How Equity and Inequity Can Emerge in Pair Programming

Colleen M. Lewis
Harvey Mudd College
301 Platt Blvd
Claremont, CA 91711
001-909-607-0443
lewis@cs.hmc.edu

Niral Shah
Michigan State University
116P Erickson Hall
East Lansing, MI 48824
001-517-432-9991
niral@msu.edu

ABSTRACT

Research suggests that pair programming increases student performance and decreases student attrition. However, less is known about the ways in which pair programming can unintentionally lead to inequitable relationships between students. Audio data were collected from pair programming interactions in a sixth-grade computer science enrichment program designed to promote equity. However, even in this context, there were surprising instances of inequity. We measured inequity by documenting the distribution of students' questions, commands, and total talk within four pairs. Analysis revealed that less equitable pairs sought to complete tasks quickly and this may have led to patterns of marginalization and domination. Notably, this focus on speed was not evident in the more equitable pairs. These findings are important for understanding mechanisms of inequity and designing equitable collaboration practices in computer science.

Categories and Subject Descriptors

K.3.2 **[Computers and Education]:** Computer and Information Science Education—computer science education

General Terms

Human Factors.

Keywords

Equity; diversity; pair programming; collaborative learning

1. INTRODUCTION

Research has shown that pair programming (i.e., having two students share a computer while programming) can increase students' learning, retention in a CS major, and sense of belonging (see [31] for a review of pair programming benefits). While in aggregate these results appear overwhelmingly positive, students and educators have noted instances where pair programming appears to limit one or both of the partners' opportunities to learn. We found that while the pair-programming structures were designed to promote equitable participation [34], in some cases gross inequity emerged within a partnership. In the examples presented here, we attribute the students' goals for completing work as quickly as possible (i.e., speed) as facilitating inequitable interactions.

ICER '15, August 9–13, 2015, Omaha, Nebraska, USA.

ACM 978-1-4503-3630-7/15/08…$15.00.
http://dx.doi.org/10.1145/2787622.2787716

With student and parent consent, data were collected in a 2012 summer computer science (CS) course taken by academically advanced students entering the sixth grade (i.e., 11-12 years old). Our main data source was audio recordings of pairs of students working together to solve computer-programming problems on a single computer. To triangulate this data source and guide the research focus, we considered additional data including: students' written and electronic work, videos of the class, and ethnographic fieldnotes focused on students' interactions and whole class discussion.

We chose to focus on a single student, "Jason" (pseudonym), and the interactions with his partners because the research and teaching teams perceived Jason's interactions to span from more equitable to less equitable. This variety offered an opportunity to understand the ways in which a single student may engage in very different interactions. Our prior work developed a coding scheme to measure the approximate level of equity within a pair [35]. This coding scheme allowed us to quantify features of collaboration that we argue are indicative of equity or inequity (e.g., the distribution of students' questions, commands, and total talk). Additionally, this coding scheme allowed us to compare across interactions.

Our current analysis focused on four 90-minute audio recordings of Jason. In each of these he is working with a different partner. Our analysis began by applying the coding scheme from our prior work to gauge the relative equity within each of the four dyads. Based upon this coding we were left with the following open question: Why were two of the dyads (Aaron-Jason and Peter-Jason) far less equitable than the other two dyads (Samantha-Jason and Kim-Jason)? We attempted to catalogue differences between the more and less equitable dyads to try to explain the differences.

We identified three central patterns in the less equitable collaborations: sequences of commands interspersed with Jason asking clarifying questions (*command-clarify sequences*); the use of shortcuts (*shortcuts*); and frequent comparison of progress or accomplishment with peers (*peer comparison*).

Across these patterns, we observed a central focus on completing tasks quickly (i.e., speed), which may have produced the patterns of inequity within the Aaron-Jason and Peter-Jason dyads. Upon evaluating a number of alternate hypotheses, we argue that a focus on speed best explains the patterns of inequity that developed. This insight is relevant for understanding how inequity can emerge within pair programming, which was designed to improve students' learning opportunities.

2. THEORETICAL FRAMEWORK

In educational research, the term "equity" has been used to refer to the degree of students' access to the resources needed for learning [10, 29]. Defined in this way, equity can be analyzed at a structural level, in terms of students' access to qualified teachers, material resources like textbooks, and opportunities to take advanced coursework at their school (see [17]). Research in CS education is increasingly focused on ensuring that all students have equitable access to the resources needed for learning [35, 24]. This scholarship recognizes that large segments of the population—particularly women and people of color—remain excluded from opportunities to learn CS [14, 24, 47]. Not only can such inequities have implications for these groups' access to future economic opportunities, but they also raise basic moral concerns about fairness.

Complementary to this structural view of equity, our equity research emphasizes whether all students have opportunities to participate in the everyday social interactions central to learning environments [15, 21]. This approach is grounded in situated [22] and sociocultural perspectives [38, 32, 45] on learning, which illuminate the impact of student participation on student learning. Participation, from a situated and sociocultural perspective, draws attention to the particular ways in which students participate in particular classroom activities, such as working with peers or explaining their ideas. This use of "participation" differs from how the term is often used in conversations about equity (e.g., representation of different groups in CS).

Using participation as a measure of equity, researchers have focused on different dimensions of the collaborative learning setting [7, 8, 15, 21]. Research shows that, while promising, collaborative learning is complex and insufficient to guarantee equity [13, 33].

Ideally, an equitable collaboration would mean that no student disproportionately dominates the conversational floor. For example, when students are brought together in a collaborative learning situation, the teacher's intention is that all of the students will contribute ideas that influence the ultimate outcome of the joint problem solving process. Further, all of the students would feel they have license to critique and build on their group mates' ideas.

Our research is important because it expands considerations of equity beyond issues related to the K-16 "pipeline." That is, while it is important that we continue to strive for equitable representation of all demographic groups in CS, it is also important that we consider how inequities can arise in classroom interactions as students engage in the learning process. In that sense, the present study complements much of the existing literature on equity in CS education—which tends to focus on structural inequities—by considering how equity and inequity operate at the level of everyday activity in learning environments.

