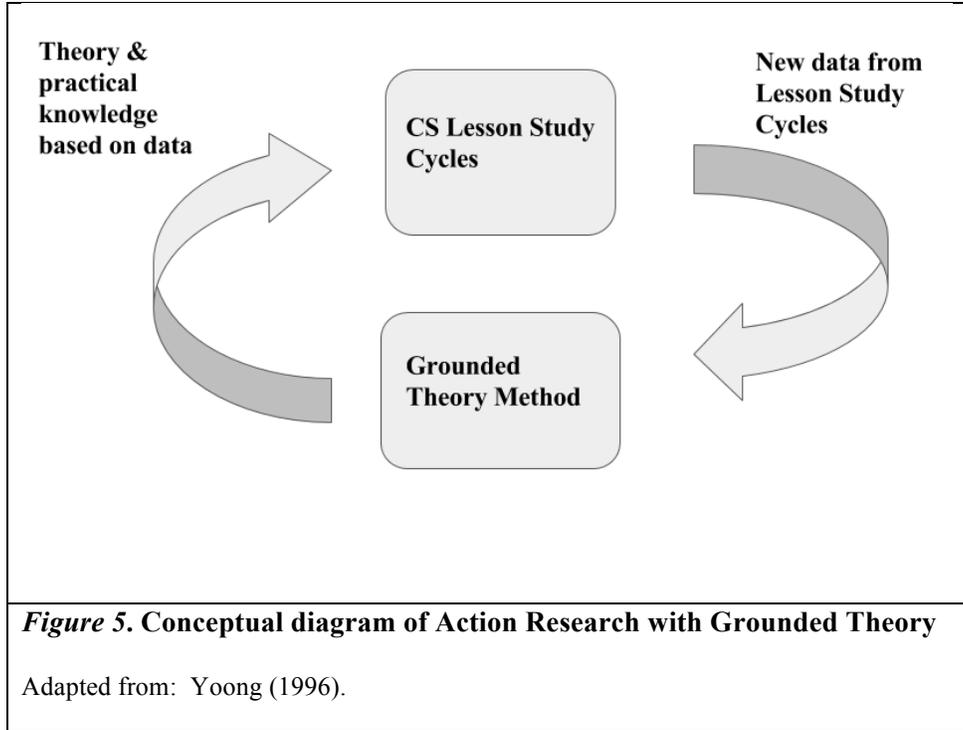
CHAPTER 2

Review of Literature

Introduction

Since the purpose of this research is to examine the effectiveness of *Computer Science Lesson Study*, a CS PD approach that combines CS PD for teachers with the practice of lesson study, this chapter will look at the literature to help answer questions such as (a) how do teachers develop content and pedagogical knowledge? (b) what does the research say about different approaches to teacher PD? (c) how do education reforms create the need for teacher PD? (d) what collaborative approaches do school districts use to provide PD for teachers? (e) how does the practice of lesson study compare with other forms of teacher PD? A review of litereature on international CS education reform will highlight some of the key ideas that underlie effective CS initiatitves. The chapter will conclude with a summary of motives behind the use of technology in schools and a review of essential concerns for CS education reform.

Content & Pedagogical Knowledge

Educational theorist Lee Shulman (1987) complained that research on teacher training focused primarily on management of the classroom rather than how ideas are presented and managed in it. At the time Shulman (1986) had been working on research projects that examined the process that neophyte teachers followed that took them from a state of expertise as learners through a novitiate as educators.

Shulman (1986) began to make a connection between his observations and Piaget's studies on knowledge and growth in the very young. By observing classroom practice and comparing how experienced and novice teachers taught the same

27

material, such as quadratic equations and photosynthesis, Shulman (1986) began to learn how particular kinds of content knowledge (CK) and pedagogical strategies (PK) necessarily interacted within the minds of the teachers (PCK).

Because teaching knowledge depends on a complex web of factors, such as prior knowledge and experience, Shulman's concept of teacher evaluation extends to the entire educational community. Shulman (1986) compares the process of education with that of the study of law, in which case knowledge and precedent is an essential feature of legal training. According to Shulman (1986) legal educators train novice lawyers through the study of instructional cases that exemplify knowledge of accurate, detailed descriptions of events. Comparing instructional examples used by teachers with those used within law instructors suggests that high-quality lesson materials are essential to effective teaching. Shulman (1986) emphasized comprehension, reasoning, transformation and reflection as fundamentals to improve teacher content and pedagogical knowledge and student achievement.

Shulman's work was important for several reasons. First, he identified a trend in the educational system that discounted teacher content knowledge and focused teacher evaluation on pedagogical accomplishment alone. Secondly, Shulman recognized that effective teaching arose from the intersection of content and pedagogical knowledge (PCK). Referring to a "missing paradigm," Shulman (1986) viewed current trends favoring pedagogy over content as conflicting with centuries-old tradition. Shulman (1986) failed to see the humor in George Bernard Shaw's maxim "He who can, does. He who cannot, teaches" (Shaw, 1903). Concerned that such a demeaning image of the teaching profession might underlie current policy,

28

Shulman (1986) studied teacher licensing tests kept by state school superintendents in the 1800's to understand how teaching policy had changed. Asking whether Shaw's comment should be treated as the last word on what teachers know or can do, Shulman (1986) said:

> We have thus seen that the sharp distinction between knowledge and pedagogy does not represent a tradition dating back centuries, but rather a more recent development. Moreover, identification of teaching competence with pedagogy alone was not even commonplace during Shaw's time. A century ago the defining characteristic of pedagogical accomplishment was knowledge of content. The pendulum has now swung, both in research and in policy circles. The missing paradigm refers to a blind spot with respect to content that now characterizes most research on teaching and, as a consequence, most of our state-level programs of teacher evaluation and teacher certification. (p.7)

Through his research on education policy and research with teachers, Shulman (1986) developed a theory, *Pedagogical Content Knowledge* – PCK, that teachers' knowledge consists of a complex relationship between content and pedagogical knowledge and teaching experience.

Teacher PD Activities

Education policy initiatives aimed at creating high standards of student achievement have generated a large body of literature suggesting that high-quality teacher PD is essential to successful reform (Covay, Desimone, Lee, & Hochberg, 2016: Darling-Hammond, 2010; Lee, Duncan, Yoon, Scarloss, & Shapley, 2007). However, the type, scope, depth, duration and cost of teacher PD, and the impact of

29

PD on student achievement, are topics of ongoing discussion. For example, researchers surveyed a national probability sample of 1,027 mathematics and science teachers to provide a large-scale comparison of effects of different characteristics of professional development on teachers' learning (Garet, Porter, Desimone, Birman, & Yoon, 2001). The researchers began designing their study by examining the relationship between PD features identified in the literature and self-reported changes in teacher content knowledge (CK), pedagogical knowledge (PK), and classroom teaching practices. After integrating the data with ideas on "best practices" from the literature, the researchers created a set of scales describing the characteristics of PD activities aligned with the Eisenhower program, a grant system to fund teacher PD for math and science. The researchers sent their survey to a nationally representative sample of Eisenhower PD attendees. The survey asked each teacher to provide detailed information about the specific Eisenhower-assisted professional development activity that the researchers had drawn, in the sampling process (that led to the teacher's being selected for the study).

To help organize the data the researchers identified three structural features that set the context in which a PD activity took place: activity type, duration and collective participation. The activity type (reform vs. traditional) differentiates between activities such as lesson study (reform) and single-session workshop (traditional). Receiving a 72% response, the researchers created a profile that organized PD characteristics into either "structural" or "core" categories. The researchers created core categories by which to assess the quality of the PD. Survey respondents indicated whether the PD emphasized content, active learning, coherence,

and teacher outcomes. Garet et al. (2001) identify four key aspects that are the focus of most teacher PD activities: content, teacher practice, student learning, and the ways in which students learn. Garet et al. (2001) described the four key aspects of teacher PD as follows:

Focusing on content: Although there is a large body of literature on professional development, surprisingly little attention has been given to what teachers actually learn in professional development activities, that is, their content. In particular, little research has been conducted on the relative efficacy of professional development activities that focus on different types of knowledge, skills and teaching practice.

Focusing on teaching practice: Garet et al. (2001) found that some activities are intended to improve teachers' knowledge of subject-matter content; others are designed to improve general pedagogy or teaching practice, such as lesson planning or classroom management; and some are intended to improve pedagogical content knowledge (PCK). PCK is a term that Shulman (1986) used to describe a teacher's skill in selecting the most useful representation of ideas and the most powerful explanations, analogies and examples to make the subject comprehensible to others.

Focusing on student learning: Garet et al. (2001) found that PD activities vary in the emphasis placed on goals for student learning. For example, some PD activities emphasize memorization and mastering procedural skills, while others focus on students' conceptual understanding, such as the ability to explain the reasons behind an analytic strategy.

31

Focusing on the ways students learn: According to Garet et al. (2001), the emphasis given to the ways students learn subject matter varies between PD activities. For example, PD activities may give considerable emphasis to helping teachers understand how children learn by focusing on common perceptions, misconceptions, and solution strategies in a specific subject domain. Garet et al. (2001) say that studies suggest that focusing on subject matter content and the ways students learn is an especially important element in changing teaching practice.

Several significant results of their study were reported. Firstly, the PD activity type influences duration, with reform activities taking longer and having slightly more positive outcomes, regarding CK and PK, than traditional activities. Secondly, the time span and length substantially affects the PD experience and opportunity for active learning. Thirdly, enhanced knowledge skills have a positive influence on the change in teaching practices. Finally, coherence of PD activities improves teaching practice.

Garet et al. (2001) also found that teachers reported that many PD activities, including traditional and reform, do not have features of high quality. Results suggest two factors differentiate high-quality from less productive PD activities: (1) the time allocated to them and; (2) the amount of planning that was undertaken to ensure that the PD activity aligned with the goals and objectives of the school and district.

Regarding the duration of PD activities, Garet et al. (2001) found that teacher PD activities featuring sustained and intensive professional development are more likely to enhance knowledge and skills, as reported by teachers, than short duration PD

32

activities. Results also suggest that PD activities focused on the academic subject matter (content), provide opportunities for "hands-on" work (active learning), are integrated into the daily life of the school (coherence), and are more likely to produce enhanced knowledge and skills. Finally, Garet et al. (2001) estimate that a high-quality activity costs twice as much as an activity that is not high quality.

A literature review conducted by the Education Commission for the States (ECS) was undertaken to identify the factors that determined what types of teacher PD were more effective than others and the extent to which CK contributes to teacher effectiveness (Allen, 2003). According to Allen (2003) the ECS launched a comprehensive literature review because there was no clear definition of "effective teaching", despite a consensus that adequate knowledge was essential to it. Results of the ECS literature review indicated that teacher content knowledge (CK) has a moderate impact on effectiveness. However,the results are limited since most of the 92 studies reviewed focused on teaching mathematics. It is possible that the moderate impact of the PD was due to prior knowledge of mathematics among the participants that made the content provided by the PD seem less valueable. Another possibility is that the PD did not successfully convey the content knowledge in ways that teachers could adapt it to the context of their classes. Further, the ECS analysis of the literature does not provide a sufficient level of detail to show CK impact on teacher effectiveness across different grade-level categories (Allen, 2003).

33

Education Reform

The literature shows that successful education reform depends on classroom teachers, not policymakers, as the leaders who will develop the content and pedagogical skills required to teach the material effectively (Wynne, 2001). In the decentralized US educational system, local educators work out the implementation details for curriculum reforms. For example, the Common Core State Standards (CCSS), a large-scale reform initiative launched in 2009, presented the US educational system with a curriculum-wide set of standards for student achievement. CCSS sought to equip students with skills that they would need for success in the 21st-century. The CCSS program website describes the project as follows:

> The Common Core outlines a set of high-quality academic standards in mathematics and English language arts/literacy (ELA). These learning goals describe what a student should know and accomplish at the end of each grade. The standards are intended to ensure that all students have the skills and knowledge necessary to succeed in college and in life, regardless of where they live (CCSS, 2016).

As a nationwide reform initiative, CCSS generated much publicity. By 2013 all but four states had developed plans to implement CCSS by 2015 (Gewertz, 2015). Studies show the implementation planning for CCSS established a consensus among experts that successful implementation of the reforms would require high-quality teacher PD on a scale that presented an unprecedented challenge to the US educational system (Jenkins, & Agamba, 2013; Marrongelle, Sztajn, & Smith, 2013).

34

The substantial investment in teacher PD needed to implement CCSS at scale is evident in the following recommendations developed by a working group of educational leaders and experts who joined forces to tackle the "massive professional development task ahead of the nation," (Marrongelle et al., 2013, p.203). Marrongelle et al. developed five recommendations. Firstly, states must use experts to ensure that teachers learn CCSS. Secondly, states need to target a variety of role groups and attend to the PD requirements of each group. Thirdly, local educators and policy makers must work to inform the general public on the impact of CCSS. Fourthly, educational leaders must implement ongoing assessment and evaluation of CCSS. Lastly, consortia are needed to oversee and improve the role PD plays in successful implementation of the CCSS (Marrongelle et al., 2013).

Despite the initial excitement provoked by CCSS, and extraordinary collaboration and effort across all levels of government and education, the scope of the project was beyond the resources of many states. By 2015 several states had reversed their adoption of the standards, and nearly half had backed out of their initial promises to use tests designed to measure mastery of them (Gewertz, 2015).

The gap between instructional practice and educational research, presented through curricular changes, often leaves teachers with a perception that the research-based curriculum is not relevant to their classroom (NCTE, 2010). Ongoing discussion of top-down educational initiatives that rely on standardized tests to assess student achievement, and hold teachers accountable, highlight both the central role that teachers play and the incentive to find practical ways to meet their needs.

35

Because most schools today implement an array of curriculum reforms, the complex realities of the classroom make it impossible to isolate the effects of a single program (Fullan, 1992; Guskey, 1997; Guskey & Sparks, 1996). Fullan (1992) suggests that in an environment of reform, school principals should strive to make vision-building a joint exercise that supports and recognizes teachers as instructional leaders.

A report by the Institute of Education Sciences (IES) looked at the impact of teacher PD programs on student achievement (Lee et al., 2007). Researchers reviewed 1300 studies that sought to determine the effect of teacher PD on student achievement in three content areas, mathematics, science, and English/language arts. Finding only 9 (of 1300) studies that met What Works Clearinghouse standards, the researchers concluded that the lack of consistency and high variability in time and intensity of the studies made it difficult to discern any pattern of effects on student achievement among the nine studies. However, Lee et al. (2007) found that PD activities that directly involved the teacher, and were of a duration of 14 hours or more, were likely to have a significant impact on student achievement.

Collaborative PD Activities

Attempts to provide high-quality teacher PD with the limited resources available to most school districts has led to a proliferation of "reform" style teacher PD activities that involve collaboration among educators. Professional Learning Communities (PLCs) have emerged as the most well-known and widely embraced approaches to collaborative teacher PD (Dufour, 2011; Talbert, 2010; Vescio, Ross, & Adams, 2008). PLCs involve groups of teachers who regularly meet to discuss topics

36

such as curriculum change, lesson materials and learning objectives. While PLC's often consist of groups who work in the same school and meet in person, the literature shows a trend toward online collaboration among PLC's using video conferencing software.

The proliferation of PLC's has emerged in response to the increased pressure on schools to find ways to involve teachers and administrators in reform efforts (Hargreaves, 2007; Talbert, 2010). Student achievement improves when teachers participate in PLC's that are highly involved in the school community and focused on student achievement data (Vescio et al., 2008). There is also research showing that PLC's require appropriate planning, support and implementation resources, such as allocation of time for teachers to meet (Talbert, 2010).

Per Talbert (2010), enthusiastic efforts to implement PLCs on a large scale often backfire because many teachers participate only to comply with the mandate, rather than fully engage and focus on student learning. Talbert(2010) also notes that PLCs do not succeed when school administrators either fail to understand the underlying principles or create PLCs in ways that alienate teachers. For example, PLCs that school administrators implement to achieve near term gains in student test scores, emphasize individual teacher quality or undermine principles of collective responsibility may face pusback from teachers (Talbert, 2010).

In a review of current studies on the effectiveness of PLCs Vescio et al. (2008) report that there have been few rigorous evaluations of PLCs' contribution to effective instructional practices. Nonetheless, school administrators and policymakers have strongly supported expanding PLCs (Talbert, 2010).

37

A report by the Institute of Educational Sciences (Blitz, 2013) examining results of online and hybrid PLCs found that nearly two-thirds of the reports on online PLCs involve K–12 institutions' using online technology to extend the scope of traditional PLCs, where participants meet in person. The report found an increasing interest in regional and national PLCs in countries that have education systems over vast geographically dispersed and rural areas. While the popularity of PLCs has increased, little prior research shows exactly how the dynamic process within PLCs helps teachers build content and pedagogical skills.

In a recent study Popp and Goldman (2016) sought to understand how discourse within PLCs helps teachers develop new knowledge. The study involved PLCs' participating in a larger university project designed to improve literacy and student achievement. Noting that most research studies included science and mathematics, and fewer literacy and language or arts teaching, Popp and Goldman focused on PLCs comprised of grade-bands of pre-kindergarten through sixth-grade. The study examined the discourse within grade-level bands of PLCs of elementary school teachers to ascertain if and how the intended focus of the PLC meeting tasks supports the composite construction of pedagogical content knowledge (Popp & Goldman, 2016).

Popp and Goldman (2016) created three grade-level band PLCs for the study: a primary grade-level PLC consisted of teachers in Pre-K to 2nd grade; an intermediate grade-level PLC for the 3rd and 4th grade, and an upper elementary PLC consisted of 5th and 6th grade-level teachers. Examining the discourse data generated in 92 meetings of the three PLCs, the researchers found marked differences in the frequency

38

of topics of discussion and dialogue regarding materials and student achievement. Results of the study suggested that an increase in knowledge building occurred when PLCs were focused on discussion of assessment rather than on instructional activities they had implemented. Popp and Goldman (2016) make an important point regarding the impact of PLC's on student learning: knowledge building was less when the discourse among PLCs focused on teacher action, such as instructional activities, rather than on the assessment of instruction on student thinking.

Lesson Study

Lesson study is a form of professional development which originated in Japan and is widely practiced in many countries, including the United States (Lewis & Hurd, 2011). As a kind of action research, lesson study follows a four-step cycle of planning, action, reflection and evaluation. Lesson study came to the attention of researchers Stigler and Hiebert (1999) through their involvement in a project for the Third International Mathematics and Science Study (TIMMS). After collecting and analyzing videotaped eighth-grade mathematics lessons from three countries (the United States, Germany and Japan), Stigler and Hiebert concluded that other nations were continually improving their teaching approaches, while the United States did not have a system for improving teaching (1999).

Stigler and Hiebert (1999) pointed to Japanese lesson study as the ablest and most purposeful approach to teacher PD that they had encountered. The difference, they found, was that lessons taught by Japanese teachers showed coherence with a clear beginning, middle and end. The entire lesson was planned as a sequence of events that fit together to reach a conclusion, like a well-formed story.

39

In contrast with Japan, Stigler and Hiebert (1999) say the approach to curriculum reform in the United States follows as an incoherent route in which policymakers adopt a program and wait for student achievement scores to rise. When scores do not rise in the short term policymakers begin hearing complaints that there is something wrong with the policy. As the momentum for the policy starts to reverse, experts meet to develop a recommendation for a new policy, often in the opposite direction.The whole process goes on without undergoing a process of inquiry on whether or not the original program was ever implemented in classrooms, or if implemented, how effective it was in promoting student learning (Stigler & Hiebert, 1999).

Hiebert and Morris (2012) found that teacher PD in the US is focused on instructional products rather than improving teaching in the classroom through a process of lesson study. According to Hiebert and Morris (2012), improving teaching and student achievement will happen only when the US is ready to embrace cultural changes that focus teacher PD on solving problems of the classroom rather than on enduring characteristics of teachers.

Lesson study is often used by teachers to improve academic outcomes in subjects which are difficult to teach, e.g proportional reasoning (Lewis, 2011). Lesson study follows a four-step process that begins when a group of teachers identifies curricular goals in a content area and then starts working together to plan a lesson (Saito & Atencio, 2013). Lesson study typically involves several cycles of planning, teaching and reflection before the lesson is deemed complete. This achieved, the

40

materials are collected and made available for use in other classrooms. Over time the lesson-study materials and records of discussion become part of a growing repertoire of lesson materials that have undergone review in the classroom. This helps teachers develop shared ownership of detailed lesson plans that provide the school with a consistency of teaching practice from year to year and from teacher to teacher (Hiebert and Morris, 2012).

Through observation and participation in hundreds of lesson study projects across many Asian countries, Saito and Atencio (2013) identified key social relationships and interactions that hold between teachers and students, faculty and their peers, and teachers and administrative personnel. They note that, while emerging findings explicate how lesson study enables teachers criticallyto examine their practice, more research is needed to understand the complex social interactions that occur during the lesson study process. Individuals negotiate power and construct identities in social interactions of daily life. In lesson study teachers develop and use power, underpinned by discourse, to govern individuals' daily practice (Saito & Atencio, 2013).

In the US educational system reformers often focus on extrinsic rewards, such as monetary bonuses for student achievement, rather than on intrinsic motives such as the satisfaction of seeing students learn (Lewis, Perry, Friedkin, & Roth, 2012). According to Edwards (2014), as a semi-formal four-step process; lesson study has the facility to slow down the complex process of teaching and learning, thus enabling teachers to improve their classroom practice and pedagogical and subject knowledge in ways that enhance the quality of their pupils' learning. Lesson study can provide

41

advantages in school systems with a high turnover of teachers. Because teachers participating in lesson study groups are encouraged to share in discussion and participate in teaching and learning, as equal partners, it is reasonable to expect that lesson study provides novice teachers with a range of knowledge, learned through collaboration with peers, beyond the education focus of the cycle (Lewis et al., 2012).

Collaborative teacher PD approaches can improve teaching practice and raise student achievement (Ermeling, 2010; Garet et al., 2001; National Research Council, 2001). However, Lewis et al. (2012) say that success in lesson study depends on a variety of intrinsic and extrinsic factors relevant to the individual teachers and the school. To increase the potential of lesson study to achieve learning objectives, Lewis et al. (2012) call for more research focused on high-quality instructional materials, practice-based instruction and organizational structures that ensure collaboration among colleagues.

In the rapidly changing world of technology computing teachers in high schools often view PD activities through a constructivist and social constructivist lens (Kordaki, 2013). Constructivist teacher PD empowers teachers as learners who must first construct understandings of intrinsic processes while they collaborate with peers to invent, discuss and reflect on lesson topics (Cannella, & Reiff, 1994). Japanese educators have used lesson study for decades as a constructivist approach to systematically refine the process that teachers follow to improve the teaching of mathematics and science (Lewis, Perry, & Murata, 2006; Perry & Lewis, 2009). Some researchers suggest that action research and lesson study have become a focal point where Eastern and Western thought converge through epistemological commonalities

42

which share both an aversion to rigid standards and a constructivist view that the individual is profoundly involved with others in society (Somekh & Zeichner, 2009; Sim, 2009).

Motive for Technology Reform

The proliferation of desktop computers in the mid to late 1980's initiated education reforms focused on bringing technology into classrooms. The purpose of these reforms was to align the US educational system with the needs of a booming technology industry. Cuban (2001) describes the forces behind technology-driven education reforms of the late 1980s and early 1990s as a loosely organized coalition of public officials, corporate executives, vendors, policymakers and parents. According to Cuban, these early technology reform advocates saw the need for computers in schools to serve three goals: (1) to make schools more efficient and productive than they currently are; (2) transform teaching and learning into an engaging and active process connected to real life, and; (3) prepare the current generation of young people for the future workplace.