3. PREVIOUS RESEARCH

A significant body of research shows that collaborative learning is beneficial for students' learning (see [15]). Researchers have identified a number of conditions and interactional forms conducive to learning in collaborative contexts (for comprehensive reviews, see [6, 40, 15]). The literature has primarily focused on the impact of particular discursive moves, such as asking questions [20], explaining one's thinking [16, 28, 41], and taking up a peer's ideas [2].

Building upon the success of collaborative learning, research has demonstrated the value of a CS-specific form of collaboration: pair programming [19, 23, 25, 26, 31, 46, 18]. Pair programming involves two students sharing a single computer as they work on solving programming problems [46]. Pair programming has demonstrated improved performance outcomes in introductory CS courses [25, 27] and software engineering courses [5, 46]. It has been used to improve student performance [25, 27] and increase retention among students who are underrepresented in CS [26].

Pair programming research has focused on the compatibility of pairs [18]. Researchers frequently recommend pairing students of similar ability to increase compatibility. This pairing strategy has been correlated with increased student satisfaction [11, 36, 37], decreased reports of compatibility problems [39], and increased performance for students in the lowest quartile of performance [4]. Students in our class were not paired with similar ability students. In fact, the current work explores the interactions among four higher performing students (Aaron, Kim, Peter, & Samantha) when paired with a lower performing student (Jason). Our analysis may add complexity to the field's understanding of the nature of interactions between higher and lower performing partners.

The majority of pair programming research has taken place in industry and at the college level; the generalizability of these results to middle school students remains an open question [18]. While less common, researchers focusing on middle-school students have sought to explore the conditions under which pair programming is most effective [23, 12], as well as the dynamics within pair interactions [43, 44]. While it is unclear if research focused on adults generalizes to younger students, the current and prior qualitative research focused on middle-school students [12, 43, 44] illuminates patterns of interaction that are likely applicable to adults.

4. METHODS

4.1 Research Context

Data were collected in a twelve-day summer CS course for students entering the sixth grade. This 36-hour course was offered through a university-sponsored program for academically high-achieving students. The course was taught by the co-authors with assistance from two adult teaching assistants.

In the course, students learned the basics of computer programming using the programming languages Scratch and Logo. Although the course required no prior programming experience, the course was designed to be challenging and to offer significant practice with iteration, and other CS topics. Each of the twelve instructional days typically included lecture, programming tasks sequenced within an online curriculum, and a 15-minute, paper-based assessment. On alternating days students completed programming tasks in pairs using pair programming. The course instructors assigned students to pairs. Every five minutes, students in the course alternated roles of "driver," who operated the keyboard and mouse, and "navigator," who provided verbal direction without touching the keyboard and mouse. These roles were intended to promote equitable collaboration (cf. [30]). Details regarding the goals, structure, and design of the class have been documented in a previous publication [34].

All data presented here are from one of two offerings of the course in the summer of 2012. In that offering, there were 45 students, 23 (51%) of whom were identified as female on course enrollment paperwork.

4.2 Data Collection

With student and parent consent, we collected all of students' hand-written and electronic work as well as audio recordings of students working, video recordings of the class, teachers' notes, and ethnographic fieldnotes. All class time was video recorded and observed by at least one of three ethnographic researchers. After the first class, six students from each course offering were selected as focal students; these students were selected to attempt to maximize the variation between focal students with respect to gender, race, and personality. For each of the remaining eleven class days, a researcher observed each focal student for at least forty-five minutes and audio recorded for approximately 90 minutes.

4.3 Selection of Analytic Focus

Analysis of the data began with a review of the collection of fieldnotes. Three researchers read, discussed, and summarized each of the 98 total fieldnotes. Based upon these preliminary analyses, our analysis narrowed in on one of the 12 focal students, Jason (all names pseudonyms), and his interactions with four partners: Aaron, Peter, Samantha, and Kim. Jason was selected as the primary focus because his interactions varied considerably across each of his pair programming collaborations. Across these pairs, we perceived Jason as both engaged and unengaged and to be positioned as both competent and incompetent. Our analysis sought insight about supporting equitable collaborations through exploring what may have produced this dramatic variety in participation by one student across four dyads.

The goal of our analysis, and the focus of this paper, is to gain insight into what could explain Jason's varied behavior. Since Jason's collaborations appeared to span from equitable to inequitable, understanding these interactions can help illuminate the dynamics of equitable and inequitable collaboration. In our previous work, we developed methods to document equitable and inequitable collaborations [35]. In the current paper, we build upon these methods for describing and documenting equity, or lack of equity, within a pair programming dyad.

4.4 Quantitative Methods for Classifying Equity in Pair Programming

While a goal of the paper was to explain Jason's varied behavior, a prerequisite for this analytical work was verifying that Jason's behavior or, more accurately, his interactions varied. In previous work [35], we used an iterative process of open coding [9] to develop a coding scheme to capture the degree of equity within a pair programming dyad. This coding scheme was applied to transcripts of audio recordings of individual pairs. The coding scheme was designed to provide multiple levels of granularity. In prior work [35], we showed how additional granularity provided insights into the nature of two of Jason's collaborations. In the current paper, we apply the same coding scheme across transcripts of four of Jason's collaborations. The coding scheme served to document the variation in Jason's interactions, which then allowed for further qualitative analysis of differences.

Our coding scheme privileged quantity and content of talk within the dyads and was customized to capture characteristics of pair programming. We developed metrics for measuring equity within a pair programming dyad. Four of these metrics are featured in the current paper and for each, we describe what we measured, our rationale and any tradeoffs we made.

4.4.1 Distribution of Total Talk

Our first of four metrics for a collaboration was the distribution of talk between the pair. Transcripts of students' interactions were divided into turns. Turns indicate a new sentence or topic by one speaker or a new speaker. We assumed that an equitable collaboration would provide both students access to the conversational floor. Prior research has found that participation in social practices is a core element of the learning process [22, 44]. We used a 50-50 split of students' total talk to evaluate a coarse measurement of the equity within the collaboration. Although equal amounts of talk does not guarantee equity, an expectation of a 50-50 split within an equitable collaboration provided a helpful, coarse evaluation of the pairs.