As a professor emeritus at Stanford University, Cuban decided to examine the use of technology in schools in Silicon Valley to find out if the technology-driven education reforms were meeting objectives. In his report Cuban acknowledged the need for educators to come to terms with technology as an educational tool. However, in studying the use of technology in Silicon Valley Cuban found that over a decade of investment in information technologies schools had not achieved the transformation in teaching and learning nor the productivity gains sought by the reform coalition of

43

corporate executives, public officials, parents, academics and educators. Cuban (2001) urged district administrators to ask tough questions regarding the impact of funding diverted to technology reform. Cuban (2001) suggests that technology reform advocates often fail to balance technology spending with other critical needs, such as smaller classes, higher entry-level teacher salaries and renovation of decayed buildings.

21ST Century CS Reform

The increasing publicity generated by the rising number of unfilled technology jobs, and popularity of the "Hour of Code" has created significant momentum for CS reform. Recent CS reform initiatives such as CS4RI (2016) have generated much discussion of the need for PD to increase the number of teachers qualified to teach CS. That the US system of education is unprepared to cope with a large CS initiative, on top of ongoing reforms such as CCSS, is evident in the results of a Gallup poll sponsored by Google (2015). Results show that 9 out of 10 parents consider CS to be an important 21st-century skill, on a par with other STEM subjects. On the other hand, the report shows that most principals and school district superintendents do not consider CS a top priority. The results of this poll suggest that, while policymakers and parents are pushing for CS reform, school administrators are working from a priority list that does not include CS.

The shortage of college graduates ready to fill the millions of technology-related jobs is well documented (Code.Org, 2016; CSTA, 2011; EEOC, 2016; Google, 2015). While industry leaders are demanding that education supply an ever-increasing number of skilled workers, many students are not interested in CS. According to

44

Albinson (2013) the way schools introduce computing is a critical reason why students lack interest in CS. Computing education in schools typically focuses on how to use computers while failing to provide students with a full understanding of computing and its value towards enhancing students' problem solving skills (Morreale & Joiner, 2011). With CS currently a low priority in most US schools and a lack of understanding regarding the nature of CS, implementing CS curriculum reform without empirical review may perpetuate the negative perceptions of CS among students.

A study conducted by Taub, Armoni and Ben-Ari (2012) examined interest in CS among students who had participated in a set of 24 activities called Computer Science Unplugged (CS Unplugged). CS Unplugged provides an entertaining and challenging way to engage students in CS without a computer. The researchers found that, while CS Unplugged activities did increase students' understanding of CS, they became less interested and less attracted to CS as a result of participating in the activities.

On the other hand, there are those who argue that providing substantial programming experience to students helps them develop essential skills. Seymour Papert (1998) said that young programmers liked to work hard as long as what they were doing was interesting. Papert (1998) said, "The preoccupation in America with making things easy is self-defeating and cause for serious worry about the deterioration of the learning environment" (p.88). Computer programming is an essential part of CS; like reading and writing, computer programming is a skill that students are capable of learning and understanding. CS curriculum innovations that do

45

not include computer programming may provide an enjoyable activity but may not help students develop fundamental CS or computer programming skills. Taub, Armoni and Ben-Ari (2012) suggest that "unplugged" activities intended to teach CS without computer programming should include explicit instructions that show students how problem-solving and analysis in the activity relates to the solution and the role of the computer in finding the solution.

Computer programming is a discipline that requires patience and perseverance on the part of the student, much like learning to play a musical instrument or solve a mathematical problem. Papert (1998) viewed CS education through a constructivist lens where knowledge is constructed through classroom activities in which the teacher becomes a mediator who helps students learn to balance fun and frustration. Papert (1998) viewed the "hard fun" of problem-solving as an essential part of living a successful life.

Because computers are machines designed and built by people they have limitations and restrictions that must be understood. Systems developers reconcile conceptual design with the capabilities of technology before releasing new software. Computers share a common architecture that emerged with the invention of the stored-program computer in the late 1940s (Patterson & Hennessy, 2013). As is the case with many different types of machine (e.g. cars, bicycles, airplanes), different types of computers share similarities. Firstly, each computer has a finite set of instructions. Secondly, they have a memory device capable of storing programs and data as binary numbers. Thirdly, they have a control system to coordinate the internal operations of

46

the computer. Fourthly, a set of internal memory locations is used to hold data during the execution of a single instruction.

Because computers share architecture, teaching computer programming fundamentals, rather than specific technologies, helps students develop skills that apply to any programming language or hardware platform. Because computers execute one instruction at a time, computer programming requires students to learn a step-by-step analytical approach to problem-solving. Solving problems in this way can become a useful and rewarding skill beyond computer programming.

Problem-Solving Skills

Before there were many computers Polya (1945) wrote "How to Solve It", a classic work on problem-solving designed to improve the effectiveness of teaching math and science. In addition to offering practical advice on how best to help students learn problem-solving, he describes why problem-solving is fun, rewarding, and has lifelong effects on student achievement:

> A great discovery solves a great problem, but there is a grain of discovery in the solution of any problem. Your problem may be modest; but if it challenges your curiosity and brings into play your inventive faculties, and if you solve it by your means, you may experience the tension and enjoy the triumph of discovery. Such experiences at a susceptible age may create a taste for mental work and leave an imprint on the mind and character for a lifetime. (from the preface, p.v)

In 2014 computing researchers Armoni and Gal-Ezer argued that the goal of teaching CS in elementary schools is to expose students to the foundations of CS, not

47

to create a supply of workers for the technology sector. Barr and Stephenson (2011) observe that, because computing is an essential part of life, it is no longer sufficient to wait until students are in college to begin teaching algorithmic problem-solving and computational methods.

Claiming that students' scores in mathematics can be influenced by CS, Felleisen and Krishnamurthy (2009), proposed imaginative programming as a CS curriculum that aligns programming with algebra to "bring mathematics to life" (p.38). Armoni (2013) argued that, unlike most scientific curricula, CS in K-12 needed a stable foundation based on constructivist concepts that promote hands-on activities that help students develop their ability to solve problems through inquiry.

Teaching CS in elementary schools goes well beyond career development. Yongpradit (2014) described the computer as our era's best and most accessible tool for engaging young students with powerful ideas. By coding puzzles and participating in unplugged activities, students become passionate about expressing themselves and using their imagination (Yongpradit, 2014).

With many articles discussing CS in schools, there are an emerging number of empirical studies that are beginning to clarify the issues and concerns involved in launching CS initiatives. For example, a study examining an ongoing project designed to increase participation in CS through a computer game development suggests that CS can successfully be implemented in elementary schools.

Aimed at providing foundational skills in computational thinking (CT) in middle schools, Repenning et al. (2015) developed *Scalable Game Design* as an engaging way to introduce CS and CT to teachers and students. The Scalable Game

Design software includes a tool for measuring student learning skills by analyzing the interactions of objects within a game and associating those interactions with a computational thinking pattern.

The idea originated through insight that while there are many programming languages, game designs usually involve observable behavior between the objects on the screen. Repenning et al. developed an algorithm to identify computational thinking patterns by analyzing the behavior of game objects:

> Are they static? Are they moving around? If they are moving around, are they moving in some straight direction or are they wandering around randomly? If there is more than just one object, how are they, if at all, interacting with each other? Are objects colliding, pulling each other, or tracking each other? (p.21)

Repenning et al. analyzed thousands of games, created by students, to identify basic and advanced computational thinking skills that equate to algorithms, such as Hill Climbing, a search algorithm that looks at neighboring values and moves toward the larger. Scalable Game Design is popular because it provides a fun, yet challenging experience that helps students and teachers develop foundational skills.

As one of the few large-scale CS initiatives to grow beyond the initial pilot phase, Scalable Game Design is accessible across school settings, promotes a deeper understanding of computational thinking among students and teachers, and can be used by students as young as 7. Repenning et al. describe the results of their study as highly motivational for students and potentially valuable in engaging more women and underrepresented minority students in CS.

49

Educational Systems

Increasing access to CS in K-12 is just one part of the problem. There is also the evolving world of technology, and cultural issues in the technology sector, that need to be addressed. One area that clearly needs more attention is the nature of the relationship between the technology industry and the educational systems of the US. Over the past decade the decentralized educational system in the US has been criticized as a barrier that prevents CS from reaching all students. While that may seem obvious to some, it is also reasonable to assume that if CS were made a top priority, the US educational system would respond, as it did with the recent CCSS. Because CS must compete for a place on the curriculum with more established disciplines, it may be beneficial to find ways to work within the current educational system, as other countries are doing. For example, from an international perspective, Tenenberg and McCartney (2014) say:

Because each country has its own traditions, history, languages, cultures, economic system, and institutionalized forms of education, it would be absurd to think that "one size fits all," that there is a fixed set of "best practices" for teaching computing in schools that is best across such a diversity of contexts. Rather, as we see here, the forms of education adopted in each country are adapted to that country's traditions, history, languages, etc. Not only does this diversity represent the state of computing education sampled across many parts of the globe, but it provides a great opportunity for learning. (p. 6:2)

50

CS PD Activities

International studies on CS in education show that strong PD for teachers is the key to successful CS reform efforts, yet many reform initiatives do not allocate adequate resources to support PD for teachers (Hubweiser, et al., 2014). There are also studies that show how CS reforms that depend on outside funding often fail when funds are not available, for teacher PD, within the local education system. One of the largest and most ambitious CS reform initiatives was a six-year project funded by the NSF and led by computing researchers Guzdial, Ericson, Mcklin and Engelman (2014). The project called Georgia Computes! (GaComputes) aimed to improve computing education across the state of Georgia while engaging members of under-represented groups including women, African Americans and Hispanics. GaComputes' interventions included a broad range of activities for students and intensive PD for teachers. Despite its initial success, the project proved vulnerable to institutional barriers. In a blog post on April 15, 2014 researcher Guzdial (2014) explained what happened:

Here in Georgia, we were one of the first states to use the CSTA Model K–12 Curriculum to design a computer science set of courses in high school. The initial course was "Computing in the Modern World". However, soon after adoption (2007), the professional development budget was cut dramatically. Too few teachers learned to teach the new courses. As the curriculum were [*sic*] revised, the learning objectives were lowered. Most of the CS content was removed. The new (2013) initial course is "Introduction to Information Technology", and learning objectives now include the skills necessary to run a

51

customer support call center ("Determine the best method to maintain a customer list and communication platform"). (Guzdial, 2014, p.8)

Teaching CS is arguably more challenging than teaching other subjects. In addition to knowing how to explain what computer science is and why students need to learn how to code, there are hardware and software issues that must be handled prior to and during the lesson. The complexities of teaching CS in college led to the development of a collaborative PD approach called the *Disciplinary Commons* (Fincher & Tenenberg, 2007). The Disciplinary Commons project was developed to address the need to create a professional learning community among CS teachers based on two beliefs: firstly, that CS teachers need to acquire and maintain teaching skills that focus on the discipline of CS rather than specific skills, such as a particular computer language; secondly, that teaching CS is treated as a reflective practice that benefits through collaboration and sharing of ideas and techniques among practicioners. In 2005/2006 a Disciplinary Commons project involving two cohorts of CS teachers from the US and UK met monthly to document and share knowledge about CS teaching and student learning, and to establish best practice for the scholarship of teaching CS by making peer-reviewed documentation available for future use by other educators (Fincher & Tenenberg, 2007). Similar to the present research which used lesson study as a framework for collaborative lesson development, participants in the Disciplinary Commons project focused on an introductory CS lesson (for college students). During monthly meetings the participants focused on a single aspect of the course portfolio submitted by the two

cohorts. Between meetings the Disciplinary Commons involved the CS teachers in lesson observation, selected readings and peer-review of the course portfolio. Results of the project showed that participants developed deeper insights that prompted them to make specific improvements in their practice.

The power of collaboration is evident in the reports from several participants who reported that working with peers led to improvements that enhanced the effectiveness of the CS course. Because the study also involved teaching and re-teaching a single lesson, Fincher and Tenenberg (2007), found the process of reification within the Disciplinary Commons especially powerful because it focused on a single discipline rather than multiple subjects within a curriculum portfolio.

The value of collaboration and development of high-quality lesson plans and materials is a fundamental concept of lesson study and the present study. Because the focus on a single topic by multiple teachers requires time, support from the school principal and school district is essential. Because CS is a complex discipline, time for teachers to learn CS is the most valuable resource that can be made available and is the most important factor identified as critical to success (Hubweiser et al., 2014).

Providing the appropriate teacher PD necessary to help students succeed is a problem not only for CS but across all subject areas. Regarding student achievement and graduation rates, inadequate or ineffective teacher PD is considered a major factor in the continuing failure of schools in the United States to maintain parity with other countries (Darling-Hammond, 2010). Teacher preparation matters because it can enhance teacher effectiveness and increase the likelihood that teachers will stay in the profession long enough to become thoroughly experienced as educators (Darling-

Hammond, 2010). In 2015 Hubweiser, Armoni and Giannakos published "How to implement rigorous computer science education in k-12 schools? Some answers and many questions ". The report followed up on previous work comparing relevant aspects of particular implementations of Computer Science Education (CSE) in K-12 schools in different countries.

Because there are many unanswered questions, Hubweiser et al. offer a review of important articles and seek to encourage further discussion by providing a set of open research questions (Table 1).

Table 1: **The Big Questions of computer science education**

1. At what age should CS start? Which content, learning objectives, methods, and media are suitable to learn CSE concepts in primary schools?
2. Does it pay off to give CSE teaching time in primary schools, taking this time away from other important learning fields?
3. What is CSE good for? Which superordinate competencies that are regarded as necessary and valuable by the majority of the society are supported by CSE? And which parts of CSE do really support these competencies, and in which respect?
4. How and when should programming be learned in K-12 schools?
5. Which contributions to general education could be provided by learning to program?
6. Which programming languages and which didactical approaches are the most suitable for the different age groups and school contexts?

In their report of 2014, "Perspectives and Visions of Computer Science Education in Primary and Secondary Schools", Hubweiser et al. developed the *Darmstadt Model*, a framework of categories and themes for analyzing CS education across many countries and educational systems. In addition to presenting an

54

organizing framework, Hubweiser et al. (2014) found three topics that were common to CSE, regardless of the organizational structure of a local education system (Table 2).

Table 2*: Common topics in CS education reform*

1. Proper teacher education in substantial extent and depth seems to be one of the most critical factors for the success of rigorous computer science education on the one hand and one of the hardest goals to achieve on the other.
2. There is a convergence towards computational thinking as a core idea of the K-12 curricula.
3. Programming in one form or another seems to be absolutely necessary for a future oriented CSE.

Determining exactly what CS content should be taught to students is a critical decision facing designers of CS reform programs. Futschek (2006) suggests that an understanding of computational thinking is the key to understanding computer programming. As a CS faculty member at the Vienna University of Technology, Futschek and colleagues observed that students entering the CS program did not have the skills and pre-knowledge necessary to start a university study in Computer Science, resulting in high dropout rates during the first year.

Futsheck (2006) identified five factors that cause an alarming dropout rate and low success rates among CS students. Firstly, students lack pre-knowledge of CS. Secondly, students do not know how computers work. Thirdly, students do not know how to develop algorithms. Fourthly, students lack programming experience. Lastly, students do not have a basic understanding of mathematics.

55

Futschek found that students learned when the problems were chosen carefully; solvable independently of the programming language, and when they used tools that allow visualization of algorithms. Futschek's article raises the important concern regarding the need carefully to choose the curriculum and materials involved in CS education. The selection of CS lesson materials and instructional examples for this study was intended to follow Shulman's (1986) suggestion that educators select instructional examples that sharply delineate a concept or theory, similar to an example that might be used in a class on law, per Shulman (1986):

Can we learn from other disciplines or professions such as law or architecture, where analogical reasoning from cases is much more typical, how to conceive of and use case knowledge in education? Why are cases memorable? Is it because they are organized as stories, reflecting the grammar of narrative forms of discourse, that makes them more readily stored, ordered and retrieved than their expository or propositional analogs? (p.8).

CS Curriculum Research

Because there is so little empirical research showing examples of practical, substantive CS education in US schools, materials and methods need to undergo an examination to determine whether they help students learn fundamental concepts, or add to the confusion regarding CS. For example, Fincher (2015) calls for cognitive research asking questions regarding the presentation of ideas and whether graphical programming environments, such as Scratch Jr, offer a productive way to teach computational thinking and computer programming. Without evidence-based research on the effectiveness of different CS curriculum materials, teachers are faced with a

plethora of plausible CS approaches. With no way to distinguish between them, teachers are forced to make a choice that is based primarily on the claims and charisma of curriculum developers (Fincher, 2015).

Regarding the compelling need for more research on the effectiveness of CS curriculum materials that are being deployed without appropriate studies Franklin (2015) explains that the lack of investigation in CS deployments creates the possibility of high-profile failures that may affect the credibility of CS and in turn harm future students' chances of receiving quality CS education. Extensive research by Shulman (1987) and others showed the need to develop "standards without standardization", which requires that we develop an understanding, through empirical studies, of both the sources of content knowledge and complexities of the pedagogical process.

CHAPTER 3

METHOD

Introduction

Qualitative research is both highly creative and technically challenging. For others to judge the quality of findings, researchers must disclose explicit details of their data collection procedures and the process of analysis (Patton, 1990). Equally important, per Creswell (2014), is that researchers explicitly state their philosophical ideas which, though remaining hidden in the final report, influence the conduct of the research. This chapter will provide details of the selection of the research design; techniques used by the researcher to establish credibility and minimize threats to trustworthiness; the participants, setting, and study schedule; the data collection procedures and analysis process. Because the researcher is the instrument in qualitative inquiry, this section will provide a summary of my qualifications to conduct the study and my perspectives on the topic.

Selection of the Research Design

The process of selecting an appropriate research design for this study began in 2013 when I enrolled in the Ph.D. in Education program at Rhode Island College and the University of Rhode Island. My interest then, as now, was to find ways to help schools provide computer science and programming to most students in public schools. When I began to look for a teacher PD approach for CS that would work in the context of US public school systems my perspective was based on previous experience as a CS teacher and application software developer. At that point I

58

assumed that the primary reason why schools did not offer CS was that they simply did not have any teachers who knew what CS was or how to teach it. However, as I began to study the literature on CS education reform, I found that solving the problem involves not only increasing the supply of teachers who understand CS, but also dealing with cultural issues and stereotypes in the technology sector that perpetuate ethnic and gender disparities.

When I began to study this problem in 2013 I found that computing researchers had been working to move CS into the mainstream K-12 curriculum for over a decade. Because CS had never been part of the curriculum in the US a set of CS K-12 standards was needed. In 2011 the publication of the CSTA K-12 Computer Science Standards finally provided a clear definition of CS and seemed to overcome a major barrier preventing CS from being included in the K-12 curriculum. Encouraged by the number and scope of ongoing CS reform initiatives I assumed that it was only a matter of time before the number of large-scale CS reform initiatives such as Georgia Computes! (Guzdial et al., 2014) would bring CS into the mainstream curriculum in US schools. I was also convinced that the success of these CS reform initiatives would begin to close the diversity gap in the technology industry. Given these assumptions, my initial research designs were simply aimed at contributing to the emerging literature on CS reforms focused on students and teachers in high schools, using quantitative and qualitative studies aimed at examining CS workshops and curriculum interventions. Research showing the difficulties encountered by large-scale CS reforms began to suggest that their focus primarily on high school students might not be the best way to solve the problem. Working through initial research designs with

my professors and colleagues not only deepened my understanding of the problem but prompted me to focus on a "grass roots" approach based on action research.

Because CS reforms for high schools reach only a small percentage of K-12 students I decided to develop a PD design for elementary school grade levels that would provide CS to all students. I reasoned that introducing CS and computer programming to all students at elementary school would, over time, begin to increase the number of students moving into technology careers. There is general agreement among computing researchers that computer programming is an essential part of CS education (Hubweiser et al. 2014). Therefore I was interested in creating a PD program that featured computer programming, history of computers and fundamental skills outlined in the recently released *K-12 Computer Science Framework* (CSTA, 2016).

Finding a suitable programming language for this project was a challenge that was solved with the release of the Scratch Jr. programming language in 2014. Prior to Scratch Jr the options for elementary school computer programming were limited. Scratch Jr. provides a "real" programming environment that uses a graphical instruction set, is easy to learn and accessible to all students. The release of Scratch Jr made it possible to provide students and teachers with an introductory course in CS that included the fundamental concepts of computer programming.

Three major factors influence the design of this study. Firstly, the resource constraints and political realities of the US educational system make any education reform extremely difficult to implement and sustain (Gewertz, 2015). Despite many thousands of hours expended by CS advocates, educators, administrators and

60

researchers and millions of dollars in resources expended over many years, computer science education remains on the margins of the K-12 curriculum at US public schools (Google, 2015). The frustration among computing researchers and educators over the continued marginalization of CS in K-12 education is palpable. Harsha (2015) puts it as follows:

> If "fixing" computer science education in kindergarten through grade 12 is so clearly necessary, why has there not been more progress in the U.S.? In an age when the ability to think computationally already is, or certainly will be, a prerequisite for success in so many endeavors, why do we still struggle to reform K–12 computer science and make it more relevant?

The second factor influencing the design of this study emerges from evidence that approaches to CS PD that focus on individual teachers are vulnerable to institutional, situational and attitudinal barriers. For example, a large-scale case study designed to bring CS into the pre-collegiate curriculum in Los Angeles public schools and providing high-school teachers with in-depth CS PD through workshops, seminars and mentoring encountered a variety of implementation difficulties as the study began to scale beyond the pilot phase. Researchers Goode and Margolis (2011) reported:

> The reality is that within the world of public education (again, this is where the majority of students are!), things are constantly changing and/or moving at a different pace than we would like. Think: one step forward, two steps back; find a great teacher, teacher gets moved to another class; find a great school, school has budget crisis and must cancel class; have a great professional

61

development and discover that teachers have no prep periods during the day and lacking this time to prepare and practice this new skill, revert to their previously known ways when they return to the classroom, etc. Since there are no blueprints for this type of innovation, we are deeply engaged in a learning-by-doing process and all that entails. (p.14)

Rather than focus only on teacher workshops I began to look at research that showed that significant instructional changes were more likely when teachers collaborated in teams and used inquiry-focused protocols to secure continuous improvement in instruction (Ermelling, 2010). In addition to the literature promoting PLC's as an effective PD approach, the research also shows that PLCs can take many different forms, leading to confusion over exactly what a PLC entails. Dufour (2004) observed that educators are likely confused about the term "professional learning community" because:

> The term has been used to describe every imaginable combination of individuals with an interest in education. – a grade level teaching team, a school committee, a high school department, an entire school district, a state department of education, a national professional organization, and so on. In fact the term has been used so ubiquitously that it is in danger of losing all meaning. (p.6)

A third factor behind the design of this study arises from studies showing the steep learning curve and difficulties that confront high school students who have little

or no prior CS education (Simon, Fincher, Robins, Baker, Box, Cutts, … Tutty, 2006). At the precollege level CS courses have traditionally been offered as electives and focused on computer literacy or applications, rather than teaching the core concepts that enable them to become innovators and developers of computing systems (Gal-Ezer, & Stephenson, 2014). There is considerable churn in how to teach CS at both the high school and college levels, due to the degree of complexity and likelihood that the languages of today may be obsolete by the time students graduate (Felleisen, & Krishnamurthi, 2009).

In a rush to provide CS to high school students CS courses may compromise scope and depth due to the limited time and resources available. This situation results in students' entering college lacking fundamental knowledge of CS concepts. Texas A&M professor and inventor of C++, Bjarne Stroustrup (2009), suggests that spotty CS knowledge among college undergraduates arises from a spotty knowledge of CS that they develop in precollege years. Arguing for a more substantive approach to CS education across the K-12 spectrum, Stroustrup (2009) describes the situation among college students as follows:

For many, "programming" has become a strange combination of unprincipled hacking and invoking other people's libraries (with only the vaguest idea of what's going on). The notions of "maintenance" and "code quality" are at best purely academic. Consequently, many in industry despair over the difficulty of finding graduates who understand systems and can architect software. However, to remain an applied discipline – as it has been from its inception – computer science must emphasize software development and CS programs

63

must allocate time for student skills to mature. If we don't, we are like a music department that does not require musicians to practice before a concert or an athletics department that "trains" its athletes primarily through lectures. (p.7)

While it is important to continue CS reforms aimed at precollege students, we need to expand the scope to include elementary schools and stress fundamental concepts rather than specific languages or particular instances of technology. Therefore the intent of this study is not only to examine an approach based on lesson study, but also explore how fundamental concepts of computer programming can be understood by teachers and incorporated into lessons that help students develop the basic problem-solving and programming skills that are endemic to CS. The concept is also to develop a curriculum for CS that is not taught to some students as an "elective" but blended into the core curriculum.

Lesson Study as Action Research

Action research methods have been promoted as a transformative approach to deepen reflective practice and improve teaching (Hagevik, Aydeniz, & Rowell, 2012). Action research is used extensively in educational and social practices as an approach to resolving problems in schools and for improving teaching practice (McTaggert, 1994). Action research involves four basic stages that are derived from the work of Kurt Lewin. Lewin's action research design was developed as a new approach to educational psychology that sought to involve participants in a spiral process of planning, examining and exploratory action intended to improve social formations (Somekh, & Zeichner, 2009).

64

Action research, conducted by educators, is implemented as a cyclical process that includes: (1) describing a problem; (2) obtaining information to answer questions; (3) collecting and analyzing data, and; (4) devising and implementing a plan of action or strategy for change (Fraenkel, Wallen, & Hyung, 2011). Action research designs have several advantages and disadvantages that need to be considered by researchers. Advantages of action research include: a) that it can be conducted by any teacher or group of teachers; b) it is a proven method of improving teaching; c) it helps teachers develop deeper understanding of problems; (d) it can help teachers build a repertoire of knowledge that can be used to solve future problems; (e) it promotes a systematic method that can be shared with colleagues; (e) it can help build a community of inquiry among teachers who may feel isolated in day-to-day practice (Fraenkel et al. 2011).

Research in teacher PD and CS reform shows that single-teacher workshops and seminars are less likely to ensure sustainable CS reform than activities based on collaborative PD. Review of the literature also revealed that teacher PD activities that are of longer duration,14 hours or more, are more likely to be effective than short-term PD activities. Because the research suggests that high-quality PD usually costs twice as much to implement, there is a compelling need to find high-quality CS PD activities that can be used in schools with limited resources. There is also a need to develop a CS PD activity that proves sustainable past the pilot phase. Because CS is an unfamiliar topic a PLC activity for CS needs to provide a way for teachers to gain CS expertise before they attempt to teach CS. After reviewing a variety of collaborative frameworks, including Communities of Practice (Wenger, 1998), Collaborative

65

Inquiry (Ermeling, 2010), and Disciplinary Commons for Computing Educators (Guzdial, Morrison, Tew, & Galanos, 2011) I found articles on Japanese lesson study and read "The Teaching Gap", by Stigler and Hiebert (1999). I was struck by the practical approach embodied in Japanese lesson study and the primary importance of teacher PD for successful education reform. Stigler and Hiebert (1999) explain the primary importance of teachers as follows:

> Improving something as complex and culturally embedded as teaching requires the efforts of all the players, including students, parents, and politicians. But teachers must be the primary driving force behind change. They are best positioned to understand the problems that students face and to generate possible solutions. (p. 135)

The lesson study approach for this research draws from the work of lesson study researchers such as Catherine Lewis, Rebecca Hurd and Peter Dudley. In this study I provided all of the teachers with the textbook "Lesson Study Step-by-Step" (Lewis & Hurd, 2011). As the name implies, the book provides a systematic, step by step approach that teachers can use to develop and implement lesson study.

Ongoing research to develop the theoretical model seeks to understand how lesson study improves teaching and learning by connecting two major theoretical traditions, cognitive and situated learning theory. Cognitive theories of teaching and learning conceive improvements to teachers' knowledge as changes to an individual's mental schemata, often in response to opportunities to make one's ideas visible (Lewis, Perry, & Hurd, 2009). Lesson study draws from situated learning theory of Lave and Wenger (1991), which stresses the importance of community participation in shaping

66

the identity of individual members. Identity, according to Wenger (1998), is a way of talking about how learning changes individuals beliefs and personal histories in the context of the community. Involvement in a community shapes the future actions of individuals in accordance with an identity that they develop through participation in the community (Wenger, 1998). The participatory nature of lesson study is at the heart of the knowledge-building process. Wenger (1998) identifies essential components that are characteristic of building knowledge through social interaction. The first component, *meaning*, provides a way of talking about our evolving individual and collective abilities in a way that facilitates shared understanding of the world. For example, families struggle to establish a habitable way of life by finding ways to deal with each other through good times and bad. Doing so, they develop their own rituals, routines, symbols and stories to do what it takes to keep going (Wenger, 1998). *Practice* is a way of talking about social and historical resources that can sustain mutual engagement in action (Wenger, 1998). For example, workers organize themselves with their colleagues in order to do their jobs, have some fun and fulfill the requirements of their employer (Wenger, 1998). As communities of practice, lesson study groups talk about their work, their students and their lives. They share stories, develop routines, rituals, and symbols that they use to improve teaching and raise student achievement.

The theoretical model of lesson study, proposed by Lewis et al. (2009), draws on constructivist learning theory, lesson study research and general research of professional learning (Lewis et al., 2009). The theoretical model describes how lesson study aligns with cognitive learning theory by making various types of knowledge

67

more visible (Lewis et al., 2009). The theoretical model thereby enables teachers to encounter new or different ideas and to refine their knowledge using the systematic inquiry method that forms the basis of lesson study. The theoretical model shows how lesson study builds knowledge and ultimately improves student achievement. Figure 6 shows how the activities involved in the lesson study cycles lead to a collective understanding of community norms and thinking, developed through participation. The model further shows that the lesson study cycle leads to intervening changes in teacher CK, PK and PCK, and the development of a shared pool of lesson resources. Changes in teacher beliefs and knowledge then lead to instructional improvement and student learning.

Features of Lesson Study

1. Investigation

-*Consider students' current characteristics*
-*Consider long-term goals for student development*
-*Study content area, identify key concepts*
2. Planning
-*Develop research lesson*
-*Anticipate student solutions*
-*Write instructional plan*
3. Teach the lesson
-*Observers collect data*
4. Reflection
-*Share & discuss data*
-*Redesign lesson*
-*Summarize the changes*
Optional: Revise & re-teach the lesson

Thinking Becomes Visible

Community Norms, Tools, Identity, Participation develop

Intervening Changes in:

1. Teachers Knowledge & Beliefs
-*CK, PK, PCK*
-*Student thinking and how to capture it*
-*Long term goals for student development*
2.Professional Community
-*Motivation to improve instruction*
-*Sense of mutual accountability to improve instruction*
3.Teaching & Learning Resources
-*Tasks that reveal student thinking*
-*Data collection protocols that identify student*

Instructional Improvement

Student Learning

Figure 6. **Lesson Study theoretical model**
Adapted from Lewis et al. (2009)

68

Process of Inquiry

Carr and Kemmis (1986) define action research as a form of self-reflective inquiry undertaken by participants to enhance direct practice. Teachers and schools are drawn to action research because it is implemented in a collaborative context, leads to understanding and enables future prediction of personal and organizational change (Butterfield, 2009; Reason & Bradbury, 2008; Schön, 1983). While action research is typically undertaken as a teacher-directed form of professional development, results of initial studies can lead to district-wide improvements. For example, a classroom action research study in Madison, Wisconsin (Caro-Bruce , Klehr, Zeichner, & Piedrahita, 2007), aimed at helping teachers improve student achievement, became connected to a district-wide reform focused on improving learning among a rapidly growing number of English Language Learners and students with disabilities. One remarkable aspect of the study was that it has remained in place despite continuing budget cuts and the effects of national and local education policies aimed at deprofessionalizing teaching (Somekh, & Zeichner, 2009). As an action research project, the teachers followed an inquiry process outlined in the book, "Lesson Study Step-by-Step" (Lewis & Hurd, 2011), provided by the researcher. The book contains guidelines and examples of how to focus the steps of lesson study on student learning. Videos provided by the publisher show teachers practising each of the lesson-study steps in meetings and classrooms.

The inquiry process for examining the CS Lesson Study, as a PD activity, is focused primarily on answering the research questions. Charmaz (2014) emphasizes the importance of connecting data with the purpose of the study and the research

questions. In this study the underlying constructs of interest are aimed at understanding how the CS lesson study model helps teachers build CS content and pedagogical knowledge as well as how CS lessons influence student learning (See Table 3).

Table 3: Process of Inquiry-Research questions aligned with constructs and data sources

Step	Procedural Sequence	Constructs	Duration Setting	Data Sources
1	Process of Inquiry: CS PD Sessions 1-6: Study CS Fundamentals. Introduction to Scratch Jr, Introduction to Lesson Study.	Investigation CS Content Knowledge Collaboration Planning	1 Hour Sessions Online	Audio, Webex Meeting Recording, Researcher Field Notes in Journal, Correspondence, Class Notes and materials posted on group website RQ1, RQ2
2	Lesson Planning: Review the source curriculum, identify potential issues, talk through lesson plan, anticipate student questions and response.	CS Pedagogy Lesson Planning Student Learning Objectives	1-2 Hours School or Online	Lesson Review Notes, Journals, Anticipated questions List. RQ1, RQ3
3	Teaching Observation Lesson: Teacher teaches the lesson in the classroom. Co-teacher assists to provide logistical and technical support. Other observers watch and take notes but do not teach the class.	CS Pedagogy CS Content Student Learning	1 Hour Classroom	Observation Notes, Participant Interviews, Audio recordings of teaching sessions, Researcher field notes. Questions from Students. Student work examples: Scratch Jr projects and Handwritten instruction sheets. RQ1, RQ2, RQ3
4	Post Lesson Discussion & Reflection: Held immediately after teaching the lesson. Teachers meets to review and discuss the Observation	Student Learning CS Content CS Pedagogy	1-2 hours Study School	Audio recordings of group meeting, researcher field notes, follow up correspondence from participants.

70

Lesson. The group discusses student response, questions, and understanding of CS concepts. Research Lesson plan is revised and posted on the group website.

RQ1, RQ2, RQ3

Grounded Theory

The procedures for this study build upon previous work by action researchers including Butterfield (2009), Dick (2003) and Hayes (2011) who use grounded theory to introduce rigor in action research. The grounded theory procedures employed in this study draw primarily from the original work of Glaser and Strauss (1967) and constructivist interpretation of grounded theory described by Charmaz (2014). Constructivist grounded theory highlights the flexibility of grounded theory as an inductive, comparative, emergent, and open-ended approach of Glaser and Strauss's (1967) original work. The constructivist approach to grounded theory research recognizes the preconceptions and involvement of the researcher and potential influence on the results of the study. Rather than deny the preconceptions that researchers bring to a study, Charmaz (2014) advises researchers to first reflect and understand how preconceptions can influence data analysis and then focus on what is happening in the data rather than extant concepts that are not integral to the data.Per Butterfield (2009) the interest and concern of the action researcher is to integrate action and reflection so that the knowledge gained in research inquiry is directly relevant to the issues studied.

71

Lesson study affords teachers the opportunity to reflect with their colleagues. In lesson study reflection occurs when teachers think about the nuances of the subject matter and objectives of the research lesson and engage in thoughtful discussion with their group. In this study reflection operates on two levels. In Phase I CS PD Sessions teachers build and deepen their content knowledge of CS through reflection and discussion of teaching practice and student learning needs. By anticipating the questions that students might ask, teachers develop strategies to answer questions and make preparations for supplementing the research lesson with instructional examples. In Phase II, CS Lesson Study Sessions, teachers engage in formal reflection meetings following each cycle of lesson study. Reflection discussions are aimed specifically at identifying problems and developing solutions that will be used in the next teaching session.

In both situations reflection revolves around questions of content, pedagogy, and learning objectives for the respective classrooms. Reflection includes questions such as: what concepts do we want the students to learn? How can the research lesson plan be adapted to each of the five classrooms? How do we blend CS concepts with what the students are already familiar with? What responses and questions are the students likely to have at each point of the lesson? What technical and logistical concerns do we anticipate and how do we ensure that the lesson flows smoothly? As a cyclic process focused on building knowledge through simultaneous data collection and analysis, grounded theory fits well with the practice of lesson study. A variety of data sources were used in this study, including interview, CS PD sessions, lesson study meetings, correspondence with participants, and classroom teaching

observation. Raw data were confirmed through member-checking with the teachers. Data were triangulated from multiple sources, including field notes, Webex recordings and audio recordings to ensure accuracy.

Participants

There were seven participants in this study, 5 classroom teachers, 1 special education teacher, and 1 teacher working as the STEM coach for the local school district. Participants in this study were volunteers who signed a consent form. Participant identities were kept confidential using pseudonyms applied prior to importing data to the NVivo software for analysis. To facilitate the analysis of incremental improvements to the research lesson resulting from the lesson study process, pseudonyms for the classroom teachers and their respective classrooms were assigned according to the teaching schedule. For example, pseudonyms Teacher1 and Classroom1 were assigned to the first teacher/classroom to teach the research lesson, Teacher2/Classroom2 was assigned to the second teacher/classroom, and so on.

As a former third grade teacher at the Study School, before taking on the position of STEM coach for the local school district, Teacher 6 made significant contributions to the study. In addition to helping coordinate the classroom teaching sessions, Teacher 6 provided valuable insights on how the CS lesson study process might be implemented in other district elementary schools. As a special education teacher, Teacher 7 was assigned as an additional teacher for classes serving students needing additional assistance. Teacher 7 joined the study following the first teaching session. After receiving a briefing on the study and signing a consent form Teacher 7 participated as a member of the lesson study group and as a co-teacher during four

classroom-teaching sessions. Teachers 6 (the STEM coach) and 7 coordinated their schedules to ensure that one of them was available to help out in the classroom during the CS teaching sessions. More than one hundred students received a CS lesson during this study (Table 4).

Table 4: Third-graders enrolled at StudySchool 2016 who received CS lesson

Teacher	Boys	Girls	Total	ELL	SpEd
Teacher1	9	10	19		
Teacher2	11	12	23	9	
Teacher3	11	10	21	16	4
Teacher4	11	11	22		
Teacher5	10	9	19		
Total	51	53	104	25	4

Note: **ELL = English Language Learners, SpEd = Students with disabilities.**

Data Collection

Interviews

Interviews provided a primary source of data for this study. Both pre and post-study interviews were arranged as semi-formal convergent interviews that began with broad, open-ended questions and concluded with an agreement for follow-up questions. The pre-study interviews took place at a convenient location, a public library, chosen by the teachers. The pre-study interviews ranged from 20 to 40 minutes in length. The interview questions were designed to encourage an open discussion on teaching, computer science and professional development. The post-study interviews were held in the weeks after the study had concluded and at the convenience of the teachers. The post-study interviews ranged in duration from 40 to 90 minutes. They consisted of broad, open-ended questions designed to encourage the teachers to share

www.manaraa.com

their thoughts on the design of the CS Lesson Study PD activity, their CS knowledge before and after the study , and the effect on student learning in CS.

Charmaz (2014) advises grounded theory researchers carefully to plan their interview strategies. Among the most important considerations is to recognize the many factors that can affect interviewing such as, race, gender, social status and the credibility of the researcher. Equally important is that the research questions drive the methods of data collection (Charmaz, 2014). In this study the underlying constructs of interest behind the research questions include: *Instructional Planning, Instructional Effectiveness, CS Content Knowledge and CS Pedagogical Knowledge*. These were treated as anticipated categories that were used to organize data throughout the study.

Convergent Interviewing

In this study a technique known as convergent interviewing was used to guide both the pre- and post interviews with participants. Convergent interviewing is used by action researchers in a variety of complex and uncertain situations as a structured approach to data collection and analysis (Dick, 2016). Convergent interviewing seeks to engage the interviewer and interviewee in a four-stage process. The interview begins with an introduction and discussion to build rapport, followed by a single open-ended question. The interviewer pays close attention to what the participant says and develops appropriate general or specific probing questions based on the initial response. As the interview wraps up, the interviewer may summarize and ask the participant to confirm the interviewer's understanding. The interview closes with the

75

interviewer's thanking the participant for taking part, repeating the assurances of confidentiality, and a request for permission to return with follow-up questions.

Digital Recording

Audio recordings were used throughout this study to provide an audit trail of content and duration of events, facilitate triangulation of data sources, and to ensure that interpretations were based on actual experience of the participants. All audio recordings were made using a Zoom H1 - 24 bit/96kHz digital audio recorder.

Online Meeting Software

Online meeting software provides convenience, flexibility, and provides a cost-effective means to enable collaboration. The use of online meeting software for teacher PD is supported by research showing that teachers who collaborate online, as a PLC, are engaged with the group, develop a sense of community, improve their knowledge of subject and pedagogical content, and tend to modify their instructional practices accordingly (Blitz, 2013). Flexibility is the strongest advantage of online PLCs over the traditional face-to-face environment in facilitating teachers' learning (Blitz, 2013). The online environment enables teachers to access and share knowledge in a timely manner. One interesting finding of a literature review comparing online with face-to-face PLC's is that online PLC's are consistently found to be better at promoting self-reflection on learning and instructional practice than are the face-to-face environments, even though both models appear to contribute equally to learning and mastering content knowledge (Blitz, 2013).

76

Webex online meeting software provided the participants with a convenient way to learn about CS during the CS PD phase of this study. During online CS PD Webex meetings the researcher used the Webex screen-sharing feature to allow participants to view educational materials on CS, including slide presentations, videos, and Scratch Jr programming demonstrations. Webex recordings were used to facilitate transcription of participant statements and triangulation of other data sources, such as researcher field notes, and to ensure that interpretations were based on actual experience of the participants.

Researcher Field Notes and Correspondence

Researcher field notes were entered by the researcher in spiral bound notebooks throughout the study. Field notes provided a source of data that was useful as an audit trail and to facilitate triangulation with other data sources. Field notes were transcribed and formatted to include relevant information and pseudonyms prior to being imported to NVivo. Correspondence with participants provided both a rich data source and an audit log aligned with the timeline of the study and contained details of questions and topics of discussion. Correspondence was anonymized and copied into text files before being imported to the NVivo project.

Data Analysis

Initial coding was performed as the raw data were organized and formatted to replace proper names with appropriate pseudonyms. Per Charmaz (2014), the coding strategy remained open and close to the data yet moved quickly through the data. Codes were kept short and, where possible, focused on words and phrases of the

77

participants that conveyed ideas and preserved action. During the initial coding process memos were used to keep track of themes and categories that would inform changes in the process and to guide further analysis. Coding notations were entered using a convention that made them stand apart from the data and easy to retrieve using automated queries in the NVivo software. Because the voices of the teachers provide a primary data source, significant time was spent transcribing and verifying the accuracy of the statements made on the audio recordings. Transcriptions of audio recordings were made by the researcher and coded using line-by-line coding during and after transcription. Codes were entered in the research journal during interviews and transcribed to field notes after their conclsion. Notes entered in the research journal during interviews were reviewed during transcription and coding to ensure accuracy and meaning. After five teaching and reflection cycles the lesson plan was finalized, and posted to the group's shared folder. The lesson plan for CS Lesson 1 was used as the basis for proposed CS Lesson 2 research lesson that the teachers developed after the study.

Ethical Concerns and Confidentiality

Qualitative research must ensure that investigation is conducted in an ethical manner and includes techniques to minimize threats to trustworthiness. Of primary concern in any type of research is that participants are treated with respect, that they suffer no physical or psychological harm, and that their identities and confidential information, remain confidential. In planning this study I spent several months discussing it and meeting prospective participants before formalizing the arrangements with the school principal and third-grade teachers at the Study School. During the

recruiting phase prospective participants were provided with a summary of my research proposal, demonstration of the CS concepts that they would be learning, and an overview of the lesson study process.

     All participants remained volunteers who were informed that they could leave the study at any time, for any reason. There were initially 6 teachers who agreed to participate; the special education teacher at the school joined the study later. Having agreed to participate, the teachers signed an approved consent form. Table 5 presents the study participants' pseudonyms assigned to maintain their anonymity.

Table 5*:* Pseudonyms used in CS Lesson Study research

Classroom Teachers	Teacher1, Teacher2, Teacher3, Teacher4, Teacher5
STEM Coach	Teacher6
Special Educator	Teacher7
Principal	Principal
School	Study School
Students	Student, Students, Another Student

CHAPTER 4

FINDINGS

Introduction

This chapter will begin with a brief statement of the problem and research questions followed by an overview describing the data collection and coding process. Explanation of how data exemplars were chosen for presentation will include background on the context in which the data were collected. A framework of major and sub-codes will be provided to illustrate the different levels of coding and how data were categorized. The top level coding will be described to help the reader differentiate anticipated and emergent themes. Whenever possible the presentation of findings will include actual examples of data, including comments selected from transcriptions and figures showing actual student work product.

The chapter is organized in two sections. The first will provide a summary of the findings and supporting examples relevant to the constructs of interest. The second section will provide details of each of the four data collection phases that align with the research questions. The next chapter will provide a summary of the study, conclusions and interpretations of the findings, limitations to the study, and suggestions for future research.

Findings: Section I

Research Question Review

Despite the growing pressure from policymakers and industry for schools to teach CS, teachers in US public schools have limited time and resources available to

engage in substantive PD for CS. This study examines a CS Lesson Study teacher PD activity designed to provide substantive CS PD within the realities found in US school districts. Three research questions were stated to help guide the investigation:

RQ1: How does CS Lesson Study influence CS instructional planning by elementary school teachers?

RQ2. How does CS Lesson Study influence CS instructional effectiveness of elementary school teachers?

RQ3. How does CS Lesson Study among teachers, influence student CS learning?

Overview of data collection and coding

Per Glaser and Strauss (1965) the presentation of the qualitative findings is intended to present a schematic but accurate description of how the research was conducted to make it possible for others to use in other settings. In this study the presentation of the findings is intended both to explain the results of the study and provide details of CS lesson materials, participant background and strategies used throughout the research.

Data collection and analysis were performed simultaneously, following grounded theory methods described by Charmaz (2014). As an action research project aimed at improving the underlying design of CS Lesson Study PD, the constant comparative analysis of data led to process changes during the study.The categorization of data during the study advanced both my conceptual understanding of the process as well as the teachers evolving CS content and pedagogical knowledge.

Initial coding was performed through each of the four phases of data collection including pre-study interviews, CS PD Sessions, CS Lesson Study Sessions and post-

study interviews. Throughout the study the researcher kept field notes and made audio recordings during each interaction with participants. The researcher attended classes when the research lesson was taught in each of the five cycles. The researcher participated in the classes by providing technical support for the teacher, and taking part in the Simon Says game. The researcher also answered questions from teachers and students during classes. The researcher kept a journal and made audio recordings of each class. Data entered in field notes included date, time, location, participant information, and descriptions of the interactions. Grounded theory researchers use memos to document analytic ideas and concepts (Charmaz, 2014). In this study memos were created using a variety of media including handwritten documents, text messages and emails. Text messages proved particularly effective as a means of documenting ideas that emerged immediately after CS Lesson Study session. Using text messages as grounded theory memos not only helped the researcher document ideas as they occurred but also established an audit trail of analytic concepts emerging throughout the study. Because this action research project was focused on exploring the ongoing CS lesson study process, data and emerging analytic theories were shared and discussed among the group. Transcript excerpts and data exemplars used in this report were reviewed by the participants.