4.4.2 Distribution of Talk within Pair Programming Roles

Our second metric for a collaboration was the distribution of talk when partners were in each pair-programming role. The roles of navigator and driver lend themselves to different interactional patterns. For example, students might expect the navigator to do the majority of the talking. We calculated the percentage of turns each student took when they were acting as driver and when they were acting as navigator. We anticipated that an equitable collaboration would demonstrate mirroring in the distribution of talk. For example, if the distribution of talk was 70-30 when the first partner was navigating, we hope that the distribution of talk with the second partner was navigating would mirror that distribution (i.e., 30-70). While we expect mirroring to be an indication of equity, we could still observe mirroring if the navigator is consistently unengaged (e.g., 5-95 and 95-5) or if the driver has few opportunities to talk (e.g., 95-5 and 5-95).

4.4.3 Distribution of Commands

Our third metric for a collaboration was the distribution of commands within the dyad. We tagged all lines of transcript that included a command. We classified a command as any statement that included a request to perform an action. Indirect requests (e.g., requests starting with "we should") were not classified as commands. The tag of "command" was one of two high-frequency tags that we selected from a large collection of tags that we developed through an open coding of the transcripts (see [35] for additional details).

While the navigator is expected to help direct the actions of the driver, a prevalence of commands may position a partner as incapable of contributing to the collective task. We expect that an equitable collaboration will have a 50-50 distribution of commands. However, it is unlikely that a collaboration is equitable if it is dominated by commands, even if the partners equally issue commands. Therefore, it may be important to identify if a collaboration has minimal commands, which may be additional evidence of an equitable collaboration.

We expect that the tone of commands shapes the impact the command has on equity. A command issued with an urgent tone or dismissive tone may communicate a lack of respect to the partner. Given that tone would be difficult to consistently document and we cannot know the impact on the participant of a particular command, we aggregate all commands and examine commands that appear particularly impactful using qualitative methods. We accept that not all commands will have the same impact to the equity within the collaboration.

4.4.4 Distribution of Questions

Our fourth metric for a collaboration was the distribution of questions within the dyad. We tagged all lines of transcript that included a question. This was the second, high-frequency tag that we decided to highlight from our original, open coding [35]. We assume that questions are an important mechanism for shaping the relative status of the partners. It appears that being asked a question provides that individual with additional status. Therefore by asking a question a student might give their partner status and by being asked a question a student might receive status. We expect that within an equitable collaboration partners will ask each other questions at similar rates (i.e., a 50-50 distribution). Like commands, not all questions are likely to have the same impact. A student could ask a question with the tone or content indicative of an insult. We accept that including tone could improve our understanding of the impact of these questions, but have chosen to not do so because of difficulty achieving consistency.

4.5 Qualitative Methods

The quantitative methods described above are novel contributions from our prior work [35] and identified gross inequities within the Aaron-Jason and Peter-Jason dyads when compared to the Samantha-Jason and Kim-Jason dyads. However, these quantitative methods provide a relatively narrow lens on the four audio recordings. Their primary contribution in analyzing these data is in identifying a pattern of inequity, which we can then seek to explore and explain using qualitative methods. After completing the quantitative analysis, we employed the following three modes of qualitative analysis for the purpose of exploring and explaining the pattern of inequity in the Aaron-Jason and Peter-Jason dyads.

First, we read, discussed, and re-read transcripts of the four audio recordings. From these readings and discussions we sought to build upon our existing familiarity with the transcripts to identify the salient patterns of interaction within the Aaron-Jason and Peter-Jason dyads that contrasted with patterns within the Samantha-Jason and Kim-Jason dyads. From these reviews, we identified patterns in the Aaron-Jason and Peter-Jason dyads that we referred to as *command-clarify sequences*, *shortcuts*, and *peer-comparison*. Based upon these patterns we attempted to identify representative cases of the patterns.

Second, we looked for commonality across these three patterns of interaction to see larger themes that distinguished the Aaron-Jason and Peter-Jason dyads from the Samantha-Jason and Kim-Jason dyads and from each other. Through this process we identified an overarching focus on speed within the Aaron-Jason and Peter-Jason dyads, which appeared to be absent from the Samantha-Jason and Kim-Jason dyads.

In parallel with other research tasks, we attempted to develop a comprehensive list of plausible alternative hypotheses that could explain the differences between the Aaron-Jason and Peter-Jason dyads and the Samantha-Jason and Kim-Jason dyads. For each of these alternative hypotheses we enumerated what data we would need to confirm or deny the hypothesis and when possible we reviewed these data.

5. QUANTITATIVE RESULTS

Tables 1 and 2 show how talk was distributed between Jason and his four partners measuring the distribution of: total talk, talk within pair programming roles, commands, and questions. The quantitative data suggest patterns of domination and marginalization in Jason's collaborations with Aaron and Peter, and patterns of equity in his collaborations with Samantha and Kim.

Within the Aaron-Jason and Peter-Jason dyads, Jason only contributed roughly one-third of the total turns in both collaborations. Analysis of the distribution of talk within pair programming roles also suggests an inequitable dynamic. Neither dyad exhibited a mirroring pattern when they switch roles. When Jason was the navigator in his partnerships with Aaron and Peter, he contributed only 50% and 45% of turns, respectively. When Aaron or Peter was the navigator, Jason contributed fewer turns than his partners, 33% and 31%, respectively. That Jason did not contribute more than half of the turns when he was navigating further suggests that he may not have had an opportunity to take up a leadership role. Additionally, Jason asked the majority of the questions and Aaron and Peter issued the majority of commands.

Like the Aaron-Jason and Peter-Jason dyads, the data from the Samantha-Jason and Kim-Jason dyads were nearly identical, but in the opposite direction along most metrics. Unlike the Aaron-Jason and Peter-Jason dyads, overall talk was equally distributed and exhibited a mirroring pattern within pair programming roles. The one area where the Samantha-Jason and Kim-Jason dyads were similar to the Aaron-Jason and Peter-Jason dyads was discursive moves: Samantha and Kim asked fewer questions than Jason. Additionally, Samantha issued disproportionately more commands than Jason.

Overall, the quantitative findings in Tables 1 and 2 reveal a stark contrast with respect to equity across the dyads. What might explain this pattern? In the next section, we consider several hypotheses before discussing our conclusion that a focus on speed produced the inequitable patterns present with the Aaron-Jason and Peter-Jason dyads.

Table 1. In each of the four dyads, the percentage of talk Jason contributed in total (row 1) and when serving as navigator (row 2) and driver (row 3). N indicates the combined turns taken by Jason and his partner.