Charmaz (2014) suggests that grounded theory coding procedures be kept simple, direct, analytic and emergent. Because grounded theory is used in action research to establish rigorous data collection and analysis processes, the coding procedures were kept simple and limited to a two-level schema that included intial and focused coding.

82

Focused (secondary) coding was performed as raw data were organized and imported into the NVivo software. Per Charmaz (2014), focused coding provided a way for the researcher to organize the initial codes into conceptual categories that align with the analytic directions of the study. In this study secondary codes and themes were developed by moving back and forth between the transcribed notes, audio tapes and correspondence to compare data with data and data with codes. Initial codes were derived from participants words, phrases, notes, comments and from student programming projects generated through the four phases of data collection. Prior to the study the researcher had explained the grounded theory method and coding strategy that would be used. Figure 7 shows the coding strategy used in this study.
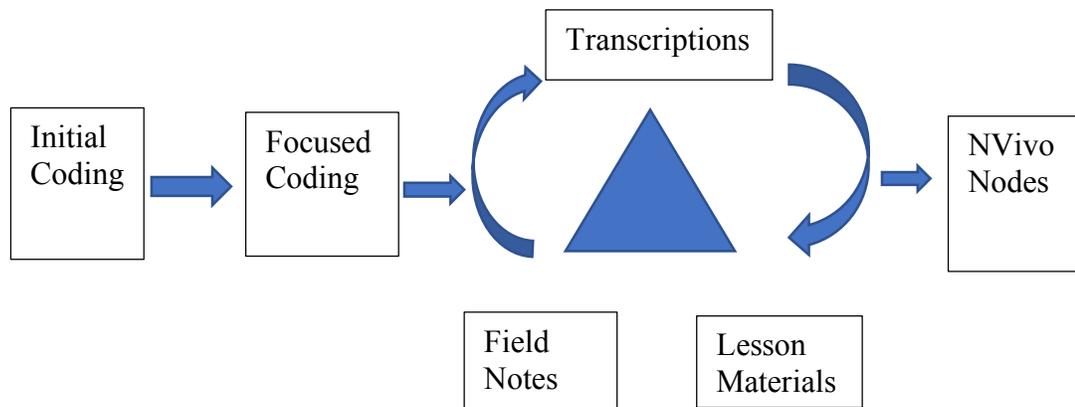
Figure 7. **Diagram showing coding procedures**

Data exemplars chosen for presentation

Each example of data presented in this chapter will be explained in terms of the data collection process that was used and the timeframe within the the study when the data were collected. In this study there were two sets of Scratch Jr projects collected, those created by the teachers during the CS PD Session phase and those created by students during the ten hours of classes in which teachers taught the CS lesson.

Following each of the five lesson study cycles, the researcher cleaned and re-powered the tablets and retrieved Student Scratch Jr projects. The Student Scratch Jr projects were posted in a shared folder used by the teachers for CS Lesson 1. With over thirty Scratch Jr projects collected the researcher had the teachers review and select Scratch Jr projects that exemplied student CS learning.

Coding Schema

Initial coding produced a wide range of codes that were revised and refined throughout the study. Since the bulk of the data originated from audio recordings made during the four phases of data collection, a majority of initial codes were created by listening to audio recordings. Pre and post study interviews were transcribed verbatim by the researcher and imported into the NVivo after pseudonyms had been substituted for names of people and places. Recordings made in classrooms were used within the lesson study cycles to inform changes to the process. Transcriptions of the classroom sessions were used to compare each of the parts of the lesson between cycles.

84

Word frequency and synonym queries were run after data had been imported into NVivo. Results of queries were further analyzed through comparison with source documents. Major themes were established and compared with the constructs of interest relevant to the research questions. Because the purpose of this study aimed to investigate the CS Lesson Study PD activity, the final coding schema contained constructs relevant to the research questions and context of CS Lesson Study in the school and district. NVivo nodes were used to organize high-level codes and subcodes linked to data were used as the second tier. For example, the high-level code *Community* includes sub-codes relevant to collaboration among teachers in the school and school district.

Four major codes were derived that aligned directly with the research questions, including: *CS Content Knowledge*, *CS Pedagogical Knowledge*, *Instructional Planning, Instructional Effectiveness, and Student Learning.* Two major themes, *Curriculum Enhancement* and *Community*, emerged and were included in the coding schema. Table 6 provides a summary of major codes and sub-codes derived from the data.

Table 6: CS Lesson Study Major codes and sub-codes

Major Codes (NVivo Nodes)	Sub-codes
CS Content Knowledge	Understanding how computers work
	Understanding why we have computers
	Understanding how to write programs
	Understanding program constructs

85

www.manaraa.com

CS Pedagogical Knowledge	Explaining CS to others
	Teaching CS to students
	Teaching Scratch Jr programming
	Explaining algorithm development using program constructs
Instructional Planning	Planning effective CS lessons
	Collaborating with peers
	Planning to engage all students
	Planning to blend CS with curriculum
Instructional Effectiveness	Formulating goals for student learning
	Tailoring CS lesson to the classroom
	Improving the lesson
	Engaging all students
Student Learning	Students demonstrate coding skills
	Students learn about CS careers
	Students demonstrate teamwork and social skills
	Students demonstrate CS problem solving skills
	Students demonstrate creativity
	Students demonstrate ingenuity
Curriculum Enhancement	Aligning with district initiatives
	Collaborating with Principal
	Blending with curriculum
	Identifying barriers to CS
	Identifying opportunities
	Shaping the future
Community	Collaborating
	Managing Career Responsibilities
	Supporting District Initiatives
	Promoting CS
	Engaging students and parents

Major Findings

This section will describe the four major findings of this study. These findings have been organized and presented to answer the research questions as follows: the first three findings address RQ1 and RQ2, the fourth finding addresses RQ3.

- **Finding 1**: CS Lesson Study PD helped all of the teachers to increase their CS content and pedagogical knowledge and acquire computer programming skills. This finding addresses RQ1 which examines increased CS content and pedagogical knowledge relevant to the major theme *Instructional Planning*.

- **Finding 2**: All teachers in the study found that CS Lesson Study enhanced CS instructional planning.The majority of teachers suggested that the school district recognize CS Lesson Study as an approach to make CS available to all students. Three of the five teachers suggested that the school district offer CS Lesson study as an option that teachers could choose to fulfill required PD hours. This finding addresses RQ1 relevant to the major theme *Instructional Planning*.

- **Finding 3**: CS Lesson Study enabled teachers to teach effective CS lessons in their classes. Data show that teachers were able to introduce and explain fundamental concepts of CS to their students in a way that prompted students to engage in collaborative CS projects with their peers.
This finding addresses RQ2 which seeks to understand how CS Lesson Study influences *Instructional Effectiveness*.

- **Finding 4**: Data collected through classroom observation, student work examples and interviews with teachers suggests that CS Lesson study had a positive influence on CS student learning. Data suggests that CS Lesson Study provided teachers with a way to help students make progress toward long and short-term learning objectives, such as collaboration, perseverance and problem solving.

**Finding 1 – CS content and pedagogical knowledge**

Finding 1 helps answer RQ1 because high-quality instructional planning depends, for the most part, on teacher content and pedagogical knowledge (Harris & Hofer, 2011). Data collected during post-study interview showed that all teachers participating in the study reported an increase in CS content and pedagogical knowledge as a result of CS Lesson Study PD. Four sub-codes align data to CS content and pedagogical knowledge: *Understanding how computers work, Understanding why we have computers, Understanding how to write programs, Understanding program constructs.*

Teachers reported that they learned fundamental CS skills through their participation in CS PD Sessions. For example, in her post-study interview, Teacher1 said "I think that I understand how computers work and how programmers and computer scientists make computers work. Yes, I definitely have a better understanding and could teach it to other people."

88

In her first year as a classroom teacher CS Lesson Study provided Teacher4 with a vehicle to develop CS knowledge and pedagogical skills. In her post-study interview Teacher4 said:

> I didn't have that much knowledge of computer science at all so this process did help me to learn more. I did like how we broke it down and we focused on certain areas rather than altogether at once. We broke down the lessons and things like that. I liked that we all worked together as a group.

Evidence of increased CS pedagogical skills among the teachers was found in materials teachers developed and used to teach CS. Figure 8 provides examples of how the teachers introduced CS by aligning it with concepts that students were learning in other disciplines, such as science and literacy. For CS Lesson 1 the example on the left shows a slide that teachers used to align CS with an inquiry process that the students were familiar with from their science lessons. The example on the right, from the CS Lesson 2, shows how the teachers used literacy skills, developing a narrative, that students were familiar with further to engage students in CS and computer programming.
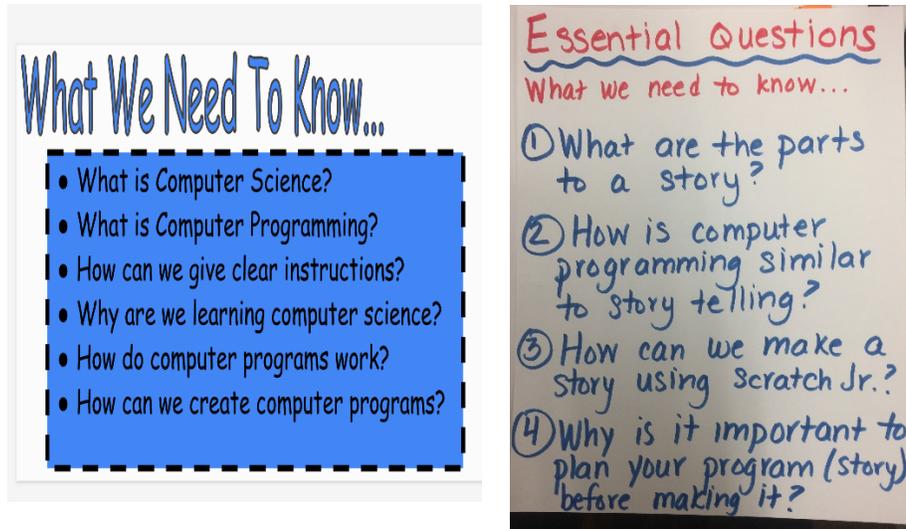
Figure 8. **Lesson Materials – Inquiry Questions CS Lesson 1 and 2**

That teachers learned to write computer programs is evident in examples of Scratch Jr projects that the teachers created during the CS PD phase of the project. Figure 9 is a screenshot from the first Scratch Jr project created by Teacher2. The program entitled "A Walk in the Woods" shows various characters (frog, snake, bird, lizard) interacting. All of the teachers became proficient at writing Scratch Jr programs. In this study the term "proficient" means that the teacher, or student can develop a Scratch Jr project that includes several objects and backgrounds (settings) and uses a variety of instructions to animate the project.

Figure 9. **Scratch Jr Project Teacher2**

Tables 7 and 8 provide additional examples of coded excerpts that support Finding 1.

Table 7: Major Code - CS Content Knowledge

SubCode	Excerpts from Data Sources
Understanding how computers work	I think that I understand how computers work and how programmers and computer scientists make computers work. Yes, I definitely have a better understanding and could teach it to other people. Source: Post-study interview: Teacher1
Understanding why we have computers	I began to really think about what things have computers and what different programs are used for different things, like radio and TV, and thinking about it a little more. Source: Post-study interview: Teacher4
Understanding how to write programs	I knew they would be excited about it and I thought it was great. They took to it. And it definitely showed that they can work together. You know, collaborate, stick with it. I saw there were quite a few of them who wanted to keep going with it because by the time we were ready to show their projects they wanted to keep going, they wanted to fix what was wrong with it. Source: Post-study interview: Teacher5

91

Table 8: Major Code - CS Pedagogical Knowledge

SubCode	Excerpts from Data Sources
Teaching CS to students	Teacher1 Alright, when you are creating programs you have to give very clear, very specific directions. Have you ever given very clear or specific directions about anything? Have you? Student How to have someone help clean the room. Teacher1 Right, so you must give very clear directions about how you want them to help you clean the room. So, learning how to program, you must give very specific directions. So, we are going to do a little activity to practice listening to directions, and then you are going to practice giving specific directions. In computer programming, computers have to be given specific instructions. And the instructions must be given in a certain order. Source: CS Lesson 1/Cycle 1: Transcript
Teaching Scratch Jr programming	Teacher5 The checkmark. Now remember here is where you are going to get it to move. And what are these numbers down here? Students How many. Teacher5 How many times you want them to do something. Except, for the jumping. The jumping number doesn't tell you how many times, it tells you how high. So, if you want it to jump higher you make the numbers bigger. If you want it to jump more than once you have to put another jump block in. Invisibility are these purple ones. Right here, what happens? Students

92

Disappears

<u>Teacher5</u>

It disappears, and then it comes back. This one makes it get, what do you think?

<u>Students</u>

Bigger!

<u>Teacher5</u>

Bigger right. See how it is starting off small and the person is getting bigger? And then this one is bigger and it is getting…

<u>Students</u>

Smaller.

<u>Teacher5</u>

Why might you want to make a person get bigger or smaller? How about art class?

<u>Students</u>

They're at a far distance.

Source: CS Lesson 1/Cycle 5 : Teacher5

Able to explain algorithm development using program constructs	Programmers write instructions. And the instructions are called programs. And the program is the language of the computer. You can write programs in different languages. We are going to learn one particular one, Scratch Jr. And they tell the computer what to do in a specific order, a sequence, right? Remember when we do the algorithm for addition and subtraction?

<u>Students</u>

Yes.

<u>Teacher5</u>

Do we have to follow a specific order? Sequence?

<u>Students</u>

Yes.

Source: CS Lesson 1/Cycle 5 : Teacher5

93

Finding 2 - CS instructional planning

In this study all teachers found that CS Lesson Study enhanced CS instructional planning. The teachers continued to work together after the study to plan and implement a second CS lesson, CS Lesson 2. The majority of teachers suggested that the school district recognize CS Lesson Study as an approach to make CS available to all students. Sub-codes associated with *Instructional Planning* include: *Planning effective CS lessons, Collaborating with peers, Planning to engage all students, Planning to blend CS with curriculum*

When teachers plan their instruction they focus on content knowledge, teaching strategies and student learning needs (Harris & Hofer, 2011). In the theoretical practice of lesson study instructional planning is regarded as a vehicle to develop teachers' content and pedagogical knowledge, sense of professional community and shared beliefs on instructional improvement (Lewis et al., 2009). For this study teachers worked as a lesson study group to plan and implement multiple cycles of lesson study focused on CS. Teachers' use of CS to enhance student learning became a part of the planning process. For example, in planning the lesson for the classes with ELL students, the teachers found the CS lesson engaged all the students in a way that they had found difficult in other subject areas. Teacher2 described her view of how Scratch JR programming helped her plan the lesson for her students:

It's interesting, though, that it's like a language. Computer science is like a language: we know that languages are best when they are learned young. So that's been the U.S.'s problem all along, that they wouldn't begin teaching languages until middle school, when that window is already closed. The thing I

www.manaraa.com

like, being an ESL teacher, is the fact that you don't need a language skill to be able to do this. It would make them feel successful. It would be giving them a voice. You know what I mean? It would be giving the ESL students a voice that they would otherwise not have . It would give them a showcase where they could show their thinking and thought processes. They could tell their stories. And the whole collaboration part of it. I think it is really ideal for english language learners.

With a significant number of ELL students in their classes, Teachers 2 and 3 took advantage of CS to plan lessons that engaged all students. This shows that instructional planning is a process that teachers use to achieve greater flexibility and responsiveness to students in class (Harris & Hofer, 2011).

Finding 3 - CS instructional effectiveness

In this study the materials and examples used in class were created by the teachers working as a collaborative lesson study group. Instructional effectiveness is enhanced when educators engage in sustained, intensive, high-quality PD activities that might include peer observation, analysis of student work, and student-centered discussion (Garet et al., 2001; Ratts, Archibald, Street, Andrews,…& Street, 2015).

Sarama and Clements (2009) observed that computer programming helps young children develop analytical skills by requiring them to construct a sequence of symbolic commands to control an object. Working with Logo programming language (a predecessor of Scratch Jr), Sarama and Clements (2009) found that the immediate feedback that students receive as they write their programs helps them become explicitly aware of geometric and mathematical concepts. Sarama and Clements

95

(2009) observe that effective teachers integrate CS into an ongoing curriculum in ways that constantly encourage experimentation with open-ended problems and closely guided tasks focused on fundamentals. Furthermore, according to Sarama and Clements (2009), effective CS teaching includes a mix of on and off computer activities and cooperative projects.

In this study data collected through interview and observation support the finding that all of the teachers taught effective CS lessons. Since the objective of CS Lesson Study is to provide a CS PD activitiy that teachers can use to teach CS to all students, an effective CS lesson is one that engages all students. Table 9 provides a number of examples aligned with *CS Instructional Effectiveness*.

Table 9: Major Code - Instructional effectiveness

SubCode	Excerpts from Data Sources
Formulating goals for student learning	Researcher What did you do to prepare for the class? Teacher3 I found the book *The Magic Bus Gets Programmed* and I thought that might be good just to make a connection with the concepts that we wanted them to learn. And for them to see that with programming things can go awry. Source: Post-study interview: Teacher3. Note: in this example Teacher3 (with 16 ELL students in her class) read a story about CS on the day before she taught CS Lesson 1. This example also shows how the teacher blended CS into her reading lesson.
Tailoring CS lesson to classroom	Teacher4 Computers work when they are given a clear set of instructions in a specific sequence. Ok? Think about instructions. What are instructions? [calls on a student] Student It shows you what to do first, and then second, and third. Teacher4 Very good. Instructions are things that tell you what to do. Kind of like my Work board here. You come in in the morning and you know what to do.

What about sequence? What does that mean? Sequence? So, we have instructions in a specific sequence. What do we think that means?

Student

Order

Teacher4

Order, very good. You're coming in the morning, you know you put your chairs down, you know you get started on your morning work. There is a sequence. Every day you are supposed to do math first. Then you are supposed to do daily language. When you finish both those things, you read. Correct?

Students

Yes.

Teacher4

We have a sequence every day to our morning routine. That is the same thing with computers. You have a clear set of instructions and it has to be done in a specific sequence.

Source: Transcript from CS Lesson 1/Cycle 4 : Teacher4. Note: This example shows how a novice classroom teacher developed an analogy for *the Sequence* program construct.

Improving the lesson	### Teacher2 What happened in my class was that they were having so much fun with the audio that they didn't use other instructions. ### Teacher3 Right, they had no instructions. And they went right to the audio and they were trying to figure that out. And so I am like, why don't we start moving first? Let's write some instructions to get it to move. And then if you have time you can play with the other instructions. ### Teacher6 Right and so when we handed out those tablets in Teacher4's class I said, you will have instructions for both of your characters. Trying to really reinforce that. Source: Transcription from CS Lesson Study *Reflection* meeting following Day2 of Cycle3. Note: This example illustrates how the teachers worked to identify issues and improve the lesson. As a result of this discussion Teachers 4 and 5 emphasized that students pick Scratch Jr characters and create a story that they can use as the basis of their Scratch Jr project.

Engaging all students	Teacher3 I have 22 total students and I have four students that have IDP's and I have two of those receive speech and language services; one of them has occupational therapy and one physical therapy.
	Researcher And how do you think they did with this class?
	Teacher3 Oh I think they loved it. One of my students, he was the top one with the Scratch JR, it gave him a chance to relate to the other students. It wasn't like a math test or involved reading or anything like that. They were all equal and able to express themselves. So I think they all did a great job with it. No one was excluded or anything like that.

Source: Post-Study interview with Teacher3.

Finding 4: Student Learning.

The results of this research show that CS Lesson Study among teachers had a positive influence on student learning in CS. The students working in pairs and three produced Scratch Jr projects that showed creative use of Scratch Jr commands, many of which the students learned how to use themselves. Data from observation notes, audio recordings of the classroom sessions, and teachers' comments in Reflection meetings and post-study interviews support the finding that the students not only learned the basic instructions taught by the teacher but also how to use additional instructions. Students in each of the classrooms shared their 'discoveries' of new ways to use Scratch Jr instructions. The sharing of knowledge between groups of students created a dynamic environment in which simple Scratch Jr projects quickly evolved to include advanced audio and visual features beyond what was expected. In the next example Teacher2 describes how the students learned from each other.

I think it was because they learned. They learned that whole audio thing from each other. I think that they were figuring out that they were learning from each other. And they were like "oooo, what did I miss? What else could I have learned". I don't think they recognized that but I think that's why they were interested in doing that.

Teacher-centered collaborative learning activities seem to be more effective in improving student achievement than individual PD activities such as PD workshops, university courses and individual learning activities (Akiba & Liang, 2016; Vescio et al., 2008). Educational researchers say that teacher expertise is the single most important factor influencing student achievement and that ineffective teachers can damage a student's academic career (Sack, 1999; Wong, 2000). Studies also show that that high-quality teacher PD involving lesson study can enhance student achievement (Waterman, 2011). Studies on student learning in mathematics have shown that students whose teachers were in lesson study groups scored significantly higher than students whose teachers were not (Barrett, Riggs, & Ray, 2013).

In this study student CS learning is defined as the ability of students to learn fundamental CS concepts from their teacher and use those concepts to create Scratch Jr projects with their classmates. The Scratch Jr projects produced by students provide evidence of student CS learning. Four sub-codes associated with student learning in CS were developed from the data including: *Students demonstrate coding skills, Students learn about CS careers, Students demonstrate teamwork and social skills and Students demonstrate creativity.*

Table 10: Major Code - Student learning

SubCode	Excerpts from Data Sources
Students demonstrate coding skills	**Teacher6** I had the flexibility to see most of the lessons and kind of support the kids and the teacher. It kind of made me feel like wow! They really, kind of almost like an outsider, they really can do this. They can persevere, they can listen to those directions and say "ok, I can do this". **Teacher3** Yes, it was nice to see them go back and reevaluate what they had. Like I had one student who went back and said, "oh yeah, wait a minute, I can't go that way with it I have to go this way" **Teacher5** I have to say I was a little apprehensive about how this was going to go with the kids because they've had very limited exposure. I know a lot of the kids don't even have computers at home never mind programming. I knew they would be excited about it and I thought it was great. They took to it. And it definitely showed that they can work together, collaborate, stick with it. Source: CS Lesson Study/Cycle 5 Reflection meeting
Students learn about CS careers	**Teacher1** Have you ever thought about what kind of job you want to have? Or what you might want to do when you're older? So, what we are going to practice today is something else that you can maybe do when you're older. And that is to become a computer scientist or a computer programmer. Can everybody say computer programmer? **Students** Computer programmer! **Teacher1** A computer programmer is a type of computer work. So, what we want to think about today is being a computer programmer, working on computer science, or being a computer scientist. When we think

100

|	
	about being a computer scientist, we think about how computers work.
	Source: CS Lesson 1/Cycle1 : Teacher1
Students demonstrate teamwork and social skills	<u>Teacher6</u> I really saw how it could be really purposeful, meaningful, and almost like blend into the day to day curriculum without it being considered like a separate entity to everything going on in the room. <u>Teacher1</u> It's that you're really create, do these problems, and solve problems in different ways. Not just doing math, not just doing the science test, that there are other ways that you can incorporate problem solving and collaboration. To get them ready for the future and their next grade. Source: CS Lesson Study/Cycle 5 Reflection
Students demonstrate creativity	<u>Teachers Meeting</u> We had them pick their own character. And I was explaining to them that it's a noun. It's a character, it doesn't matter if it's a ball or something like that. I saw that some of them were trying to get the player and the ball to go and they were really trying to get it to go in the goal. And then there was another group that had a nice little program that had a car and I showed them how to click on the grid and change the color of the car. And that group went and showed four other groups how to change the colors on things. Then I showed one group the camera and then, oh my gosh! They all went around and did it. It was just like you might have thought that they might not know how to do it, but within a minute, they all were doing it. There were scuba divers, people in cars, you name it.