	Aaron	Samantha	Kim	Peter
Total Talk	37% (N=772)	49% (N=526)	50% (N=419)	35% (N=311)
Jason as Navigator	50% (N=282)	55% (N=274)	55% (N=197)	45% (N=82)
Jason as Driver	33% (N=490)	47% (N=252)	46% (N=222)	31% (N=229)

Table 2. In each of the four dyads, the percentage of commands issued (row 1) and questions asked (row 2) by Jason. N indicates the combined count of commands issued and questions asked by Jason and his partner.

	Aaron	Samantha	Kim	Peter
Commands Issued	7% (N=116)	35% (N=68)	47% (N=44)	18% (N=37)
Questions Asked	63% (N=82)	59% (N=52)	75% (N=66)	65% (N=74)

6. QUALITATIVE RESULTS

6.1 Alternative Hypotheses

The quantitative data presented above suggests a stark difference in interactions when Jason was partnered with Aaron or Peter versus when Jason was partnered with Samantha or Kim. This aligned with our fieldnotes and researchers' initial instincts about the quality of these collaborations. We claim that the focus on speed within the Aaron-Jason and Peter-Jason dyads best explains these differences, but we originally explored many plausible explanations. Below we describe hypotheses that we considered and either evaluated to be less likely or determined that the necessary data was not available.

6.1.1 Hypothesis: Friendship

Jason's more equitable collaborations with Samantha and Kim could be caused by Jason's friendship with them. We expect that friends would be more cordial with each other, which could produce a more equitable interaction. Reviewing the fieldnotes, teacher notes, and research recollection, we have no evidence that Jason was friends with Samantha or Kim outside of class (i.e., spent time together during recess). Based upon this we rejected this hypothesis. In fact, we have evidence that Jason and Peter were friends because they both requested to work together on their final project. However, we have no evidence that Aaron and Jason were friends outside of class, so the opposite hypothesis that friendship produces inequitable interactions is unlikely.

6.1.2 Hypothesis: Task Content

Jason's less equitable collaborations with Aaron and Peter could be caused by the more difficult, and possibly more frustrating, nature of their task. We expect that difficult tasks are more likely to be perceived as high-status and are more likely to result in more active positioning. Additionally, we expect that students engaged in frustrating tasks may engage less equitably because their frustration distracts from the interpersonal demands of collaboration. In contrast, Jason's more equitable collaborations with Samantha and Kim could be caused by the more playful tasks that they were engaged in. We expect that when engaged in cooperative play students would engage more equitably because the task is not high-status and the playful tasks require a partner (e.g., playing tag). Reviewing the curriculum from the day, Samantha and Kim both worked with Jason on making and testing games while Aaron and Peter worked with Jason on non-game tasks that involved creating drawings in Scratch and Logo, respectively. After first inspection, this is a strong hypothesis. Additionally, this aligns with the work of Chizhik [8] and Ames [1]. Chizhik argues that open-ended tasks (e.g., designing a game) produce more equal collaborative participation rates, and Ames [1] argues that "personal relevance and meaningfulness of the content" (p. 263) is associated with students' productive engagement. We expect that the nature of the task plays an important role in shaping students' interaction and equity. We expect that this effect was secondary to the focus on speed because those data present a clear connection between the focus on speed and particular inequitable interactions.

6.1.3 Hypothesis: Difference is Content Knowledge

Jason's less equitable collaborations with Aaron and Peter could be caused by gaps in Jason's content knowledge. We expect that collaborations between students with drastically different content knowledge would tend to be less equitable because the students are unequally prepared to contribute to the collaboration. Reviewing students' performance on daily, paper-based assessments, we found that Aaron, Samantha, Kim, and Peter all had scores among the highest scores in the class and Jason had scores among the lowest in the class. This gap in content knowledge could explain the less equitable collaborations with Aaron and Peter, but does not explain the relatively equitable collaborations with Samantha and Kim.

6.1.4 Hypothesis: Preferences for Collaboration

We expect that students who prefer to work alone, rather than in pairs, might engage in less equitable interaction. Before the class began, students were asked to complete a survey about their prior experience with programming, which included the question: "Do you prefer to work alone or with a partner" Jason indicated that he preferred to work alone, as did Peter. This is surprising given that they chose to work together for the final project when working alone was an option. Additionally, Aaron indicated that he preferred to work with a partner, but did not chose to work with a partner on the final project. We have incomplete information for Samantha and Kim. Samantha wrote in "it depends" and Kim did not answer the question or any of the other questions on the back page of this survey.

On the 10th day of class, students turned in a homework assignment on which they answered a similar question of whether they prefer to work alone or in a partner. Jason replied "I think I work well with either because I've had experience in both areas." In contrast, Aaron, Samantha, Peter, and Kim reported a preference for working alone. Aaron's and Peter's responses suggested a lack of investment in collaboration. Aaron wrote "Solo. Pair is too slow and drivers switch rapidly." and Peter wrote "Solo because you don't have to explain anything." Both of these responses seem to focus on speed, either directly in Aaron's frustration with going "too slow" or indirectly in Peter's desire to avoid explaining things to his partner. Aaron's and Peter's responses hint at experiences with partners who were not as competent because Aaron described it as "slow" and Peter seemed to want to avoid having to explain concepts to his partner. In contrast, Samantha and Kim preferred to work alone, but their responses hinted at experiences working with a more competent partner. For example, Samantha wrote, "I like solo programming better because I just like doing things on my own, and not having someone constantly interrupting/bossing me around. I just like to keep up with my own pace and have some quiet." Similarly, Kim wrote, "Solo programming, because I feel that I am never confused, and I feel more confident alone."

These survey data provide more questions than answers. However, the written explanations provided by Aaron and Peter strengthens our hypothesis that they were focused on speed, while those provided by Samantha and Kim do not indicate a speed focus.

6.1.5 Hypothesis: Beliefs about Collaboration

Jason's more equitable collaboration with Samantha and Kim could have been caused by Samantha and Kim's prosocial beliefs about how to treat a low-performing partner. There is strong support for this from Samantha and Kim's written responses to questions on the homework assignment that was collected on the 10th day of class. Kim demonstrated a number of prosocial attitudes on this homework. When explaining whether she preferred to be the driver or the navigator she wrote "Driver, because then I can be sure that my partner and I are both contributing the same [amount] to the project." Additionally, when identifying things you should do during pair programming she wrote "Pay attention, answer your partner's question." When identifying what you should not do, she wrote, "Don't do too much. Don't get side tracked." Samantha when identifying things you should not do during pair programming she wrote that you should not "Go ahead of your partner, even if they don't understand, and do all the quizzes yourself. They won't learn anything." Aaron and Peter also replied to these questions, but their responses demonstrate less evidence of a commitment to equal partnership. Aaron wrote that partners should "try to work together" and should not "touch mouse and keyboard as navigator." Peter wrote that partners should "check in with each other" and not "boss each other around."

While Kim and Samantha's answers restated classroom policies, they also included explanations that mention the classroom goals of partners. For example, they echoed the classroom policies by stating that it is important to "pay attention" (Kim), "don't get side tracked" (Kim), and don't "go ahead of your partner" (Samantha). However, they both seemed to provide an explanation for these policies, for example, Kim explains the goal of "both contributing" and Samantha seems focused on her partner's learning opportunities, "they won't learn anything."

These prosocial beliefs may be inseparable from students' maturity or personality. Women are frequently stereotyped as more collaborative, but this stereotype alone provides insufficient explanation for why Jason's collaborations with Samantha and Kim were more equitable.

6.1.6 *Summary of Alternative Hypotheses*

While there were a number of hypotheses that could not be eliminated, none of the hypotheses described above explained the differences we observed to our satisfaction. Most promising was the hypothesis that Samantha and Kim both held beliefs about engaging equally with a partner. While neither Aaron nor Peter demonstrated these beliefs, the absence of these beliefs seemed to be an insufficient explanation for the similarly inequitable Aaron-Jason and Peter-Jason dyads.

6.2 Qualitative Evidence of a Speed Focus

6.2.1 *Command-Clarify Sequences*

In our quantitative analysis of the transcripts, we focused on the distribution of commands within the dyad. This was based upon our assumption that commands shape the equity of the collaboration because frequent commands may communicate a lack of respect for the partner being commanded. Below we present examples from the Aaron-Jason and Peter-Jason dyads to demonstrate one of the prevalent patterns, *command-clarify sequences*. We interpret these interactions as evidence of a focus on speed. These interactions additionally suggest a focus on accurately completing the tasks, but in examining the use of shortcuts we see counterexamples showing that the dyads were not focused on accurately completing the tasks.

6.2.1.1 *Aaron-Jason: Command-Clarify Sequences*

There is ample evidence that Jason and Aaron's interaction was dominated by Aaron issuing Jason commands [35]. Throughout the ninety minute episode Aaron issued 108 commands, which accounted for 23% of all of Aaron's statements during the interaction. We described the dominant pattern as *command-clarify sequences*, in which Aaron issued commands and Jason occasionally clarified content from the commands.

The following transcript shows a prototypical example of the *command-clarify sequence*. Immediately before the following transcript, Jason and Aaron acknowledged that the code did not work as intended. The example below begins with Jason making a suggestion of how they can change the picture they drew to achieve the goal.

105 Jason: Just gotta move - this - over.
106 Aaron: Oh! I got an idea.
107 Aaron: So completely take that second script off.
108 Aaron: From the "Go to y 55 x 55" (referring to a block)
109 Aaron: No.
110 Aaron: No. Yeah - And take - and take that block off the blue block at the bottom.
111 Aaron: Run that script.
112 Jason: This script?
113 Aaron: That - uh - the C. C. (referring to a script that starts when you press the C key)
114 Jason: C?
115 Aaron: Yeah
116 Jason: Okay.
117 Aaron: Okay - then.
118 Aaron: Now, now move the cat away. (pause)
119 Aaron: So now. Select image.

This excerpt serves as an example of the dominant interaction pattern within the Jason-Aaron dyad, which we refer to as command-clarify sequences. In the interaction above, we identified seven of Aaron's statements as commands where he appears to be directly telling Jason what to do (Lines 105, 107, 110, 111, 113, 118, 119). We classified Jason's two questions ("This script?" and "C?" Lines 112 and 114) as clarifications of Aaron's commands. Of Aaron's remaining three statements, two were responses to Jason's questions ("No" and "Yeah" Lines 109 and 115) and the third "Okay - then." appears to be an incomplete command. Although we have not classified these statements as commands, they contribute to the command-clarify pattern.

Most notably these statements (i.e., "No" and "Yeah" Lines 109 and 115) are noteworthy in that Jason's clarifying questions did not elicit explanations from Aaron. Similarly, Aaron did not respond to Jason's suggestion "Just gotta move - this - over." (Line 105). Instead, Aaron said "Oh! I got an idea." (Line 106), but did not explain the idea and only provided commands to execute the idea. This lack of an explanation happened at other times within the interaction. Most notably, in the following excerpt, Aaron ignored Jason's request for more information.

069 Aaron: I've got an idea that is gonna make it faster. (pause)
070 Jason: How do you know?
071 Aaron: Trust me, it's gonna make it exactly two times as fast.

Aaron's statement "Trust me" (Line 071) is blatant in not providing an explanation. This highlights a central feature of the command-clarify pattern: the lack of an explanation. Due to space constraints, we have not included contrasting examples from the Samantha-Jason and Kim-Jason dyads where Jason's clarifying questions elicited explanations.

Overall, the pattern appears to provide minimal collaboration. The dominant pattern of command-clarify appears to be optimized for having Jason quickly construct and test programs. Although Aaron and Jason made distinct contributions to this pattern, the pattern was one that prioritized completing tasks and appears to compromise intellectual engagement.

6.2.1.2 *Peter-Jason: Command-Clarify Sequences*

Similarly, the Peter-Jason dyad included examples of the command-clarify pattern. In the following exchange, Peter was giving Jason, the driver, instructions. Peter's instructions include references to Logo commands "Forward" and "RT," which move the character forward and right, respectively.

326 Peter: Not at the end!
327 Peter: Forward 1, RT 1.
328 Peter: Down there.
329 Peter: You're doing it wrong, there's another (unclear speech).
330 Jason: Here?
331 Peter: No, not that.
332 Jason: Here?
333 Peter: Yeah, basically.