Student CS Projects

Examples of student work provided a rich source of data for this study. Students worked together to develop their Scratch Jr programs and present them to the class. Many of the student projects provide evidence that knowledge building occurs though collaborative interaction. Student projects were retrieved from the tablets after each lesson-study cycle. The following examples highlight the creativity and collaboration that took place in the classrooms.

Example 1: Students collaborated on a story about going to the moon. They learned how to use the camera to put themselves in the space suits. They remembered how to use the "looks" instruction, demonstrated by the teacher, to resize the different astronauts (Figure 10).
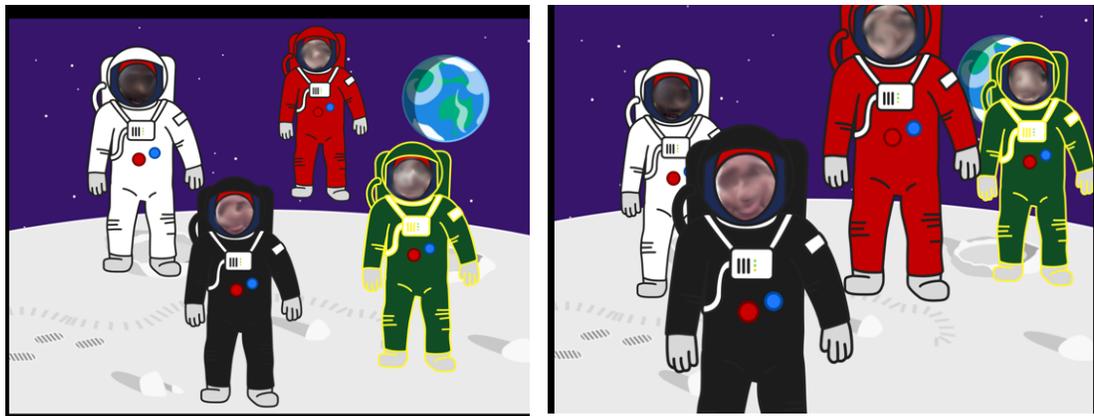
Figure 10: **Four students worked to develop a story about going to the moon**

Example 2: In the next example students persevered in writing their program to make the ball stop where they wanted. They added an audio instruction and recorded their voices shouting "Goal!!". As they presented their work to the class the students

talked about how they chose their objects and worked hard to get their program to work the way they wanted. One of the students in this group said, "That took a lot of work!" (Figure 11).

Figure 11. **Goooaaal! Lots of counting as students wrote their code**

Example 3: The next two examples show how the students learned to use Scratch Jr instructions to create particular Scratch Jr programs to help them tell their stories. Rather than use the predefined settings available, these students figured out how to use Scratch Jr instructions and techniques that were not presented by the teacher. The teacher demonstrated the "Resizing" instructions and asked the students what they might use these for. The students remembered their lesson on perspective from art class and used this to create depth in their project scenes.
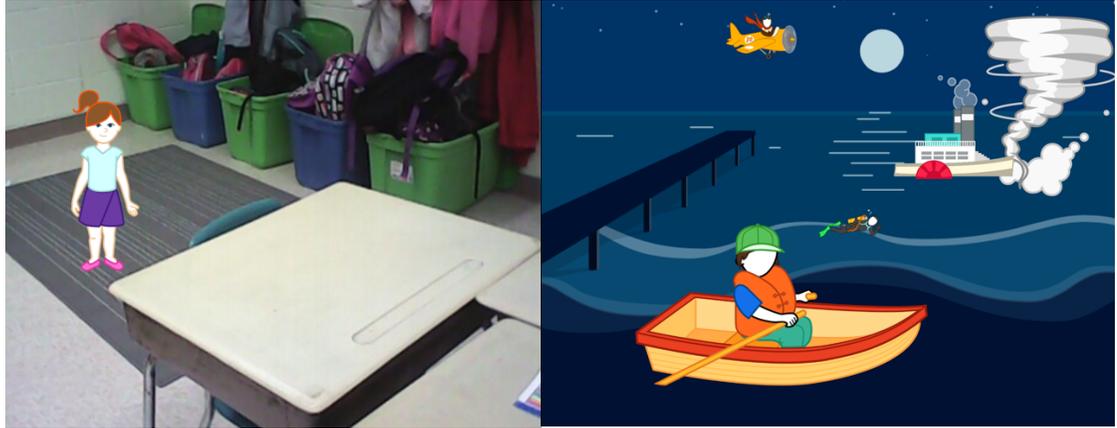
Students created their own setting   Resize instructions create perspective

Figure 12. **Students projects went beyond what the teacher demonstrated**

Findings: Section II

Without prior research examining CS Lesson Study the question arises whether the findings represent a "substantive" grounded theory. Glaser and Strauss (1965) say that the grounded theory researcher's job is to inform the reader so that they can make sensible judgments regarding the value of the findings for their own situation. To accomplish this the researcher must: (a) provide the reader with enough data so that they understand the theoretical framework, and (b) vividly describe the social world studied.

The data suggest that CS Lesson Study provides an effective teacher PD activity that increased teacher CS content and pedagogical knowledge, enhanced instructional planning and had a positive influence on student achievement in CS. The next section intends vividly to describe the CS Lesson Study project through the four chronological data collection phases. Examples of the CS lesson material, participant profiles and dataexamples will be provided to inform future research. The successful

104

implementation of CS Lesson Study is highly dependent on: (1) the willingness of the teachers to participate and persevere through difficulty; (2) the spirit of cooperation among participants; and (3) the support of the school principal.

Four Phases of Data Collection

This section will describe how the CS Lesson Study process was implemented over a three-month period from mid-August to November 2016 at the StudySchool. The section provides a summary of the four data collection phases: pre-study interview, CS PD Sessions, CS Lesson Study Sessions, and post-study interview.

Data Collection Phase I: Pre-Study Interview

Pre-Study interview with teachers were conducted on August 2 through 4, 2016 at a public library in Rhode Island. The interviews were aimed at understanding the teacher's perspectives on CS, teaching in elementary school, their reasons for wanting to be involved in the study, and prior experience with collaborative PD such as lesson study. In addition to collecting background information, the pre-study interview served three additional purposes. Firstly, as an action research project, it was essential to establish a rapport with the teachers, provide details on the project, answer questions and explain my role as a participant in the CS lesson study group. Secondly, because I was responsible for teaching CS to the teachers, the pre-study interviews provided the background information needed to finalize the syllabus for the CS PD sessions. Also, since I was responsible for providing technical support throughout the project, pre-study interview helped to ensure that teachers were comfortable contacting me for help with questions related to hardware, software or course material.

105

Finlly, the interviews provided a way for teachers to express concerns regarding potential situational or institutional barriers that might compromise the research.

Because the teachers were very interested in participating in the project the results of the pre-study interview provided the background and basic understanding that I needed. The informal, open-ended questions served to encourage frank discussion on topics of interest. Convergent interviewing helped establish a collaborative rapport between the participants and me that continued to develop throughout the study. Beginning with the pre-study interview, *Community* became a central theme of this study. The following narratives provide background, teaching experience and prior knowledge of CS and Lesson Study for each teacher in the study.

Teacher1

After finishing her degree in 2007 Teacher1 spent a year teaching second grade before being hired as a special education teacher at high school. She then moved on to a third-grade teaching position for one year, then a year teaching fifth grade, and has been a third-grade teacher at StudySchool for the past five years. Teacher1 is a dedicated professional who took on responsibility to help lead this project from start to finish. As a primary contact during the recruiting stage Teacher1 worked with me to develop a research plan that would fit in the context of the school. Teacher1 described her expectations of the project as follows:

My goal is to implement the project as we've planned. Sometimes there are hurdles along the way but I hope it all works out. I think that it will be a great process for the kids to experience. I don't really have any questions about it at this point. I think that we will make it work as we go along.

106

Teacher2

Having received a bachelor's degree in industrial design Teacher2 worked in the jewelry industry in Rhode Island. She began to look for another career as the industry began to change and companies began to downsize. As a graduate from art school, Teacher2 wanted to find something that would be creatively challenging. She said that she turned to teaching because working in a shrinking industry was no fun anymore; toward the end it had become "a lot of faxes and tedious office work". Having earned a master's degree in education in 2006, she taught 6th grade at the only bi-lingual school in the district. The school was setup to allow students to have different teachers in the same semester, learning in Spanish in one quarter, and English in the other. Following a departure from teaching to take care of a family issue, Teacher2 worked as a substitute teacher for six years before taking positions teaching second and third grades. At the time of this study she had recently received her ESL endorsement and would be teaching third grade ESL. Before joining this study Teacher2 had no prior computer programming experience. She reported her previous experience with computers and information technology was less than positive:

> Funny story was that when I was at Art School they put in a state of the art computer lab but wouldn't let the juniors and seniors use it. They were like "you're already too far gone" so we were only allowed to use it for word processing! When I went into industry I was having to do data entry on huge monitors using software that had issues. The IT guy was really condescending and I found him difficult to deal with.

Regarding learning about CS and Scratch Jr as part of this study, Teacher2 said:

107

I think it will be good that we are learning it too. Because we will be able to relate to the children more; saying "oh don't worry I was just as frustrated". You know when you become an expert, you kind of forget. Like what it was like to drive a car, until you have to teach someone how to drive a car. Then, you find kids will surprise you, like with Google drive they will show me something and say "Mrs Teacher2 look at this" and I'll say "oh, I didn't know it could do that or I didn't know how to do that", then I'll say ok everybody let's go and everything is really like "how did you do that?". I am very excited to see how the kids take to learning programming and using the tablets. I like that Scratch Jr can be programmed using drag and drop rather than text. I am curious to see how the ESL students take to this as compared with their other work.

Teacher3

With over 15 years' experience in elementary school ESL, Teacher3 contributed greatly to this study on many levels. She began her teaching career as a head teacher at a nursery school before taking a sixth-grade ESL position. Over the next few years she held ESL positions in 4th and 5th grades as well as a resource position. At the time of this study, she had been an ESL 3rd grade teacher for the previous 10 years. While having never taken any CS or IT classes, Teacher3 has participated in PD provided by the district and is confident in her ability to learn technical skills. Regarding her interest in this study Teacher3 expressed a desire both to learn CS and support the project:

108

I think it's great that you want to start this program. The earlier we start it in school, the better. Our children know how to navigate better than we do because they do it. And we have to get to the other things and this is a good way to get a motivation to want to learn and to do something fun.

Teacher3 was familiar with lesson study and looked forward to collaborating with other teachers on a CS lesson. Having participated in a science lesson study project with Teachers 1 and 6, Teacher3 had this to say about lesson study:

"Teacher1 and I and a couple of other 3rd grade teachers did a science lesson study in conjunction with the Northern RI collaborative and some teachers from URI". Here is how she described the lesson study project:

I think it's good...one person teaching it and everyone observing. And seeing, the first time it gets taught, these are the things that went wrong or these are the things that happened during that lesson, and to be able to tweak those things. Obviously not everyone is not going to be able to do it in a regimented way, or the same exact way. But I think it is a good process, to look back and say "oh, that's the way this teacher did it". Teachers1 and 6 and I did it with the science, and it was a lot of fun.

Teacher4

At the beginning of this study, Teacher4 was taking in her first regular classroom teaching position. Over the six years prior to this study she had worked as a substitute teacher in several schools before being hired as a special ed teacher at Study School. Because her mother had been a teacher for over twenty years she became interested in a teaching career at an early age. While having had no previous CS or Lesson Study experience, Teacher4 said she was looking forward to working with her peers and helping her students learn CS.

Teacher5

After taking her Master's degree in education Teacher5 taught fourth grade for eight years, sixth grade for one year and third grade for the past four years at StudySchool. As the only teacher in the study who had taken a class in computer programming, Teacher5 brought a unique perspective to the lesson study group. After earning a Bachelor's degree in business administration with a concentration in information technology (IT). Teacher5 turned to teaching because of her initial experience working in the information technology industry. She provided this perspective on her students and current teaching position: "Because we are in an urban district I definitely find that they are not your typical students. Educationally, third grade I feel is more like second grade, fourth grade is more like third grade, but life experience is far greater."

Teacher6

After receiving a Bachelor's degree in Psychology Teacher6 took a degree in Early Childhood Education and entered the teaching workforce. "I started as a

110

substitute teacher and quickly realized that early childhood education was not going to be as interesting as working with older (K-5) students", she said. After 7 years as a kindergarten teacher Teacher6 held a third-grade teaching position at StudySchool for 11 years before taking on her current position as STEM coach for the school district.

While she had no formal CS experience, Teacher6 has participated in "Hour of Code" workshops was supportive of introducing CS in elementary grades:

Based on what I saw in the Hour of Code, the kids really enjoyed the coding and working together so am looking forward to learning about Scratch Jr and seeing how that will go in the classroom. Since the students have had some experience with chrome books I am looking forward to seeing how students take to Scratch Jr and programming on a tablet. Wondering how confident they will be with it or how frustrated they may be with it. How will they feel? Will they think they know how to do this? Looking forward to seeing how the teachers teach programming with Scratch Jr. Teachers use the technology to their comfort level so the student experience with technology varies among students coming from second to third grade.

Pre-Study Interview Summary

The pre-study interviews provided me with a chance to see that the teachers were interested in providing something new and interesting for their students. I began to see how a tightly packed curriculum and focus on raising student achievement scores leaves little room in the schedule and thus presents a formidable barrier to CS. My initial impression was that the collaborative aspect of lesson study would work well for this group of teachers. However, I was concerned the time that they would

have to devote to this project would need to be effectively managed. Finally, because their schedules were full, CS could not be added as an extra course on the schedule. It would need to be "blended in" with the existing curriculum rather than "bolted on". This concept turned out to be a strength as teachers began to use the CS concepts to complement instruction in science, math and reading. The concept of blending in with the curriculum was further developed during the study as the teachers became proficient at using Scratch Jr and saw that it provided another way for students to "tell their stories'.

All teachers expressed a keen interest in learning CS and teaching CS to their students. While there are studies that show that lesson study alone does not guarantee collaboration among teachers, in this study all teachers welcomed the idea of a collaborative PD design.

One seemingly insignificant note that I wrote during the pre-study interviews proved to be important. I was aware of the term "professional learning community" but was unaware that the school district had allocated PLC hours in the teachers' schedules to promote collaborative practice improvement. During one of the pre-study interviews I wrote "PLC". I heard the term again from all teachers as we began to plan the lesson. I was told that the PLC was a districtwide initiative to encourage collaboration among teachers. Having time available for all the teachers to meet is an essential part of lesson study.

As we reached the fourth CS PD session and began to create a teaching schedule the teachers thought they could work with the principal to organize the PLC time so that they would have an hour each week to meet after each lesson study cycle.

112

This was significant, considering the tightly packed schedule. It also helped the teachers incorporate CS Lesson Study into their regular schedule once we moved to the lesson study phase. Aligning with the school district PLC initiative allowed us to work the CS lesson study meetings into the school day and made the project fit within the teachers' schedule. This part of the process was handled by the teachers who took it upon themselves to work with the school principal to re-arrange the PLC time to allow them to have a CS lesson study period of "Reflection" immediately after teaching the Wednesday CS lesson. Reorganizing the teacher PLC time made the project fit within the teachers' current workflow and allowed the group to refine and revise the lesson in accordance with the lesson study process. The overall implementation of the project was greatly enhanced by the adjustment of PLC time and because three of the six teachers had previously participated in a science lesson study project. Table 11 provides a summary of participant characteristics as of the pre-study interviews.

Table 11: Teacher background and pre-study CS Lesson Study knowledge

Pseudonym	Experience	CS	LS	Notes	Quotes
Teacher1	10 Years	N	Y	Teacher out of college.	I have always been interested in teaching and working with kids. Helping them out as best I can. Enjoying time with them. Being able to help them learn new things, and showing them ways to learn new things, that kind of thing. Basically, I enjoy just working with kids.
Teacher2	11 Years	N	N	ESL, Teaching is a second career.	With the jewelry industry shrinking, the company began downsizing. I took this as an opportunity to try something new and so signed up to volunteer in teaching to see if I had the temperament to be able to do it, and liked it.

113

Teacher3	15+ Years	N	Y	ESL, 10 years 3rd grade.	As far as computers are concerned, I don't consider myself "savvy". I consider myself, that I can figure out a few things. I'm one of those, you know, older learners that if you show me how to do it I'll figure it out.
Teacher4	5 Years	N	N	Special Ed, First year as a classroom teacher.	I started to be interested in teaching due to my mother. She was an elementary-ed teacher, for twenty something years, so I used to go help her set up her classroom. I used to like to go play with her things. Her planbook would go missing and I would have it, and things like that.
Teacher5	13 Years	Y	N	Teaching is a second career, worked in IT before becoming a teacher.	Way back, in the stone age, from what I remember, I did some programming in COBOL and DOS. I don't remember a darn thing about it and I never did anything with it.
Teacher6	18 Years	N	Y	11 yrs 3rd Grade, now STEM Coach for district.	On a whim, I just recently took on the role as STEM coach position for the district, K-3. I will be moving around at different schools. STEM K-3 is focused on Math, Eureka.

Notes: CS = Took a course in programming or IT. LS = Participated in a Lesson Study Project

Data Collection Phase II: CS PD Sessions

As the second phase of data collection, CS PD Sessions were designed to provide the teachers with the CS content and pedagogical knowledge that they would need to begin teaching CS to their students. This phase of the study was designed to help answer the first research question: How does CS Lesson Study influence instructional planning by elementary school teachers?

The term "instructional planning" is intended to suggest that, during the CS PD Sessions, teachers will work together to create a CS lesson plan while learning about computer science and computer programming. Accordingly, the CS PD Sessions

114

consisted of three parts: 1) computer science; 2) computer programming and; 3) CS

lesson planning.

Table 12 shows the syllabus and notes for CS PD Sessions.

Table 12: CS PD Sessions

Session	Discussion	Notes/Resources

8/7/16 6:33pm 45m online	❖ Welcome to CS!
	  - What is a computer?
	  - Why CS in Elementary School?
	❖ Learning to Code
	  - Introduction to Scratch Jr
	  - Scratch Jr Curriculum
	❖ Introduction to Lesson Study
	  - *Lesson Study Step-by-Step*
		1. Project Overview: Group website
		2. Review setup of Laptops
		3. Steve Jobs*: Bicycle for the Mind*
		4. Marina Bers: TED Talk – *Playgrounds, not Playpens*
		5. For next week: write your first program. Pair up, discuss Scratch Jr Curriculum Lesson1. Imagine teaching it to your students.

8/13/16 8:30am 53m online	❖ Algorithms & Program Constructs ❖ Scratch Jr: Instructions ❖ Lesson Planning: *Scratch Jr Curriculum Lesson 1*	1. Khan Academy: *Algorithms: Why do we need them?* 2. Slide presentation: Scratch Jr instructions. 3. Talk through teaching Scratch Jr Lesson 1.
8/20/16 8:30am 68m online	❖ Ada Lovelace, Charles Babbage ❖ Instructions: Sequence, Loops, Branching ❖ Research Lesson: Talking Points ❖ CS Lesson1 teaching schedule	1. Computerhistory.org: Ada Lovelace 2. Slide presentation: Developing algorithms using Sequence, Loops, Branching. 3. Scratch Jr in the classroom. 4. Catherine Lewis: Lesson Study
8/27/16 8:30am 60m online	❖ Binary computers & the Stored Program concept. Tubes, Transistors and Moore's law. ❖ Programming style ❖ Scratch Jr Lesson Planning: Reinforcing math skills with Scratch Jr. ❖ CS Lesson1 discussion	1. Konrad Zuse: DiscoveryHD 2. Von Neumann: Hoffman documentary. 3. Grace Hopper: Letterman interview. 4. Kathy Sierra: Video*: Code Like A Girl*
9/10/16 8:30am 60m online	❖ Computational Thinking (CT) ❖ Scratch Jr & CT ❖ Transition to CS Lesson Study phase: confirm teaching schedule; review lesson plan; schedule meeting at school to setup tablets and walkthrough teaching the lesson.	1. Jeannette Wing IHMC talk 2. Lesson Study step by Step Ch 2&3: facilitator, notetaker.

115

9/14/16	❖ Load example programs on Tablets	1. Tablets & case setup
3pm	❖ Review the lesson plan	2. Example programs
School	❖ Discuss observation protocol	

The CS PD Sessions were designed to facilitate transition to the next data collection phase by adjusting the ratio of time allotted to each part according to how well the teachers were assimilating the CS knowledge. In other words more time was spent on lesson planning as teachers CS knowledge and programming skills increased. Indeed, by the final CS PD Sessions the teachers had created their initial CS lesson plan, developed a teaching schedule for each class, and were working on the details of how the tablet computers would be managed during the lesson.

Computer Instructions

The CS PD Sessions begin with a presentation that helps the teachers understand that a computer is simply a machine that executes instructions one at a time. Because computers are amazingly fast these days it is hard to imagine that they grind away, one instruction at a time. Also, the concept of instructions is easily overlooked in the rush to get to something more cool. It is helpful to think of some computer languages as following an imperative paradigm where instructions look like short sentences beginning with a verb possibly followed by a noun or two. Scratch Jr is a language that follows an imperative paradigm. This means that the computer instructions represent statements that tell the computer to do something such as : add 4 to x, or subtract 4 from x. Figure 13 shows three Scratch Jr instructions that would be used to develop program logic using Sequence, Loops, and Branching constructs.

116

Move	Loop	Branch

Figure 13. **Instructions - Move right, Loop 2X, Branch**
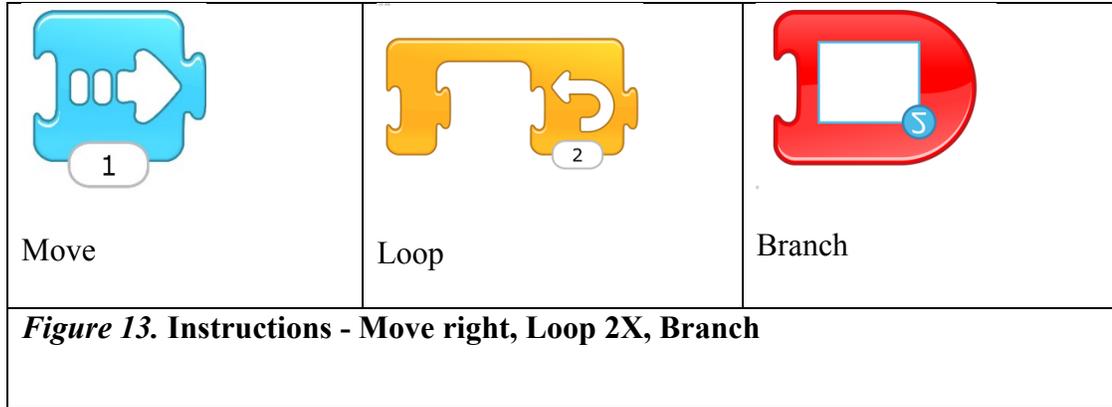
Program Constructs

Since a computer program consists of more than a single instruction, three *program constructs* are used by programmers to organize lists of instructions to accomplish a more complicated task. For this project the three program constructs are called: Sequence, Loops and Branching. Sequences of instructions execute one after another. Loops are one or more instructions that repeat, and Branching tells the computer to branch to an alternative set of instructions, such as changing to a new background. In computer programming the three program constructs are used to develop computer algorithms. Figure 14 shows examples of Scratch Jr programs using the three program constructs. Note the three single instructions from Figure 13.