When Jason was driving, command-clarify sequences were less common than command sequences that had no clarifying questions. These command sequences sometimes resulted in a dispute between Peter and Jason. For example, in the following exchange, Peter issued commands with increasing intensity and then Peter and Jason both raised their voices and appeared agitated with each other. In the excerpt below, Peter and Jason were trying to run their "square" function in Logo. Peter appears to realize (Line 218) that to run the function Logo "square" you type "square" even though when you are defining the function in Logo you type "to square."

216 Peter: Now you type "to square."
217 Peter: You have to press enter
218 Peter: Oh it's just "square."
219 Peter: It's JUST SQUARE. (sounds exasperated)
220 Peter: Just "SQUARE," not "to square."
221 Peter: It's "square" dude. (30 second pause)
222 Peter: Told you.
223 Jason: Told me what?
224 Peter: You're not supposed to use "to"!
225 Jason: You never said that.
226 Peter: Yes they did.
227 Peter: Go to that - go to the curriculum.
228 Peter: You're supposed to have a "to square"! (raised voice)
229 Jason: That's what I did! (raised voice)
230 Peter: I'm just saying, you don't type "To square" up there.
231 Jason: Stop yelling.

In the exchange above, Peter's insight about the difference between running and defining functions was correct. He attempted to explain that it is "just 'SQUARE,' not 'to square.'" (Line 220). However, from this interaction it does not appear that Jason understood Peter's point. Near the end of the exchange, Jason retorted, "That's what I did!" (Line 229) and there is no evidence from the audio that Jason understood Peter's point.

Throughout the interaction, Peter seemed focused on quickly completing the task of running the function square. It is noteworthy, that here Peter attempted to explain his command (Line 230). However, Jason's response of "stop yelling" (Line 231) suggests that Peter's tone may have been consistent with a pattern of marginalization.

6.2.2 Shortcuts

A second pattern that we observed within the Aaron-Jason and Peter-Jason dyads was taking shortcuts. These shortcuts served to speed the dyad's progress through the curriculum by leaving required steps incomplete or subverting the intended challenge of the task. Shortcuts are clearly consistent with a goal for speed. However, they are also consistent with a disregard for accurately completing the tasks. Therefore this seems to reinforce our claim that command-clarify sequences are a result of a focus on speed and not, as our recent alternative hypothesis suggests, a focus on accurately completing the tasks. The examples below demonstrate the nature of the shortcuts the dyads pursued and some of the interactions that accompanied these shortcuts.

6.2.2.1 Aaron-Jason: Shortcuts

There were four clear examples where Aaron was taking, or was directing Jason to take, a shortcut.

While Aaron was driving and Jason was navigating, the pair was trying to draw a five-sided star (Lines 199-229). In determining the amount to turn between each side, the students tried multiple values, most of which were just a little off the correct value of 144. After a particular modification to the angle, Aaron said, "Yes!" (Line 217), but Jason realizes it was not correct and seemed to suggest trying 146. Jason interjects "Eh - you're off just a tiny bit - you should put 6. Try 6," (Line 218-220). While Jason said this, Aaron interjected "I don't care" (Line 219), which we interpret as expressing an intention to move on even though the angle of the star was not correct. However, Aaron appeared to concede, next saying, "Fine" (Line 221).

A second shortcut took place when Aaron was driving. When Aaron went to upload the Scratch project, Jason objected by saying, "Hey, it didn't finish" (Line 260) Aaron responded by saying, "I don't care" (Line 261). From this interaction, we infer that Aaron did not wait for the Scratch project to complete the drawing (i.e., finish executing the code) before he began uploading the file. Within this interchange, Aaron's behavior appears consistent with a goal for speed.

In two other instances, Jason described Aaron's behavior as "cheating" (Lines 145 & 378) and these appear to mark instances when Aaron was taking a shortcut in lieu of completing the assigned task as intended. In one case (Lines 106-145), Aaron avoided creating a complex script to draw a picture by superimposing screenshots from a previous problem. In a second case (Line 378), Aaron used the paint editor to avoid creating a script to draw a football. This second case is noteworthy because Jason had two different responses to Aaron's actions. When this happened, Jason accused Aaron of "cheating" (Line 378), but when questioned about it by a teacher, Jason claimed with faux ignorance, "He drew it." (Line 382) These cases generally support the hypothesis that Aaron had a goal of working through the activities as quickly as possible.

These shortcuts appear to be a clear indication of Aaron's focus on speed. Jason does not appear to condone these shortcuts except when he defends one of Aaron's shortcuts to a teacher.

6.2.2.2 Peter-Jason: Shortcuts

There were fewer instances in the Peter-Jason dyad in which they pursued shortcuts. Of two clear examples, Jason challenged one and Jason initiated one.

In the first example of a shortcut, Peter and Jason had just successfully drawn a pentagon in Logo and Peter was the driver. After it completed drawing, Peter said "Yay!" (Line 279). Jason responded by noting "It's sideways." (Line 280), presumably because the pentagon had a different orientation than the pentagon shown in the assignment. Peter dismissed Jason concern by saying "Whatever." (Line 281). In response, Jason mocked, "Everything is 'whatever, whatever, whatever.'" (Line 282). Peter did not respond to this verbally and continued saving the file and then moved on to the next step in the curriculum and exclaimed "Circle!" (Line 284).

In the second example of a shortcut, Jason suggested moving on to the next activity before completing the current activity. Jason said, "Do you just want to save and keep going?" (Line 511). Peter rejected the proposed shortcut and responded "No! I know what's wrong." (Line 512). They proceeded to discuss the issue and eventually received assistance from one of the course instructors.

6.2.3 Peer Comparison

Overall, though, the primary way that a focus on speed manifested in the Peter-Jason dyad highlights a different theme: peer-comparison. This played out in two ways: competitiveness with other classmates, and competitiveness with each other. These peer comparisons were unique to the Peter-Jason dyad. In the following transcript. Jason compared Peter's and his progress on that day's curriculum against that of a nearby pair of classmates. By noting several times that these classmates were further along than Peter and him, Jason's comments indicate a focus on speed.

556 Jason: Wait, you guys finished already?! (directed at classmates nearby)
557 Jason: They finished!
558 Teacher: Okay, we're getting back to work (attempting to get Jason to re-focus)
559 Jason: You guys finished? (directed at the same classmates nearby)
560 Classmate: Yeah.

Immediately following this, Jason asked the same group how much progress they had made on the next part of the curriculum, programming the computer to make different letters.

His first comment, "Wait, you guys finished already?!" (Line 556), suggests that Jason was initially surprised and perhaps envious that another team was moving faster than his team. Even after the teacher attempted to re-focus Jason (Line 558), Jason persisted in asking these students whether they had finished, which is followed by a third, more specific inquiry into his classmates' progress, "What letters did you finish" (Line 563). The frequency and nature of Jason's inquiries suggests that he is anxious that he is falling behind, and that being the first to complete a task is desirable. Although this was the only exchange of this kind in the data from this pair, it does indicate that speed was something Jason valued. There was no evidence that Peter was also attending to his classmates' progress, but we saw evidence of Peter's focus on speed as he competed with Jason.

There was considerable evidence that Jason and Peter were competitive with each other, and typically this manifested in the form of the students comparing their progress on class assignments. For example, each class period began with a daily written warm-up. After a few minutes of working on it, Jason and Peter discussed how many of the problems each of them was able to complete. Later, Jason and Peter discuss and compare the progress they have made on their final projects, as well as the sophistication of their projects. Students' progress on the final projects could have been a source of anxiety, especially for a lower-performing student like Jason. Overall, these exchanges indicate a speed-orientation that was embraced at different times by both Jason and Peter. Similar kinds of interactions took place during the previous week on Day 5 when Jason and Peter were also sitting next to each other, but were not pair programming.

7. DISCUSSION

We investigated a number of hypotheses to see what could explain the patterns of equity and inequity we observed. Our primary hypothesis was that a focus on speed contributed to inequity within the Aaron-Jason and Peter-Jason dyads. This appeared to be the common thread among the dominant patterns of: command-clarify sequences, shortcuts, and peer comparison. The command-clarify sequences could also be an indication of a focus on accurately completing the assignments, but the prevalence of shortcuts, which involve not accurately completing the assignments, weakens this alternative hypothesis.

The Samantha-Jason and Kim-Jason dyads did not appear to pursue speed, which may explain their more equitable interactions. However, it is likely that the more equitable relationships observed in the Samantha-Jason and Kim-Jason dyads were influenced by the more playful tasks and Samantha and Kim's pro-social attitudes toward collaboration.

Our finding that there were gross inequities in the Aaron-Jason and Peter-Jason dyads is surprising because the classroom practices were designed to support equity [34]. For example, the pair-programming structures were carefully designed to ensure that both students had equal time as driver and navigator, and that students reliably switched roles of driver and navigator.

Using a self-paced curriculum was also designed to promote equity, but may have inadvertently contributed to the students' focus on speed and the resulting inequitable interactions. The self-paced curriculum was intended to provide students the opportunity to progress at their own pace, while offering daily synchronization points so that all students were exposed to the same material [34].

The students were allowed to progress through the curriculum at their own pace. We intended to promote mastery of the material rather than a prescriptive pace for all students. By design, the self-paced curriculum meant that students were often working on a range of different steps within the curriculum. Unfortunately, these differences in progress were visible to the students, by observing the computer monitor of their surrounding peers. The public dimension of this self-paced curriculum may have further focused students on the goal of speed.

These synchronization points took the form of open-ended projects at the end of each three-hour session. The curriculum was designed so that these open-ended projects would reinforce, but not introduce, content. Although students were exposed to all of the content even if they spent little time on these open-ended projects, students in the class valued getting time on these open-ended projects. This use of open-ended projects and the resulting value system may have promoted the students' focus on speed. In future work we plan to examine how classroom practices shape students' goals, and how these goals relate to patterns of equity and inequity. The attempt to use a self-paced curriculum may have provided more challenges than it addressed.

8. CONCLUSION

This paper builds primarily upon two lines of prior work: research on equity and research on collaborative learning. Our findings complement research focused on issues of structural equity/inequity within CS [35, 14, 24, 47] by focusing on equity at the level of interactions. Additionally, this work connects research on equity to research focused on collaborative learning, which includes research on pair programming.

Beyond the impact of equity and inequity on students' opportunities to learn, there are broader moral and political reasons to care about equity in the collaborative context. Boaler [2] has argued that how students treat each other in classrooms relates to how they end up treating people in society, as they become adults. To the extent that students can learn to respect their classmates and value their diversity of perspectives and strengths—as is necessary in collaborative learning situations—students put themselves in a better position to be good citizens later in life. In this sense, equity in collaborative learning contexts is about more than students' access to opportunities to learn content, but it is also related to the kinds of societies we hope for outside the classroom walls.

9. ACKNOWLEDGMENTS

We would like to thank the students and staff of the summer enrichment program. The research reported here was supported in part by grants from the National Science Foundation (#1044106 & #1339404), and the Institute of Education Sciences pre-doctoral training grant R305B090026 to the University of California, Berkeley. The opinions expressed are those of the authors and do not represent views of the National Science Foundation or the Institute of Education Sciences.

10. REFERENCES

[1] Ames, C. (1992). Classrooms: Goals, structures, and student motivation. *Journal of educational psychology, 84(3),* 261.

[2] Barron, B. (2003). When smart groups fail. *Journal of the Learning Sciences, 12*(3), 307-359.

[3] Boaler, J. (2008). Promoting 'relational equity' and high mathematics achievement through an innovative mixed-ability approach. *British Educational Research Journal*, 34(2), 167-194.

[4] Braught, G., MacCormick, J., & Wahls, T. (2010, March). The benefits of pairing by ability. In*Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 249-253). ACM.

[5] Cockburn, A., & Williams, L. (2000). The costs and benefits of pair programming. Extreme programming examined, 223-247.

[6] Cohen, E. G. (1994). *Designing groupwork*. New York, NY: Teachers College Press.

[7] Cohen, E. G., & Lotan, R. A. (1995). Producing equal-status interaction in the heterogeneous classroom. American Educational Research Journal, 32(1), 99-120.

[8] Chizhik, A. W. (2001). Equity and status in group collaboration: Learning through explanations depends on task characteristics. Social Psychology of Education, 5(2), 179-200.

[9] Corbin, J., & Strauss, A. (2014). Basics of qualitative research: Techniques and procedures for developing grounded theory. Sage publications.