117
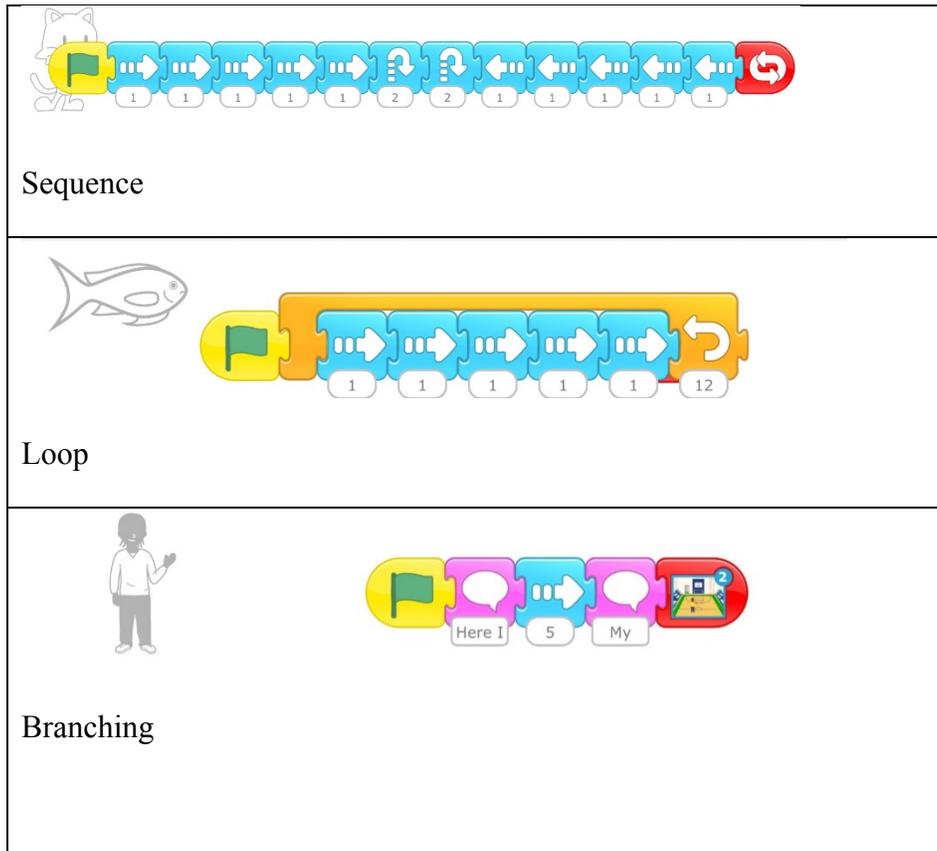
Figure 14. Program Constructs - Sequence, Loops, Branching

Algorithms

Algorithms are sets of steps followed to accomplish a task. Computer programmers use the three program constructs to organize their instructions in a given order to develop a computer algorithm to accomplish a task. The program constructs are emphasized so that the teachers will accurately convey these fundamental concepts to the students. Once we understand that the instructions tell the computer to "Do Something", and that a computer executes one instruction at a time, we can think about developing algorithms by organizing instructions within computer programs

118

using the three program constructs. A recipe analogy is used to explain the idea.

Figure 15 shows a slide from the CS PD Sessions that illustrates the recipe analogy.

Figure 15. **Computer programs are like recipes**

The CS PD Sessions were supplemented by other materials that the teachers used between the online sessions. In preparation for the first CS PD Session 1 the teachers had: spent time reading the Official scratch Jr book; watched a video of Steve Jobs explaining how a computer is like a bicycle for the mind, and watched a TED talk by Marina Bers explaining the philosophy of elementary grade level CS. Table 13 is an excerpt from the first CS PD session that intends to convey the idea that computers are machines that execute one instruction at a time, chosen from a finite set of instructions.

Table 13: Instructions - dialogue from CS PD Session 1

TN: So, we are going to talk about programming in Scratch [points to the scratch jr project] and what we have are instructions, and we are going to learn how to do loops, sequential instructions, and things like that.

[puts up the next slide with a cartoon of a chef with a recipe and an image that says "anatomy of Instructions, Instruction sets…"]

The whole idea about computers is that they follow instructions and the instructions are hardcoded into the computer. So, if you imagine a calculator, a calculator has Add, Subtract, Multiply, Divide, Square Root, those are instructions. A simple calculator doesn't have stored programs, so a calculator is just like a computer without the memory for the stored programs. Does that make sense?

Teachers: Yes, yeah, yes.

TN: All the instructions are hardcoded into the computer, so there are a finite number of instructions. So, think about a program as a recipe [points to the recipe cartoon] So here Instruction sets have a simple structure, the operation code is a verb, the operand is a noun. So just like the recipe we have move the flour to the bowl, add milk to the bowl, add egg to the bowl, mix the bowl.
Make sense?

Teachers: mmmHmm, yes, yeah.

Data Collection Phase III: CS Lesson Study

With the start of the school year, August 29, the group decided that the final CS PD meetings would focus on getting ready to teach the research lesson over five consecutive weeks, starting in October (Figure 16). The CS PD Session of 9/10 focused on arranging the schedule and making plans for setting up the 16 Android

120

tablets that the students would be using. The group also discussed formalizing the lesson study process. As is customary in lesson study practice, the group assigned a lesson study facilitator and notetaker to coordinate and document reflection meetings. The school principal had previously worked with the teachers to allocate time in their schedules for lesson study reflection after each of the five teachings sessions.

CS LS Classroom1 10/4 -10/5	CS LS Classroom2 10/11-10/12	CS LS Classroom3 10/18-10/19	CS LS Classroom4 10/25-10/26	CS LS Classroom5 11/1-11/2
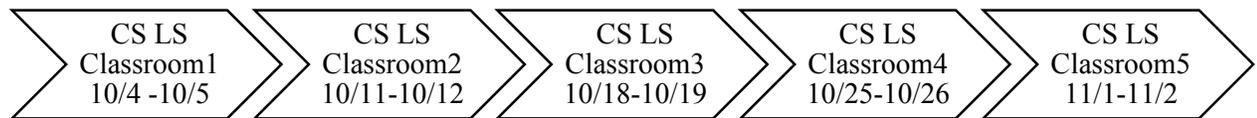
Figure 16. **CS Lesson Study Schedule**

As the third data collection phase CS Lesson Study was designed to implement the CS research lesson that the teachers had developed during the CS PD Sessions. By examining how the teachers taught the lesson in five classes the CS Lesson study phase helps answer the second research question: how does CS Lesson Study influence instructional effectiveness among elementary school teachers? By examining the students' work, teachers' observations and students' response to the CS lesson, the CS Lesson Study phase helps answer the third research question: how does CS Lesson Study among teachers affect student learning? Accordingly, data sources included: lesson plans; transcriptions of lesson study sessions; examples of student work, and observation notes.

CS Research Lesson

Lesson study is a process focused on improving student learning by improving instruction. While the process of lesson study is centered on a "research lesson", it is important to remember that "polishing" a lesson plan is not the main objective of

121

lesson study. Rather, the main objective of lesson study is to help teachers build content and pedagogical knowledge while creating a professional learning community and ultimately improving student learning (Lewis & Hurd, 2011; Norwich, & Ylonen, 2013; Yoshida, 2008). Like throwing a pebble in a lake, each cycle of lesson study creates ripples that can change the culture of a school, district or region (Lewis & Tsuchida, 2012). The following section describes how the lesson study cycle was implemented by the teachers in this study. Please see Appendix I for the Scratch Jr lesson plan that was selected as the first research lesson. Appendix II contains the lesson notes developed by the teachers through the five lesson study cycles. Appendix III contains examples of presentation slides created by the teachers. The four steps of the lesson study cycle (Study, Plan, Teach, Reflect) were repeated with each of the five classes.

Lesson Study Cycle Step 1: Study

The first step in the lesson study cycle involves considering the long-term goals of student learning, studying the existing curriculum, and identifying topics of interest. In this step teachers would either choose a new research lesson or adapt the research lesson from a previous cycle. In this study step 1 also included identifying how different activities in the CS lesson reinforce learning objectives. For example, in the first cycle of lesson study the teachers decided that the teacher conducting the lesson should wear large cardboard cutouts of letters "L" and "R" to help students who might have difficulty following instructions, for left and right, during the Simon Says and Program the Teacher exercises.

122

Lesson Study Cycle Step 2: Plan

In this step teachers discuss how the lesson would be taught in their classroom. By examining every aspect of the lesson the teachers identify areas of concern, predict questions from students, and opportunities to improve student learning. In this study Lesson 1 from the Scratch Jr Animated Genres: Curriculum Module 1 was selected as the research lesson (Appendix I). Lesson 1 introduces foundational concepts of CS and coding through a series of activities and coding exercises using Scratch Jr. Table 14 shows the different parts of the Scratch Jr lesson.

Table 14: Research lesson plan - Scratch Jr Lesson 1

Activity	*Description*	*Learning Objectives*
Introduction	Teacher explains why the students are learning CS and asks what they know about computer programming.	- The concept of programming
Simon Says	The teacher leads a game of Simon Says and discusses how giving and following specific instructions is critical to computer programming.	- The concept of following instructions - The concept of sequencing
Program the Teacher	In this activity the students give verbal instructions to direct the teacher to a given destination in the classroom	- The concept of giving instructions
Classroom Rules	The teacher hands out tablet computers and explains how important it is to respect each other and the equipment in the classroom.	- The proper use of computer equipment
Scratch Jr Programming	The teacher demonstrates how to create a project and move Motion Blocks	-The features of Scratch Jr - Creating a project

123

	(instructions) to the programming area. The students duplicate the task.	- Motion blocks (instructions)
Exploration	The teacher asks the students to try adding different instructions to get the Cat to move.	- Selecting block categories - using different instructions
Wrap Up	The teacher demonstrates how to save a project and collects the tablets.	- Saving a project

In addition to promoting familiarity with Scratch Jr, the lesson is intended to provide an understanding of two foundational programming concepts: instructions and sequencing.

For each cycle of lesson study the planning step involved adapting the research lesson in accordance with the teacher's evolving CS knowledge. Table 15 provides a lesson outline and notes that document the changes made to the research lesson through the five Lesson Study cycles.

Table 15: Evolution of the research lesson through five CS Lesson Study cycles

Activity	*Enhancements to the original lesson plan*
Introduction Day1	1. Pairing students with their writing partners - collaboration 2. Talking points for consistent dialogue. (Appendix II) 3. Added the Magic School Bus Story on the day prior to teaching 4. Steve Jobs analogy to explain that a computer is a tool, *like a Bicycle for the Mind*
Simon Says Day1	1. Cardboard "R" and "L" on teacher's shoulders. 2. Dialog added to talking points to ensure consistency.
Program the Teacher Day1	1. Using Pre-printed instruction forms to have students write their instructions. 2. Added Arrow symbols to form to help ELL students 3. Limit of three instructions, "forward, right, left" to help students develop problem solving and CT skills.
Classroom Rules Day1	1. List of student pairings to help coordinate distribution and retrieval of tablets. 2. Step by step demonstration of creating a Scratch Jr project.
Scratch Jr Programming Day1	-The features of Scratch Jr and an activity the teachers and students work through: creating a project, selecting objects, selecting a background, and programming.
Exploration Day2	- The teachers expanded the scope of this lesson to allow the students to create their own "stories", using programs they write with Scratch Jr.
Wrap Up	The wrap-up included the students' giving a presentation of the stories to the class.

Lesson Study Cycle Step 3: Teach

In this step one teacher teaches the research lesson while others observe and take notes. In this study the group felt that it would be impractical to attempt to coordinate the five teachers' schedules to allow them to observe all the teaching

125

sessions. The teachers made a change to the traditional lesson study Step 3 because not all teachers were able to observe all of the teaching sessions. Instead, in addition to the teacher teaching the lesson, all teaching sessions included the classroom teacher, an assistant (either Teacher6 or Teacher7) and the researcher. The other teachers in the group attended during parts of the teaching sessions if their open time corresponded to a teaching session. As mentioned previously, the school principal helped organize the teacher schedules to allow them to hold "reflection" meetings immediately following the Day2 teaching session for each of the five cycles.

Lesson Study Cycle Step 4: Reflect

Lesson Reflection is the step of the cycle where teachers meet to evaluate the lesson and students' response to it. This step is usually completed immediately after a teaching session. In this study all the teachers met immediately following the teaching session on Day2. During the reflection meetings one teacher acted as the facilitator of discussion while another took notes. The notes were circulated to the other members and the school principal. Reflection sessions were held in an empty classroom after school. They were open to other teachers, such as the reading specialist and physical education teachers, who were interested in hearing how the students did in their CS lessons. The reflections served to provide data on student learning, instructional effectiveness and CS content knowledge. They also helped build confidence in those who had not yet taught the lesson. Table 16 provides a summary of the CS Lesson Study Teaching and Reflection sessions.

126

Table 16: CS Lesson Study schedule

Session	Description	Reflection notes:
10/4, 10/5	Instructor: Teacher1 Assistant: Teacher6 Students: 19	Day1: Opening was good. Students learned the vocabulary and about CS careers. Lesson sections and timings were good. Students asked questions and responded to questions. Day2: Very good programming projects by the students. One group programmed soccer game with audio "goooaaal!" Others began asking how they did it and then many used audio instructions. Need to reinforce the goal to write instructions for each character.
10/11, 10/12	Instructor: Teacher2 Assistant: Teacher6 Students:	Day1: Due to a late start, they did not get the tablets on Day1. The opening CS description went great, vocabulary words went well, kids understood vocabulary words. No changes needed to opening. Kids were so excited; with more ELL kids, need to keep pace of lesson steady and keep repeating the instructions. Simon Says went fine except there were many who did not know right and left. Use a cutout R and L on each shoulder for *Simon Says* and *Program the Teacher.* Day2: Seems like they forgot everything they learned yesterday. Let's try to find a child-friendly book on what it is to be a computer scientist. Also, provide the demonstration of writing Scratch JR program before handing out tablets. Kids got distracted by numbers of characters and settings. Need to focus them on choosing 2 and writing their programs. They were very interested in making up stories. We do need two days to teach this lesson; one day is not enough.
10/18, 10/19	Instructor: Teacher3 Assistant: Teacher6 Students:	Day1: Change to the lesson plan from Classroom2: On the previous day (10/17) Teacher3 read the kids a story about CS and Programming *The Magic Bus Gets Programmed* Teacher3 held back tablets until she had gone through all preliminary material.

		Day2: Many ELL students in the class enjoyed learning about Scratch JR and were engaged working with their projects. Teacher3 mentioned how creative the kids were. Two girls chose fairy characters and added photographs to the fairies' costumes. Teacher3 taught them about loops. Teacher3 says how it went by really fast. Group discussed timing of the different sections and whether any parts can be removed to allow more programming time on Day2. Students shared and collaborated very well. Teacher3 adapted the Steve Jobs analogy *Bicycle for the Mind* and told the story about walking to the evacuation park while someone else rode a bike. Group agrees to emphasize the concept that a computer is a tool. Group feels this version of the lesson plan is solid.
10/25, 10/26	Instructor: Teacher4 Assistant: Teacher7 Students:	Day1: Teacher4 showed the Magic Bus Gets Programmed video on the day before. This helped the students understand CS and programming before Lesson1. The Intro went well. Teacher4 embellished the opening dialogue by describing how the daily routine is like a set of program instructions. Teacher4 emphasized that on the next day (Day2) the students would be learning to program by making clear decisions on which Scratch Jr characters to choose, selecting a background, and writing the program for each character. Emphasis on choosing two "Objects' and writing. Day2: The student exploration was focused and their projects showed that they were listening to the teacher and following the guidelines. Many different characters, scenes, and instructions were used by the students in this session. As in the previous classes, students were enthusiastic and creative. The group noted that the Scratch Jr CS exercises require students to use literacy and math skills. The group decided to adapt the Day2 Exploration to emphasize the literacy connection by having students work with their writing partners to create a narrative story in Scratch Jr that they would present to the class.
11/1, 11/2	Instructor: Teacher5 Assistant: Teacher6 Students:	Day1: Teacher5 also showed the Magic Bus Gets Programmed the previous day. Teacher5 created a slide presentation to organize and display all of the CS concepts in the first parts of the lesson. Day2: Teacher5 used the second day to have the students work at developing Scratch Jr projects that told a story. Before they began to program the teacher asked students to

organize their stories using preprinted forms that included the name of the story and sections for a beginning, middle and end. The data showed that the students were learning how to program, worked well with each other and had fun presenting their projects.

Data Collection Phase IV: Post-Study Interview

Post-Study interviews with teachers were conducted from November 9 through 19. The focus of the interview was to hear the teachers' perspectives on the three constructs of interest and their view on the viability of CS Lesson study in their school and school district. This section provides a summary of the key points made by the teachers during their post-study interviews. The next section will include excerpts of interview transcripts bearing on the constructs of interest. The following summary provides the post-study perspective of the teachers in their own words.

Teacher1

Ok, well before I didn't really know what to expect. I didn't know anything about computer science or computer programming. I just know how computers work. And getting the [information] from you I began to really think about what things have computers, and what different programs are used for different things like radio, TV, and thinking about it a little more.

And as I was trying to start to think about teaching it I was getting a little nervous. I was getting a little bit better idea about what computers are but I didn't know how I was going to go about teaching it. But then as we started to work through the book and planning our first lesson I started to become a

little bit more comfortable about what computer programming and computer science actually is.

So, as we started planning it out I started thinking that it was something I could actually do using the Scratch program. Before we really got into it I didn't think I would be able to do it. But once we started getting into it, it gave me an understanding of what is actually involved in computer programming and what we have to actually do to show the kids. So it helped me get to understand what it is and get more comfortable actually trying to teach it.

When it was time to teach it I was a little nervous because I had never done it before. But after doing it once I definitely feel a lot more confident and would be comfortable doing it with another group of kids, another year or another time.

Teacher 2

I think that the job opportunities, giving them access to something they are unfamiliar with….And to know that it's not just "oh you're learning math, oh you're learning science" it's that you're really create, do these problems, and solve problems in different ways. Not just doing math, not just doing the science test, that there are other ways that you can incorporate problem-solving and collaboration. To get them ready for the future and their next grade. "

Being in an inner-city school I have learned not to make a lot of assumptions about what the kids know. If I am unsure of what they know I try to be sensitive and put it out there like, so you have heard of…?

For me when I was doing Scratch Jr it was like a game, so it's fun. I am still kind of interested in seeing how this would lead into Java or C++ , real coding. I think that at this point with Scratch Jr it's good for the kids because they actually have to slow it down and they can't make assumptions.

Like we were just talking about, we can't make assumptions. I think this is one of the few times that they have to, um I mean we are always telling the children you have to be able to explain it, you can't just say, " I dunno I knew it" No, you have to be able to explain it, you have to be able to verbalize it, or you have to be able to write it down.

They always look at us like we have six million heads , like "why would we have to do that?". And I think even just the Scratch Jr, it really slows them down. I think on a subliminal level they are realizing "oh, I really have to think of all of the steps that I am doing. But like I said I don't know how we translate that to the bigger picture like C++, or Java, or Linux, or any of those.

And for PARCC [Assessment test: Partnership for Assessment of Readiness for College and Careers PARCC] even pre-PARCC, the whole idea was to have them explain their math, they have to explain their thinking. And that has always been the most difficult thing to get any of our students to do. Because they just want to say "I added it". You know that's not what they [PARCC] are looking for; they are looking for "I decomposed the numbers and added my tens and then added my ones".

It's the things that the kids actually do, like the kids do it, but they don't know they are doing it. They don't have that metacognition, they should.

You know, and some of them do, but there are very few who would be able to say "oh, well I decomposed, I used a number line, Oh I…" you know what I mean, to explain how they got their answer. They'll just say "oh I added it" and I will say "well, of course, you added, that's what I told you to do!" …because they can't just say "I moved the basketball", they would have to think about how they are going to move it. "Oh I am going to move it forward six steps, then I am going to move it.." you know it makes them have to think more about the sequence and the steps. I am really looking to milk that connection as much as I can. They just haven't been able to code often enough that we'd be able to impress upon them - look, how many times did you have to rewrite that code before you got the ball into the basket? How many times did you have to rewrite that? I think that sort of connection can, you know, I'm looking forward to that.

I think it was better than what I was thinking. I thought they were just going to move things around. You know I was thinking when I first started doing it I think I had more frustration than they did. Like when I was doing it I did not figure out the reset. So every time I started to play it , it kept not doing what I wanted it to do. You know it kept starting at a different point, you know it was driving me crazy, until I finally figured out the reset button. And I honestly did not see that frustration with them, which was great. I thought that was great that they didn't have even the littlest bit of frustration that I saw. So that was surprising and really great. I'm sure there was some frustration but they didn't let it get in their way.

www.manaraa.com

It's interesting though that it's like a language. Computer science is like a language we know that languages are best when they are learned young. So that's been the U.S.'s problem all along, that they wouldn't begin teaching languages until middle school, when that window is already closed.

The thing I like, being an ESL teacher, is the fact that you don't need a language skill to be able to do this. It would make them feel successful. It would be giving them a voice. You know what I mean? It would be giving the ESL students a voice that they would otherwise not have. It would give them a showcase where they could show their thinking and thought processes. They could tell their stories. And the whole collaboration part of it. I think it is really ideal for English language learners.

Teacher 3

It worked very well. It's baby steps because they know they want to get in there and play. It's hard because they are eight and nine. We had to really think about it because we thought "well we can't really give them the tablets until we absolutely know what we are going to do". And we saw even when we went through that we saw some of the kids just went all out exploring and they didn't have any instructions for their objects. They had lots of characters, lots of items.

You have to remember that there is a big difference between Teacher 1's class and my class. I have a large group of students who are both ELL students but they are also special ed. So a lot of things that they do have to be modified. They have to have the directions told to them more than one time.

133

Then you have to go through and have them "tell me what you are going to do". And so a lot of what we saw, I tried to put a strategic plan that put kids together who could help each other. But they need extra help like having the directions repeated and reminders like "I was going to do what??" When they lose that there is a little misfiring of comprehension as opposed to, they know what they want to do.

I found the book "The Magic Bus Gets Programmed" and I thought that might be good just to make a connection with the concepts that we wanted them to learn. And for them to see that, with programming, things can go awry. To even reiterate what we do with the "program the teacher" exercise. Even though I didn't carry that too far, like I didn't have all of them share [the Program the Teacher instructions]. I knew there would be a lot of issues with that. A lot of these students can't follow two directions at once. But it was nice to see them go back and reevaluate what they had. Like I had one student who went back and said, "Oh yeah, wait a minute, I can't go that way with it I have to go this way". And I added these [pictures of Right, Left, and Forward arrows] to the written instruction sheet to help them.

To help explain why we use computers, Teacher3 adapted the Steve Jobs analogy that a computer is a bicycle for the mind. This became part of the lesson plan. Here is how she explains coming up with the idea:

I went and looked at the video again. I listened to Steve Jobs explain it again. And I thought "oh, ok so we are really saying that the computer is a tool to help us do something better". So how can I relate that to them? Luckily we

had just had that evacuation drill. And they know they have to walk that far. So I was able to use that.