[10] Darling-Hammond, L. (2010). The flat world and education: How America's commitment to equity will determine our future. New York, NY: Teachers College Press.

[11] DeClue, T. H. (2003). Pair programming and pair trading: effects on learning and motivation in a CS2 course. *Journal of Computing Sciences in Colleges,18(5),* 49-56.

[12] Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair Programming: Under What Conditions Is It Advantageous for Middle School Students?. *Journal of Research on Technology in Education*, *46*(3), 277-296.

[13] Engle, R. A., Langer-Osuna, J. M., & McKinney de Royston, M. (2014). Toward a model of influence in persuasive discussions: Negotiating quality, authority, privilege, and access within a student-led argument. Journal of the Learning Sciences, 23(2), 245-268.

[14] Ericson, B., & Guzdial, M. (2014, March). Measuring demographics and performance in computer science education at a nationwide scale using AP CS data. In Proceedings of the 45th ACM technical symposium on Computer science education (pp. 217-222). ACM.

[15] Esmonde, I. (2009). Mathematics learning in groups: Analyzing equity in two cooperative activity structures. *Journal of the Learning Sciences, 18*(2), 247-284.

[16] Fawcett, L. M., & Garton, A. F. (2005). The effect of peer collaboration on children's problem-solving ability. British Journal of Educational Psychology, 75(2), 157-169.

[17] Flores, A. (2007). Examining disparities in mathematics education: Achievement gap or opportunity gap? The High School Journal, 91(1), 29–42.

[18] Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in education: a literature review. *Computer Science Education, 21(2),* 135-173.

[19] Jermann, P., & Nüssli, M. A. (2012, February). Effects of sharing text selections on gaze cross-recurrence and interaction quality in a pair programming task. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (pp. 1125-1134). ACM.

[20] King, A. (1991). Effects of training in strategic questioning on children's problem solving performance. *Journal of Educational Psychology, 83*(3), 307-317.

[21] Langer-Osuna, J. M. (2011). How Brianna became bossy and Kofi came out smart: Understanding the trajectories of identity and engagement for two group leaders in a project-based mathematics classroom. *Canadian Journal of Science, Mathematics and Technology Education*, 11(3), 207-225.

[22] Lave, J., & Wenger, E. (1991). Situated learning: Legitimate peripheral participation. Cambridge university press.

[23] Lewis, C. M. (2011). Is pair programming more effective than other forms of collaboration for young students?. *Computer Science Education,* 21(2), 105-134.

[24] Margolis, J., Estrella, R., Goode, J., Holme, J. J. and Nao, K. 2008. Stuck in the shallow end: Education, race, and computing. MIT Press, Cambridge, MA.

[25] McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002, February). The effects of pair-programming on performance in an introductory programming course. In ACM SIGCSE Bulletin (Vol. 34, No. 1, pp. 38-42). ACM.

[26] McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. Communications of the ACM, 49(8), 90-95.

[27] Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C., & Balik, S. (2003, February). Improving the CS1 experience with pair programming. In ACM SIGCSE Bulletin (Vol. 35, No. 1, pp. 359-362). ACM.

[28] Nattiv, A. (1994). Helping behaviors and math achievement gain of students using cooperative learning. The Elementary School Journal, 285-297.

[29] Oakes, J. (2005). Keeping track: How schools structure inequality. New Haven, CT: Yale University Press.

[30] Palinscar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. Cognition and instruction, 1(2), 117-175.

[31] Salleh, N., Mendes, E., & Grundy, J. (2011). Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. Software Engineering, IEEE Transactions on, 37(4), 509-525.

[32] Saxe, G. B. (1988). Candy selling and math learning. Educational Researcher, 17(6), 14-21.

[33] Sfard, A., & Kieran, C. (2001). Cognition as communication: Rethinking learning-by-talking through multi-faceted analysis of students' mathematical interactions. Mind, Culture, and Activity, 8(1), 42-76.

[34] Shah, N., Lewis, C. M., Caires, R., Khan, N., Qureshi, A., Ehsanipour, D., & Gupta, N. (2013). Building equitable computer science classrooms: elements of a teaching approach. In ACM SIGCSE Bulletin. 263-268.

[35] Shah, N., Lewis, C. M., & Caires, R. (2014) Analyzing Equity in Collaborative Learning Situations: A Comparative Case Study in Elementary Computer Science. *Proceedings for the 11th International Conferences of the Learning Sciences (ICLS)*. 495-502.

[36] Thomas, L., Ratcliffe, M., & Robertson, A. (2003, February). Code warriors and code-a-phobes: a study in attitude and pair programming. In ACM SIGCSE Bulletin (Vol. 35, No. 1, pp. 363-367). ACM.

[37] Vivekanandan, K., & Kuppuswami, S. (2004). The effects of pair programming on learning efficiency in short programming assignments. Informatics in Education-An International Journal, (Vol 3_2), 251-266.

[38] Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes. Cambridge: Harvard University Press.

[39] Watkins, K. Z., & Watkins, M. J. (2009). Towards minimizing pair incompatibilities to help retain under-represented groups in beginning programming courses using pair programming. *Journal of Computing Sciences in Colleges*, *25*(2), 221-227.

[40] Webb, N. M. (1991). Task-related verbal interaction and mathematics learning in small groups. *Journal for Research in Mathematics Education*, 22(5), 366-389.

[41] Webb, N. M., Farivar, S. H., & Mastergeorge, A. M. (2002). Productive helping in cooperative groups. Theory into practice, 41(1), 13-20.

[42] Wenger, E. (1998). Communities of practice: Learning as a social system. Systems thinker, 9(5), 2-3.

[43] Werner, L., McDowell, C., & Denner, J. (2013, March). Middle school students using Alice: what can we learn from logging data?. *In Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 507-512). ACM.

[44] Werner, L., & Denner, J. (2009). Pair programming in middle school: What does it look like?. *Journal of Research on Technology in Education, 42(1),* 29-49.

[45] Wertsch, J. V. (1998). Mind as action. New York, NY: Oxford University Press.

[46] Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. IEEE software, 17(4), 19-25.

[47] Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). Running on empty: The failure to teach K-12 computer science in the digital age. Association for Computing Machinery. Computer Science Teachers Association.