Teacher 4

It was good being able to work together. Not like PD sometimes, where you go and focus on different areas and no one goes to the same PD. It was good that we were all able to relate to it. We would talk about it in the morning, or after school or sometimes during lunch. It was good that we were all working together; we weren't alone on that. We worked as a team. I feel that I learned something; the kids learned something, and I had that extra support and guidance. This was my first year as a classroom teacher. I was a special educator before, so it was good to have that extra support. I'm on a good team that helps me out.

Ok well I didn't have that much knowledge of computer science at all so this process did help me to learn more. I did like how we broke it down and we focused on certain areas rather than altogether at once. We broke down the lessons and things like that. I liked that we all worked together as a group. They helped me a lot. Especially since I was one of the last to teach the lesson. So each time a teacher taught the lesson I did speak to them and they did give me some input and I feel that was very helpful. I think if I were alone on this it would have been a bit harder. I feel that it was easier to get through it all together. So I did like that. That we were able to work together and collaborate

My concerns for the first lesson was.. I was nervous. I had never taught anything like that so, you know, not as much experience so I was nervous

135

about how it was going to go. I feel that the students did a pretty good job understanding. I was a little bit concerned for the vocabulary, computer programmers and things like that. So I tried to find a way that they were able to relate to it. And I feel that the classroom schedule and how I talked about how we have routines in our own classroom was able to help them to relate to the vocabulary and the whole process.

As the students were getting a better grasp on what I was teaching I started to feel a little bit more confident in that area. I was a little bit nervous when I first started teaching the Scratch (programming) on the first day, technology problems and things like that. But it went pretty well and I already had an idea that other teachers had some bumps along the road so I was kind of expecting it, so I wasn't too nervous about that. It went pretty well the first day, once the vocabulary and things like that were over I felt that it was a little bit easier. And the second day I felt pretty confident and kept going over some of the things that the students would be taking time to work on the tablets during that time.

[Did you feel the students were learning?] Yes I did, because I did say, the second day before I handed out the tablets, I did ask them to tell me what they learned the previous day. And I did have students giving some examples of what computer programmers are and definitions and things that they could relate to such as the classroom routine, and things that we have in our classroom. And even to this day there's a Friday where they earned all their [points] that they needed. So, for an incentive they chose to go on the

136

chromebooks. So I let them play some games online. And some of the students were saying "oh I wonder how this one works, how this one would be programmed?". They were starting to think about it a bit more, they weren't just playing the game, they were actually asking me questions about it and things like that. So I did feel they did learn something from the two lessons that I did teach them.

I think they loved it. One of my students - he was the top one with the Scratch JR. It gave him a chance to relate to the other students. It wasn't like a math test or involved reading or anything like that. They were all equal and able to express themselves. So I think they all did a great job with it. No one was excluded or anything like that.

Four of them did say that they were able to put Scratch Jr on a tablet. I know one of my students, he already had Scratch Jr before and I didn't even know about it. So he already had that at home and he told me that he was already working recently on Scratch [Scratch Programming Language], he said he'd moved up to that. So I did have three other students who said that they had put it on their tablets. And I did have students say that if they had tablets or if they were to have the technology of computers that they would practice it at home.

I feel it has helped me out as well. I didn't know that much before so I feel I have a better appreciation for computers and how things work; I didn't really think about it that much so now I have a better appreciation of it. And it gave the students an opportunity to learn about computers. They usually just

137

take it for granted and go onto it. Now they kind of think about it like, "Oh, we are having a problem with the internet" or "Oh, we are having a problem with this website and I wonder why, I wonder what's causing that, I know how to problem solve I won't get discouraged, just click out of it and try it again". So I feel that they have been doing that more.

Like before they would get frustrated, now they are trying to solve the problem. Because I kept telling them that's what computer scientists do; they keep going back and try to solve problems, so they have been looking at that. I think the whole third grade, I know the kids would all get excited about this and know when it was their class would be working on it or their class would be next. I would like to see it go on to the fourth grade, if possible. I think it would be a good opportunity and from there our school would have more exposure to computer science. I know in our area not a lot of students go home and have the technology to be able to use those resources. So I think it is a good thing to have and provide it to them in the classrooms, if possible.

Teacher5

I have to say I was a little apprehensive about how this was going to go with the kids because they've had very limited exposure. I know a lot of the kids don't even have computers at home never mind programming. I knew they would be excited about it and I thought it was great. They took to it. And it definitely showed that they can work together. You know, collaborate, stick with it. I saw there were quite a few of them who wanted to keep going with it because by the time we were ready to show their projects they wanted to keep

138

going, they wanted to fix what was wrong with it. Which is nice because a lot of the time with their schoolwork they don't want to stick with it, they just want to be done with it.

So I thought it was, it turned out much better than I thought it was going to be. Yes, it was awesome. I think it was something refreshing for them. You know - as opposed to the rote math, and reading and writing. It was something that they could relate to but they didn't know they could relate to it.

[regarding previous career in IT] Well my experience was obviously very different. I didn't really get into it until college and was really overwhelmed and it was very, from what I remember, very abstract. And I think what we did with the kids definitely brought it to their level, and how cool this could be, instead of how I felt in college. It was very - for lack of a better word, boring, mundane. You know you are just staring at code, where I thought this was something I wanted to do. Where I think that with these kids, starting with Scratch Jr was just more entertaining them, enjoyable you know. And that you can actually have fun. With these kids, you know the whole idea, like Minecraft - I have no idea what that is, I've seen it but it seemed to be from what some of the kids were saying somewhat like that , and it definitely caught their attention. It wasn't just, from what I remember putting in words into a computer, trying to get it to do something. They actually could see things happening as they were programming.

Teacher 6

When you first met with us back in the spring, I was nervous about how the kids were going to respond to doing it. But at the same time I had been doing the hour of code with some of the kids so I knew that they could, the group that I had, persevering through even when things didn't work out, things like that. I was leery of, you know, if we were really going to be able to take that on and get that to go with like twenty-five kids in the classroom. But I was also concerned about the time constraints of like did we have enough time to do that?

But I think that, from my perspective, just coincidentally that I became the STEM coach at that time and that I had the flexibility to see most of the lessons and kind of support the kids and the teacher. It kind of made me feel like wow! they really, kind of almost like an outsider, they really can do this. They can persevere, they can listen to those directions and say "ok, I can do this".

And like you say about Teacher5's class [the final class of the series] once we fine tune the lesson enough to say, let's look at the setting, let's think about the stories, and characters, and what they're doing in this scene. I really saw how it could be really purposeful, meaningful, and almost like blend into the day-to-day curriculum without it being considered like a separate entity to everything going on in the room.

So it really made me feel much more comfortable with it. I wish in the bigger picture that the district would be more open to doing more of these

things. I think that it would be great to get others to see what the kids are doing. Mostly once we move into making it into stories, making it into stories and settings. I think it would be really interesting to say, ok they can really do the writing and they can now take this into technology and move their characters around on the tablet, in the coding program. I really think that's a neat piece that would really engage the kids more. So I'm excited.

CHAPTER 5

Summary of the Research

The purpose of this study was to assess the effectiveness of a CS PD activity that used lesson study as a vehicle to introduce elementary grade-level teachers to fundamental concepts of CS and computer programming. Lesson study is a collaborative PD process that teachers use to construct knowledge about student learning in a specific educational context. Lesson study is based upon the simple premise that if you want to improve teaching, the most effective place to begin is in with a lesson in the context of a classroom (Stigler & Hiebert, 1999).

In this study five classroom teachers from StudySchool, one STEM coach from the school district worked with a CS instructor (the researcher) to develop CS skills and plan a lesson in computer programming for 104 students in five third-grade classes. Over a three-month period the CS Lesson Study group examined CS concepts and systematically planned a single CS lesson that was taught over a five-week period beginning in early October 2016.

During the first week, the researcher met with participants at a local public library and explained the purpose of the study, distributed equipment and materials and collected data through pre-study interview. Each teacher received a laptop computer, an Android tablet computer, and two textbooks, "The Official Scratch Jr Book" (Bers & Resnick, 2016) and "Lesson Study Step-by-Step" (Lewis & Hurd, 2011).

Participants having agreed to participate and sign a consent form, the researcher conducted an interview with each teacher. Data drawn from the pre-study

interviews showed that all the teachers were interested in learning CS; one teacher had previous experience in CS, and three were familiar with lesson study. Initial analysis also showed that four of the five classroom teachers had taught third grade for several years and one teacher was teaching a third grade class for the first time. Following the initial meeting with participants, the researcher scheduled a series of online weekly video conference meetings and a Wikispaces website that contained supplementary materials.

Over several weeks, prior to the start of the school year, the researcher and participants met online to discuss fundamental concepts of CS, computer programming and to plan the CS lesson. During the first three weeks of the school year the teachers met online, with the researcher, and at the school to continue planning the lesson. The teachers planned a five-week schedule, beginning in October, to teach the CS lesson to each of the five third-grade classes. The lesson was taught in two one-hour units on the Tuesday and Wednesday timeslot reserved for Science. The teachers worked with the school principal to coordinate their "common planning time" so that they would all be available for a CS lesson reflection meeting on the Wednesday after the lesson.

Conclusions

The pre and post study interview, field notes and audio recordings gathered in this study answered the research questions and suggest that CS Lesson Study may provide a high-quality PD activity for elementary schools. The online CS PD Sessions provided a convenient forum in which the teachers could work with a CS instructor to

143

acquire fundamental knowledge of CS and computer programming. The CS Lesson Study Sessions offered evidence that the teachers were highly capable of implementing an ongoing, collaborative, student-centered approach to teaching CS.

The findings on RQ3 seem to support previous research that shows CS and computer programming among elementary grade students has potential to facilitate deeper understanding in other disciplines such as art, science, mathematics, and literacy (Bers, 2010; Clements, 2002; Sarama & Clements, 2009).

The five classroom teachers in this study all expressed positive views of their experiences in the CS PD and CS Lesson Study sessions. In the post-study interview each teacher expressed confidence in their ability to plan and teach the CS lesson.

As a prolonged PD activity, CS Lesson Study seemed to provide a high-quality type of PD described in the literature (Garet et al., 2001). Lesson study is centered around improving teaching practice by focusing teachers attention on the short and long term learning objectives of students. During classroom CS activities in this study the teachers guided the students through a series of activities designed to encourage teamwork an creativity. It is fair to say that the teachers and students learned from each other as they worked through the CS activities. The child-friendly programming environment, provided by Scratch Jr, seemed ideal for ELL students. Because the teachers had students work on CS projects with their reading and writing partners, the students were able to organize themselves and create their Scratch Jr projects within a 30-minute time period on the second day of the lesson.

The goal of this study was to provide a number of valuable contributions to the lesson study research community and CS education reform literature. Firstly, this

144

study adds to evidence that lesson study can provide a high-quality PD activity that can be adapted to introduce CS to an urban elementary school in the US. Secondly, the study showed that CS Lesson Study can help teachers plan and teach effective CS lessons that seem to enhance teaching and learning in other disciplines. Thirdly, the findings show that teachers working with a CS instructor can acquire CS content and pedagogical skills that allow them confidently to teach CS to their classes. Finally, the study shows that students in third grade seemed eager to work together to learn about CS and computer programming.

Recommendations

Launching a CS Lesson Study project requires a significant investment of time and effort on the part of the CS instructor and teachers. Because CS and computer programming are unfamiliar topics to the majority of people outside the technology industry ample time must be allowed over several months prior to the start of the PD activity to familiarize the participants with CS and set expectations. Follow-up communication with participants showed that after the first CS lesson students wanted to do more with CS and computer programming. In response the teachers have developed a second CS lesson and are including CS in a "Curriculum Night" with parents, teachers and students. Looking beyond the present study, the researcher hopes to continue to work with the teachers and principal at Study School to add CS Lesson Study to the fourth and fifth grade PD activities for teachers.

Implications

CS Lesson Study seems to offer a practical teacher PD activity that US elementary schools can implement using existing resources. Given results of this

145

study, CS Lesson Study offers a potential approach to engage the CS experts and elementary school educators in a positive, student-centered activity that provides access to CS to all students. Finding a sustainable approach to teaching CS and computer programming to all students starting in third grade has significant implications for students, parents, schools and the future of the technology workforce. For students, learning CS and computer programming may enhance learning in other disciplinary skills by helping them learn a systematic approach to problem solving. Parents recognize that CS Lesson Study at elementary school offers opportunities for their children to develop skills that they can use to succeed in tertiary education and ultimately in successful careers. For schools, CS Lesson Study has the potential to provide a way to establish a professional learning community focused on skills that can enhance learning in other disciplines and have a positive influence on student achievement.

Limitations to the Study

While this study seems to provide substantial evidence of the potential of CS Lesson Study, there are several limitations that need to be acknowledged. Firstly, while there is a significant body of research on lesson study in elementary school I have been unable to find research focused on lesson study and computer science PD for teachers. Secondly, while the findings in this study seem to suggest that the PD activity had a positive influence on CS lesson planning and effectiveness of CS teaching, it is unclear whether the PD activity itself or the other factors such as teacher characteristics, or both, had greater or lesser influence on the results. Thirdly, it is entirely possible that a single teacher working alone could learn Scratch JR and the

fundamental concepts of CS and teach the lesson to a third-grade class. Indeed, the current approach to CS education reform for elementary schools in Rhode Island is focused on single teacher workshops (CS4RI, 2016).

Additional limitations may include tactical and organizational issues involved in launching CS Lesson Study. For example, a significant level of cooperation and coordination between the participants, the school principal and the school district was evident throughout this study. It is likely that, without this cooperation and support from the learning community at large, the findings would be very different. Most important is the coordination required to organize the teachers' time schedule to allow them time faithfully to follow the lesson study process. For lesson study to succeed teachers need time within their schedule to hold meetings and observe lessons. In this study the PLC time allocated by the school district allowed the CS Lesson Study group to implement the project as planned. The initial plan, before we became aware that the PLC time was available, was to hold the reflection meetings online on Saturday mornings. It is evident in the data that in-person reflection meetings held immediately after each lesson cycle proved very effective in refining the lesson and fostering collaboration among the group over the five weekly cycles. It is unknown whether the online forum for reflection meetings would have yielded similar findings.

Need for Further Research

Glaser and Strauss (1967) proposed grounded theory as a method of research that would allow a sociologist to generate a theory based on evidence gathered through fieldwork. The formation of substantive theory is made credible by a strategy of studying additional comparison groups (Glaser & Strauss, 1965). Multiple comparison

147

groups maximize the credibility of a study and lift the burden of delimiting the boundaries of the theory from the reader. Only theories that have undergone rigorous systematic examination through multiple comparison groups would be considered substantive (Glaser & Strauss, 1965). Glaser and Strauss (1965) offer further guidance for grounded theory researchers by distinguishing "rigorous findings" from "credible theory". According to Glaser and Strauss (1965) the plausibility of a substantive theory may be enhanced through more rigorous or extensive fieldwork, depending on the research situation. Without multiple comparison groups a substantive theory of CS Lesson Study cannot be offered. Instead, the findings drawn from data using grounded theory methods are presented in general terms and with supporting evidence, on the assumption that readers will discount aspects of this study that do not apply to a different school situation.

Because innovations in computing are moving forward at a rapid pace, our children will likely inhabit a very different world from our own. The implication for future research is that new approaches to CS education reform need to be considered, including studies that examine specific curriculum materials and longitudinal studies that follow groups of students who have CS through K-12.

CS Lesson 2

The question of sustainability is a concern for CS reform. In the weeks following the study the teachers developed a second CS research lesson ("CS Lesson 2") on their own that builds upon the skills that were taught in CS Lesson 1. In CS Lesson 2 students work with their writing partner to develop a story. The students start by picking 2 Scratch JR characters and 2 settings. Next they use pre-printed forms to

148

compose a narrative with a beginning, middle and end. One they have agreed on the storyline they work together to write the code on their tablets.

While CS Lesson 1 emphasized basic concepts, CS Lesson 2 has the students use all three program constructs, sequences, loops and branches to develop their algorithm. The lesson wraps up with a presentation by each group of students, who tell their story while showing their Scratch JR program using the projector. Because the students already knew how to operate the tablets and write Scratch Jr programs, CS Lesson 2 fit within a 45-minute unit of time. Teacher1 invited me to stop by and observe the class. I found the students excited, engaged, and having fun learning CS. More recently I have received communtiation that the school is including student Scratch Jr projects at "Curriculum Night". This is a hopeful sign that, like throwing a pebble in a lake, CS Lesson Study is creating ripples that could change the culture of the school, district and region (Lewis & Tsuchida, 2012).

APPENDICES

Appendix I

This is the original CS lesson that was used as the research lesson for this study. Many

thanks to Marina Bers, Mitch Resnick and all the developers of Scratch Jr.

ScratchJr — Animated Genres Curriculum Module 1
Lesson 1: Instructions, Sequencing, and an Introduction to ScratchJr

Lesson 1: Instructions, Sequencing, and an Introduction to ScratchJr

Summary

In this lesson, children will be introduced to two concepts that will create a foundation for understanding programming: instructions and sequencing. Through various interactive activities, students will acquire a basic understanding of these two concepts. The lesson will conclude with an introduction to the ScratchJr interface.

Objectives Students will learn...	Objectives Students will be able to...
- Appropriate iPad use - The concept of programming - The concept of instructions - The concept of sequencing - The basic features of the ScratchJr interface	General - Give specific instructions - Sequence instructions to achieve simple objectives ScratchJr - Move blocks into the scripting area - Use blocks in scripting area as buttons - Select a block category - Save a project

Programming Blocks Introduced in this Lesson

- Right
- Left
- Up
- Down
- Bigger
- Smaller
- Visible
- Invisible
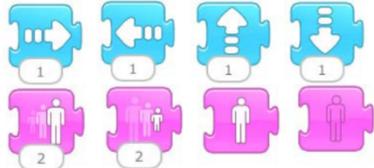
Additional Materials: Rule board

Schedule

Introduction (2.5 minutes): The lesson should begin with the teacher introducing him/herself to the class. The teacher should explain why s/he would like to teach the students about programming. S/he should briefly ask students what they know about programming.

Simon Says (10 minutes): The teacher should play Simon Says with the class. S/he should discuss how this activity is dependent on properly being able to give and follow instructions. S/he should then explain how providing clear instructions is critical to computer programming.

Program the Teacher (15 minutes): In this activity, students will be responsible for verbally directing their teacher to special destinations in the classroom (e.g. to a bookcase or a closet). The instructions the students give to the teacher must be specific. For example, students should

150

not simply say, "Move forward." They should instead say, "Move forward ____ steps." When sequences of instructions do not work (perhaps because the number of steps taken were incorrect), students should alter their instructions. After the activity is over, the teacher should discuss how important it is to be specific and how important order is in programming.
2nd grade: Small groups determine a sequence of instructions
Kindergarten and 1st grade: As a class

Classroom Rules (5 minutes): The teacher should explain to students how important it is to respect each other and the equipment used in the classroom. With the students, s/he should create a list of classroom rules governing iPad use. The teacher should write these rules down on the rule board, and hang these rules in the classroom every time the class is working with ScratchJr.
Materials: Rule board

Getting Started with ScratchJr (2.5 minutes): The teacher should **hand out the iPads** to the children, and show them how to begin a new project in ScratchJr.

Using ScratchJr Blocks (10 minutes): Everyone in the class should watch the teacher as s/he moves a motion block (right, left, up, down) to the scripting area and presses the block to make the Scratch cat move. The children should duplicate this task. The teacher should request that students raise their hands when they are finished with this task. Do this for each motion block. Do the same for the resize blocks (bigger and smaller) and visibility blocks.

ScratchJr Exploration (10 minutes): The teacher should encourage students to explore the application by placing blocks in the scripting area and seeing where the cat moves.

Wrap Up (5 minutes): The teacher should demonstrate how to save a project. Every child should save his project. The teacher should provide students with a brief explanation of what will occur during the next lesson. Collect iPads.

● ●

Appendix II

These are the teachers' shared notes that evolved over the five lesson study cycles. The online notes were color coded so that the teachers could see where the ideas were coming from.

CS Lesson 1: Teachers Notes

CS Lesson 1: Introduction to CS and Programming Constructs

Part 1

Introduction:

Have you ever thought about what you want to be when you are older? What job or career you would like to have? Today I am going to teach you a little bit about another type of job you may be interested in having. Its called a computer scientist! Can you all say that? (put up vocab word)

**Read Aloud or show "Magic School Bus gets Programmed" day before-- helpful to begin the conversation. I played the "Magic School Bus gets Programmed" video the day before the Scratch lesson and felt that it helped the students get an understanding of the concept prior to the first lesson.

- Have you ever heard that term before?

- What does the term "computer science" mean to you?

- What have they done with computers?

- Do you ever wonder how your video games are made or how computers work?

- Why is computer science important?

  - Job opportunities

- How computers/games work

- Teaches us how to think and problem solve

- Computers are everywhere. They help us solve problems faster than if we didn't have computers (steve jobs bicycle analogy)

- **I used analogy of a student walking to *** Park compared with another riding his bike to the park…, who would get there first?... the bicycle is the tool to get there faster.... like computers are the tool for our mind…(Steve Jobs)**

People who make computers work they way we want them to are called computer programmers (put up vocab word). Have you ever thought of being a computer scientist? Would you like to learn more about what computer scientists do?

Computers work when given a set of clear instructions in a specific sequence. Sequence is an order of instructions, *I only talked about sequencing, not loops and branches. **I gave an example of how we have a daily routine in our classroom and how students come in and know the sequence of instructions. Example: put chairs down, pick a lunch and get started on morning work.** You can choose if you want to mention loops and branches also. (loops are doing the same instructions over and over, and branching is moving from one set instructions to another). (put up vocab words- instructions, sequence)

Discuss Essential Questions (I will add to this tomorrow when I have the questions):

**Teacher7 suggested using the Mimio to project Android Emulator on the laptop ...better as the modeling is closer to what students will do when using

153

their tablets and can project vocabulary and directions. The mimio was useful

for modeling how to turn the tablet on and how to click and drag objects.

Essential Questions:

1.   What is computer science? What is computer programming?

2.   How can we give clear instructions?

3.   Why are we learning computer science?

4.   How do computer programs work?

5.   How can we create computer programs?

Simon Says:
- Practice giving directions
- Computers need specific, clear instructions in order to work properly. These instructions come from computer programmers who give the computer the instructions.
- Progression of directions
    - Touch your head
    - Touch your elbow
    - Use your right hand to touch your left shoulder
    - Use your left hand to touch your right shoulder then take 1 step forward
    - Take two steps forward and one step right
    - etc.etc.

Program the Teacher:
- **Model the giving of instructions --can use Teacher6 or 7... may help students with confusion of how to give the instructions…. Spend about 5 minutes on this activity (it is just to get them to start understanding how important instructions are and how we can go back to fix our mistakes, just like computer programmers)**
- Work with groups of **2 or** 3 use writing partners as pairings when possible
- Give groups different destinations
- Do example first- move forward _____ steps, move, left _____ steps
- Students work together to fill out template **(modified template with blocks) good help for ELL students**
- Groups can move around and test their instructions while working.
- Emphasize using specific clear instructions and **collaboration**

154

Review working with group members:

- How to work well with group members. How can you be a respectful and responsible group member? Give examples and nonexamples.

- Have students share "programs" for the teacher to move from point A to point B. They will notice their programs will not work exactly the way they want them to but that's ok. Emphasize the importance of going back and fixing your work as computer scientist always do.

Getting Started with Scratch:
- introduce this program as a way in which we can practice computer programming giving characters specific instructions on how to move
- Show model on computer
- Hand out tablets- show how to - power on, open scratch, sample project, how to begin new project

Appendix III

These are some of the slides that the teachers used to teach CS fundamentals.

BIBLIOGRAPHY

ABCTE-American Board for Certification of Teacher Excellence. (2015). Most states adopt common core standards but face stem teacher shortages. Washington, DC. Accessed 2/19/2017 from: http://abcte.org/high-standards-low-supply-states-sets-high-standards-but-face-stem-teacher-shortages/

Adelman, C. (1993). Kurt Lewin and the origins of action research, *Educational Action Research*, 7(24).

Albinson P. (2013). A review into the factors affecting declines in undergraduate Computer Science enrolments and approaches for solving this problem. Inspire DEC Student Conference 2013, Bournemouth University, Poole, England, 1-9.

Akiba, M., & Liang, G. (2016). Effects of teacher professional learning activities on student achievement growth. *The Journal of Educational Research*, *109*(1), 99–110.

Allen,M. (2003. Eight questions on teacher preparation: What does the research say? A summary of the findings. Education Commission of the States, Denver, CO.

Angelo, T. (2001). Doing faculty development as if we value learning most: Transformative guidelines from research to practice. To Improve the Academy, 19, 97-112. Bolton, MA: Anker Publishing.

Armoni, M. (2013). Designing a K-12 computing curriculum - The questions. *ACM Inroads*, *4*(2), 34–35.

Armoni, M., & Gal-Ezer, J. (2014). Early computing education— WHY? WHAT? WHEN? WHO? *ACM Inroads*, *5*(4), 54–59.

Atweh, B., & Clarkson, P. (2002). Globalized curriculum or global approach to curriculum reform in mathematics education. Asia Pacific Education Review, 3(2), 160–167.

Baba, T. (2007). How is lesson study implemented? In M. Isoda, M. Stephens, Y. Ohara & T. Miyakawa (Eds.), Japanese lesson study in mathematics: Its impact, diversity and potential for educational improvement (pp. 2–7). Singapore: World Scientific Publishing, Co.

Barr, V., & Stephenson, C. (2011). Bringing Computational thinking to K-12: What is involved and what is the Role of the computer science education community? *ACM Inroads*, *2*(1), 48–54.

Barrett, D., Riggs, L., & Ray, J. (2013). Teachers' collaborative use of the lesson study approach to foster student achievement in geometry. *International Journal of Social Science & Education*, *3*(4).

Baskerville, R. L. (1999). Investigating information systems with action research. *Communications of the Association for Information Systems*, *2*(3), 1–32.

158

Bers, M. U. (2010). The tangibleK robotics program: Applied computational thinking for Young Children. *Early Childhood Research & Practice*, *12*(2).

Bers, M. U., & Resnick, M. (2016). *The Official Scratch Jr Book*. No Starch Press.

Beyer, S. (2014). Beyer - 2014 - Why are women underrepresented in computer science: Gender differences in stereotypes, self-efficacy, values, and interest. *Computer Science Education ISSN:*, *24*(2), 153–192.

Blitz, C. L. (2013). Can online learning communities achieve the goals of traditional professional learning communities? What the literature says. Regional Educational Laboratory Mid-Atlantic, (September, 2013). Retrieved 2/21/2017 from:

http://search.ebscohost.com/login.aspx?direct=true&db=eric&AN=ED544210&site=eds-live&scope=site

Butterfield, J. (2009). Using grounded theory and action research to raise attainment in and enjoyment of reading. *Educational Psychology in Practice*, *25*(4), 315–326.

Bruner, J. (1960) *The Process of Education*, Cambridge, Mass.: Harvard University Press. 97 + xxvi pages.

Cannella, B. G. S., & Reiff, J. C. (1994). Teacher Education: Teachers as Empowered Learners. Teacher Education Quarterly, 21(3), 27–3.

Caro-Bruce, C., M. Klehr, K. Zeichner, and L.M. Piedrahita (2007). A school district-based action research program in the US. In the *Handbook of Educational Action Research*. Thousand Oaks, CA: Sage.

Carr, W., & Kemmis, S. (1986). *Becoming critical: Education, knowledge and action research*. London:Falmer.

Charmaz, K. (2014). *Constructing Grounded Theory*. SAGE

CINC (2012). Advocating for K-12 computer science education, Annual Report for 2012. Washington DC: Computing In The Core.

Clements, D. H. (2002). Computers in Early Childhood Mathematics. *Contemporary Issues in Early Childhood*, *3*(2), 160.

Code.Org (2016). About Code.Org. Accessed: 2/19/2017 from: https://code.org/about

CCSS (2016). Common core state standards: Program website. Accessed: 2/21/2017 from: http://www.corestandards.org/about-the-standards/

Corbin, J. M., & Strauss, A. (1990). Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology*. 13(1).

Covay, E., Desimone, L. M., Caines Lee, J., & Hochberg, E. (2016). Insights on how to shape teacher learning policy: The role of teacher content knowledge in explaining differential effects of professional development. *Education Policy Analysis Archives*, 24(61).

Creswell, J. W. (2014). *Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE.

CSTA-Computer Science Teachers Association (2011), K-12 computer science standards: Revised 2011, New York, NY: Association for Computing Machinery.

CSTA- Computer Science Teachers Association (2016). K–12 Computer Science Framework. Retrieved 3/29/2017 from http://www.k12cs.org.

CS4RI (2016). Program website. Accessed: 2/19/2017 from: http://www.cs4ri.org

Cuban, L. (2001). *Oversold And Underused*. Cambridge, MA: Harvard University Press.

Darling-Hammond, L. (2010). Teacher education and the American future. *Journal of Teacher Education*, *61*(1-2), 35–47.

Dewey, J. (1916). *Democracy and education: An introduction to the philosophy of education.* Macmillan.

Dewey, J. (1925). *Experience and Nature*. Open Court Publishing Company.

Dick, B. (2003). *AR and grounded theory*. Paper prepared for the research symposium at the Australian and New Zealand ALARPM/ SCIAR conference.

Dudley, P. (2011). How lesson study orchestrates key features of teacher knowledge and teacher learning to create profound changes in professional practice. Presented at the World Association of Lesson Studies Annual Conference, Tokyo.

161

Dudley, P. (2013). Teacher learning in lesson study: What interaction-level discourse analysis revealed about how teachers utilised imagination, tacit knowledge of teaching and fresh evidence of pupils learning, to develop practice knowledge and so enhance their pupils' lea. *Teaching and Teacher Education*, 34, 107–121.

Dufour, R. (2011). Professional learning communities: A bandwagon, an idea worth considering or our best hope for high levels of learning? *Counterpoints* Vol. 408, Teacher Leadership: The "New" Foundations of Teacher Education, pp. 159-164

DuFour, R. (2004). What is a professional learning community? *Educational Leadership,* 61(8), 6-11.

Edwards, S. G. (2014). Lesson Study: A mechanism to support effective teacher engagement with and in educational research? A think-piece. *Journal of Educational Research*, 1(1), 48–64.

EEO-1. (2014). Single, Headquarters, and Establishment Reports, EEOC. Accessed 2/21/17 from: https://www.eeoc.gov/eeoc/statistics/employment/jobpat-eeo1/index.cfm

EEOC-U.S. Equal Employment Opportunity Commission. (2016). *Diversity in high tech*. Accessed: 2/20/2017 from:

https://www.eeoc.gov/eeoc/statistics/reports/hightech/

Elliott, J. (2016). The mentoring process and Lesson Study: are they compatible? *Open Online Journal for Education*, Special Issue #5. Accessed 3/8/2017 from: http://journal.ph-noe.ac.at

Ericson, B., Armoni, M., Gal-Ezer, J., Seehorn, D., Stephenson, C., & Trees, F. (2008). *Ensuring exemplary teaching in an essential discipline: Addrssing the crisis in computer science teacher certification*. CSTA.

Ermeling, B. A. (2010). Tracing the effects of teacher inquiry on classroom practice. Teaching and Teacher Education, 26(3), 377–388.

Felleisen, M., & Krishnamurthi, S. (2009). Viewpoint: Why computer science doesn't matter. *Communications of the ACM*, 52(7), 37.

Fraenkel, J.R., Wallen, N.E. & Hyung, H. H. (2011). *How to Design and Evaluate Research* (8th ed.). New York: McGraw-Hill.

Franklin, D. (2015). Putting the computer science in computing education research. *Communications of the ACM*, *58*(2), 34–36.

Fullan, M.G. (1992). Visions that blind. *Educational Leadership*, 49(5), 19-20. Gordon,

Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science. In R. T. Mittermeir (Ed.), *Teaching Fundamental Concepts of Informatics: 4th International Conference* (pp. 159–168). Springer-Verlag.

Gal-Ezer, J., & Stephenson, C. (2014). A Tale of Two Countries, and a question. *ACM Transactions on Computing Education*, *14*(2).

Garet, M. S., Porter, A. C., Desimone, L., Birman, B. F., & Yoon, K. S. (2001). What makes professional development effective? Results from a national sample of teachers. *American Educational Research Journal*, *38*(4), 915–945.

Gewertz, C. (2015). Editorial Projects in Education Research Center. (2015, September 28). Issues A-Z: The Common Core Explained. *Education Week*. Retrieved: February 24, 2017 from http://www.edweek.org/ew/issues/common-core-state-standards/

Glaser, B. G., & Strauss, A. L. (1965). *Awareness of dying*. Chicago, IL: Aldine Publishing.

Glaser, B., & Strauss, A. (1967). *The Discovery of Grounded Theory: Strategies for QualitativeRresearch*. London, UK: Weidenfeld & Nicholson.

Goode, J., & Margolis, J. (2011). Exploring Computer Science: A Case Study of School Reform. *ACMTransactions on Computing Education*, *11*(2), 12:1–12:16.

Google. (2015). Searching for computer science: Access and barriers in U.S. K-12 education. Retrieved 2/21/2017 from

https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf

Guba, E. (1981). Criteria for Assessing the Trustworthiness of Naturalistic Inquiries. *Educational Technology Research and Development*, 29 (2), 75-91.

Guskey, T. R. (1997). Research needs to link professional development and student learning. *Journal of Staff Development*, 18(2), 3640.

Guskey, T. R. (2000). *Evaluating professional development*. Thousand Oaks, CA: Corwin Press.

Guskey, T. R., & Sparks, D. (1996). Exploring the relationship between staff development and improvements in student learning. *Journal of Staff Development*, 17(4), 34-3

Guzdial, M. (2014). Guzdial 2014 Why the US is not ready for mandatory CS education. *Communications of the ACM*, *57*(8), 8–9.

Guzdial, M., Ericson, B., Mcklin, T., & Engelman, S. (2014). Georgia Computes! An Intervention in a US State, with formal and informal education in a policy context. *ACM Transactions on Computing Education*, *14*(2), 13:1–13:29.

Guzdial, M., Morrison, B., Tew, A. E., & Galanos, R. (2011). Building a community to support HS CS teachers: the Disciplinary Commons for Computing Educators. SIGCSE Bull., (March 9-12, 2011).

Hagevik, R., Aydeniz, M., & Rowell, C. G. (2012). Using action research in middle level teacher education to evaluate and deepen reflective practice. *Teaching and Teacher Education*, 28(5), 675–684.

Harsha, P. (2016, January 30). President announces huge new "Computer Science for All" Initiative. *Computing Education*. Accessed 2/19/2017 from: http://cra.org/govaffairs/blog/2016/01/president-announces-huge-new-computer-science-for-all-initiative/

Hargreaves, A. (2007). Sustainable professional learning communities. In L. Stoll & K. S. Louis (Eds.), *Professional learning communities: Divergence, depth and dilemmas* (pp. 181–196). Berkshire, UK: Open University Press.

Harris, J. B., & Hofer, M. J. (2011). Technological Pedagogical Content Knowledge (TPACK) in Action: A Descriptive Study of Secondary Teachers' Curriculum-Based, Technology-Related Instructional Planning. *Journal of Research on Technology in Education*, *43*(3), 211–229.

Hayes, G. R. (2011). The relationship of action research to human-computer interaction. *ACM Transactions on Computer-Human Interaction*, *18*(3), 1–20.

Hiebert, J., & Morris, A. K. (2012). Teaching, Rather Than Teachers, As a Path Toward Improving Classroom Instruction. Journal of Teacher Education, 63(2), 92–102.

Huberman, M. (1983). The role of teacher education in the improvement of educational practice: A linkage model. *European Journal of Teacher Education*, 6(1), 17-29.

Hubweiser, P., Armoni, M., Giannakos, M., & Mittermeir, R. (2014). Perspectives and visions of computer science education in primary and secondary (K-12) Schools. *ACM Transactions on Computing Education*, 14(2), 7:1-7:9.

Hubwieser, P., Armoni, M., & Giannakos, M. N. (2015). How to implement rigorous computer science education in K-12 schools? Some answers and many questions, *ACM Transactions on Computing Education*, 15(2).

Jenkins, S., & Agamba, J. J. (2013). The missing link in the CCSS initiative: Professional development for implementation. *Academy of Educational Leadership Journal*, *17*(2), 69–80.

Kennedy-Clark, S. (2012). Design research and the solo higher degree research student: Strategies to embed trustworthiness and validity into the research design. *Joint International Conference of the Australian Association for Research in Education - AARE – APERA*. Sydney, AU.

Kordaki, M. (2013). High school computing teachers' beliefs and practices: A case study. *Computers and Education*, 68, 141–152.

Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge: Cambridge University Press.

167

Lee, S.W., Duncan, T.,Yoon, K. S., Scarloss, B., Shapley, K. L. (2007). Reviewing the evidence on how teacher professional development affects student achievement Reviewing the evidence on how teacher professional development. Issues and Answers Report, REL 2007–No. 033. Washington, DC: U.S. Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance, Regional Educational Laboratory Southwest. Retrieved 2/21/17 from http://ies.ed.gov/ncee/edlab

Lewin, K. 1946. Action research and minority problems. Journal of Social Issues 2, no. 1: 34–46.

Lewis, C. (2002). Does lesson study have a future in the United States? Nagoya Journal of Education and Human Development, (1), 1–23.

Lewis, C. C. (2011). Lewis Beyond Coaching. Educational Leadership, 69(2), 64–68.

Lewis, C., & Hurd, J. (2011). Lesson study step by step: How teacher learning communities improve instruction. Portsmouth, NH: Heinemann.

Lewis, C., & Perry, R. (2014). Lesson study with mathematical resources: A sustainable model for locally-led teacher professional learning. Mathematics Teacher Education & Development, 99–116.

Lewis, C. C., Perry, R., & Hurd, J. (2004). A deeper look at lesson study. Educational Leadership, 61(5), 18–23.

Lewis, C. C., Perry, R., Friedkin, S., & Roth, J. R. (2012). Improving teaching does improve teachers Evidence from lesson study. Journal of Teacher Education, 63(5), 368–375.

Lewis, C. C., Perry, R. R., & Hurd, J. (2009). Improving mathematics instruction through lesson study: A theoretical model and North American case. Journal of Mathematics Teacher Education, 12(4), 285–304.

Lewis, C., Perry, R., & Murata, A. (2006). How should research contribute to instructional improvement? The case of lesson study. *Educational Researcher*, 35(3), 3–14.

Lincoln, Y. & Guba, E. (1985) *Naturalistic Inquiry*. Beverly Hills, CA, Sage.

Marrongelle, K., Sztajn, P., & Smith, M. (2013). Scaling up professional development in an era of common state standards. *Journal of Teacher Education*, 64(3), 202–211.

McTaggart, R. (1994). Participatory Action Research: issues in theory and practice. *Educational Action Research*, 2(3), 313–337.

Microsoft. (2012). A national talent strategy: Ideas for securing US competitiveness and economic growth, 26. Accessed: 3/2/2017 from: https://news.microsoft.com/download/presskits/citizenship/MSNTS.pdf

169

Miles, M., & Huberman, A. M. (1984). Drawing valid meaning from qualitative data. *Educational Researcher*, 15(5).

Morreale, P., & Joiner, D. (2011). Changing perceptions of computer science and computational thinking among high school teachers. *Journal of Computing Sciences in Colleges*, 26(6), 71–77.

Murata, A., Bofferding, L., Pothen, B. E., Taylor, M. W., & Wischnia, S. (2012). Making connections among student learning, content, and teaching: Teacher talk paths in elementary mathematics lesson study. *Journal for Research in Mathematics Education*, 43(5), 616–650.

National Research Council. (2001). *Adding it up: Helping Children Learn Mathematics* (J. Kilpatrick, J. Swafford, & B. Findell, Eds.). Washington, DC: National Academy Press.

NCTE. (2010). Teacher learning communities. *The Council Chronicle*. National Council of Teachers of English.

Norwich, B., & Ylonen, A. (2013). Design based research to develop the teaching of pupils with moderate learning difficulties (MLD): Evaluating lesson study in terms of pupil, teacher and school outcomes. *Teaching and Teacher Education Journal*, 34, 162–173.

NSF. (2016). Elementary and secondary mathematics and science education. *Science and Engineering Indicators,* 1–115.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York, Basic Books.

Papert, S. (1998). Do easy do it? Children, games, and learning. *Game Developer Magazine*, June, 1998.

Patterson, D., Hennessy, J. (2013) Computer organization and design: The hardware/software interface. Morgan Kaufman.

Pauleen, D. J., & Yoong, P. (2004). Studying Human-Centered IT Innovation Using a Grounded Action Learning Approach. *The Qualitative Report*, 9(1), 137-160.

Perry, R. R., & Lewis, C. C. (2009). What is successful adaptation of lesson study in the US? *Journal of Educational Change*, 10(4), 365–391.

Polya, G. (1945). *How to Solve It*. Princeton, NJ: Princeton University Press.

Popp, J. S., & Goldman, S. R. (2016). Knowledge building in teacher professional learning communities: Focus of meeting matters. *Teaching and Teacher Education*, 59, 347–359.

Prottsman, K. (2014). Computer Science for the Elementary Classroom. *ACM Inroads*, 5(4), 60–63.

Ratts, R. F., Pate, J. L., Archibald, J. G., Street, N. P., Building, C. A. C., Andrews, S. P., … Street. (2015). The Influence of Professional Learning Communities on

Student Achievement in Elementary Schools United States of America United States of America. *Journal of Education & Social Policy*, 2(4), 51–61.

Reason, P., & Bradbury, H. (Eds.). (2008). *Handbook of Action Research: Participative inquiry and practice*. Thousand Oaks, CA: Sage.

Repenning, A., Webb, D. C., Koh, K. H., Nickerson, H., Miller, S. B., Brand, C., … Repening, N. (2015). Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools through Game Design and Simulation Creation. *ACMTransactions on Computing Education*, *15*(2), 11:1-11:31.

RIDE (2016). Rhode Island Department of Education: Infoworks. Accessed 2/21/2017 from: http://www.ride.ri.gov/

Sack, Joetta. (1999, May 5). Class size, teacher quality take center stage at hearing. *Education Week*.

Saito, E., & Atencio, M. (2013). A conceptual discussion of lesson study from a micro-political perspective: Implications for teacher development and pupil learning. *Teaching and Teacher Education*, 31, 87–95.

Sarama, J., & Clements, D. H. (2009). "Concrete" computer manipulatives in mathematics education. Child Development Perspectives, 3(3), 145–150.

Schön, D. (1983). *The Reflective Practitioner*. New York: Basic Books.

Scott, R. E. (2015). The manufacturing footprint and the importance of U.S. manufacturing jobs. Report, January 20, 2015. Economic Policy Institute. Washington, DC.

Shaw, G.B. (1903). *Man and Superman: Maxims for Revolutionists.* University Press, Cambridge, MA. Accessed 3/25/17 from: http://www.bartleby.com/br/157.html

Shulman, L.S. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, 15(4), 4-14.

Shulman, L. (1987). Knowledge and Teaching: Foundations of the New Reform. *Harvard Education Review*, 57(1), 1–20.

Sim, M. (2009). Dewey and Confucius: on Moral Education. *Journal of Chinese Philosophy*, 36(1), 85–105.

Simon, Fincher, S., Robins, A., Baker, B., Box, I., Cutts, Q., … Tutty, J. (2006). Predictors of success in a first programming course. *Proceedings of the 8th Austalian Conference on Computing Education,* Vol. 52, 189–196.

Somekh, B., & Zeichner, K. (2009). Action research for educational reform: remodelling action research theories and practices in local contexts. *Educational Action Research*, 17(1), 5–21.

Stigler, J. W., & Hiebert, J. (1999). *The Teaching Gap*. New York: The Free Press.

Stigler, J. W., & Hiebert, J. (2016). Lesson study, improvement, and the importing of cultural routines. *Mathematics Education*, 48(4), 581–587.

Stroustrup, B. (2009). Programming in an undergraduate CS curriculum. In: *Proceedings of the 14th Western Canadian Conference on Computing Education (WCCCE '09)*. New York, NY: ACM.

Talbert, J.E. (2010) Professional learning communities at the crossroads: How systems hinder or engender change. In A. Hargreaves, A. Lieberman, M. Fullan, & D. Hopkins (Eds.), *The Second International Handbook of Educational Change* (pp. 555–571). New York: Springer.

Taub, R., Armoni, M., & Ben-Ari, M. (2012). CS Unplugged and Middle-School Students' Views, Attitudes, and Intentions Regarding CS. *ACM Transactions on Computing Education*, 12(2), 8:1-8:28.

Tempera, J. (2016, March 7). Local businesses get behind Raimondo's CS4RI effort. The Providence Journal.

Fincher, S., & Tenenberg, J. (2007). Opening the door of the computer science classroom: the disciplinary commons. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, *39*(1), 514–518.

Tenenberg, J., & McCartney, R. (2014). Editorial: Computing Education in (K-12) Schools from a Cross-National Perspective. *ACM Transactions on Computing Education*, 14(2), 6:1-6:3.

Trilling, B., & Fadel, C. (2009). *21st Century Skills: Learning for life in our times*. San Francisco: Jossey-Bass.

Vescio, V., Ross, D., & Adams, A. (2008). A review of research on the impact of professional learning communities on teaching practice and student learning. *Teaching and Teacher Education*, 24(1), 80–91.

Wang, J. C.(2007). John Dewey in China: To Teach and to Learn. Albany: State University of New York Press. Accessed: March 4, 2017, from: https://muse.jhu.edu/book/5234

Wastell, D. (2001). Barriers to effective knowledge management: Action research meets grounded theory. *Journal of Systems and Information Technology*, 5(2).

Waterman, S. (2011). A study of lesson study's impact on student achievement. *Silicon Valley Mathematics Initiative -SVMI*. Morgan Hill, CA

Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. Cambridge: Cambridge University Press.

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). Running on empty. ACM & CSTA.

Williams, J. (2015). The 5 biases pushing women out of STEM. Harvard Business Review (March 24, 2015). Accessed: 4/2/2017 from: https://hbr.org/2015/03/the-5-biases-pushing-women-out-of-stem

Wong, H. K. (2000). There is only one way to improve student achievement. *Harry K. Wong Publications*. Saratoga, CA.

Wynne, J. (2001). Teachers as leaders in education reform. *Education*, 1–7.

Yongpradit, P. (2014). Should we teach computer science in elementary school? *Point-CounterPoint. ISTE*. Accessed: 2/21/2017 from: https://www.iste.org/explore/articledetail?articleid=216

Yoong, P. (1996). Action learning: Preparing workers for the international office of the future. *IFIP WG8.4 Conference*, Tucson, Arizona.

Yoshida, M. (2008). Exploring ideas for a mathematics teacher educator's contribution to lesson study: Towards improving teachers' mathematical content and pedagogical knowledge. In D. Tirosh (Vol. Ed.) & T. Wood (Vol. Ed. & Series Ed.), *The International Handbook of Mathematics Teacher Education*: Vol. 2. Rotterdam, the Netherlands.