Remote Pair Programming in Online CS Education: Investigating through a Gender Lens

Sandeep Kaur Kuttal
University of Tulsa
Tulsa, Oklahoma, USA
sandeep-kuttal@utulsa.edu

Kevin Gerstner
University of Tulsa
Tulsa, Oklahoma, USA
kag8789@utulsa.edu

Alexandra Bejarano
University of Tulsa
Tulsa, Oklahoma, USA
alb146@utulsa.edu

***Abstract*—Online CS education shows many gender-inclusivity problems in practice and through tools, and little has been done to address this. In CS classrooms, pair programming has been shown to significantly help women and men understand and appreciate programming concepts, and to help close gender gaps. Unfortunately, pair programming is not well supported in online CS education, especially when one of the pair members is a woman. Our overall objective is to address this gap by investigating the following question: How can we bring the educational benefits of pair programming to online CS students in gender-equitable ways? In this paper, we empirically investigate whether and how technology-mediated remote pair programming hinders online students of same- and mixed-gender pairs. Based on our results, we propose refining personas and the cognitive walkthrough to include leadership styles and preferences for pair-programming roles. We further recommend features for online CS educational tools to promote gender-inclusiveness.**

I. Introduction

Although our society has realized the importance of Computer Science (CS) education, 65% of the US population (e.g., women and people of color) is poorly represented in CS classrooms [1]. Online education is one potential solution towards promoting CS education for diverse or remote populations around the world.

However, online CS education is no more gender-inclusive than in-person CS education. In fact, much of what has been achieved so far in CS education's gender-inclusiveness is situated in the classroom. Little has been done to consider gender-inclusivity of online CS education platforms.

For gender-inclusivity in CS classrooms, one effective mechanism is pair programming. In pair programming, two people help each other program as two heads are better than one [2], [3]. The two programmers work collaboratively on the same design, algorithm, code, or test [4], [5]. Sitting shoulder-to-shoulder at a computer, one member of the pair is the driver, who actively creates code and controls the keyboard and mouse. The other member is the navigator, who constantly reviews the driver's work, proposes suggestions, and asks clarifying questions [6]. After a designated period of time, the partners switch roles. Pair programming has been shown to improve the pair members' quality, teamwork, communication, knowledge management, and morale [7]–[17].

978-1-7281-0810-0/19/$31.00 ©2019 IEEE

In CS education, pair programming has been shown to help students understand programming concepts and organize their own knowledge as a result of interactions and alternating roles [18]–[21]. It reduces the gender gap: it particularly increases women students' confidence in programming, encourages them to pursue computer science, and increases their Information Technology (IT) fluency – contemporary IT skills, fundamental grasp of IT concepts and intellectual IT capabilities [8], [22]–[25].

Unfortunately, pair programming is not well supported in online CS education, especially when one of the pair members is a woman. This leads to the central question: *How can we bring the educational benefits of pair programming to remote online CS students in gender-equitable ways?*

To find out how to support remote CS students via pair programming, we draw upon three factors – coordination, communication, and collaboration – that are at the heart of software development processes among remote teams [26]–[29]. To investigate how gender differences manifest in these three factors for remote CS students' pair programming experiences, we formulated the following research questions:

- *RQ1: How might different gender pairings of remote CS students coordinate?*
- *RQ2: How might different gender pairings of remote CS students communicate?*
- *RQ3: How might different gender pairings of remote CS students collaborate?*

II. Related Work

Remote pair programming, also known as distributed pair programming, has been known to be comparable to co-located pair programming in regards to student performance, effective code quality and productivity [30]–[32]. Tsompanoudi et al. found that the individual with more programming experience and confidence secured higher assignment grades while the individual with lower perceived pair compatibility secured higher exam grades [33]. In the literature review by Da Silva et al., concerns were raised about the inadequate number of studies related to distributed pair programming and a recommendation for more studies on coordination and communication was made [34].

Co-located pair programming has proven to reduce the gender gap by increasing female students' confidence in pro-

TABLE I
GENERAL DEMOGRAPHICS

Pair#	Gender	Level	Age	Experience	
				Prog.	Pair-Prog.
P1M1	Man	Junior	19-23	2 Years	Yes
P1M2	Man	Sophomore	19-23	2 Years	Yes
P2M3	Man	Senior	19-23	>4 Years	No
P2M4	Man	Freshman	19-23	3 Years	No
P3M5	Man	Freshman	19-23	4 Years	No
P3F1	Woman	Senior	19-23	3 Years	No
P4F2	Woman	Junior	19-23	2 Years	No
P4F3	Woman	Senior	19-23	2 Years	Yes
P5F4	Woman	Masters	30-40	2 Years	No
P5F5	Woman	Masters	19-23	2 Years	No
P6M6	Man	Senior	19-23	4 Years	No
P6F6	Woman	Senior	19-23	3 Years	Yes

gramming and encouraging them to pursue CS [17], [25], [35]. In studies, women reported pair programming helped them be more engaged, learn more from peers, feel less frustrated, build confidence, and make new friends [36]. Furthermore, studies on co-located pair programming have found differences between same- and mixed-gender student pairs in collaboration and communication [37], [38].

Prior research on pair programming has investigated gender effects only among co-located pairs. To the best of our knowledge, none of the studies have analyzed remote pairs to identify the hindrances CS students face while working remotely in same- or mixed-gender pairs. In this study, we conducted a controlled, multiple-room lab study with same- and mixed-gender CS students pair programming remotely. We identified hindrances CS students face while working remotely (coordinating, communicating, and collaborating) in pairs on their programming assignments.

In this paper, as with [39], [40], gender is a person's gender identification. We use the term "men" as a shorthand for "people who identify as men", and "women" to denote "people who identify as women." We focus on only those two genders. In future work we would like to investigate other genders, but as a first step we focus on the two genders with the highest numbers to feasibly measure the impact of our work.

III. METHODOLOGY

A. Participants

In this study, two CS students collaborated with each other remotely using pair programming to develop a simple Java program. Hence, the selection criteria was a university student with basic object-oriented programming experience. We selected participants on a first-come, first-served basis and constrained the number of participants to 6 men and 6 women to maintain a gender balance. Furthermore, we paired the participants into same- (2 man-man, 2 woman-woman) and mixed- (2 woman-man) gender pairs. Table I shows the general demographics of the participants. All participants were given a $20 gift card as compensation for their time.

B. Study Design

Participants of a pair were placed in separate rooms for the study to mimic a remote pair programming environment. The participants filled out a background and self-efficacy questionnaire, followed by a tutorial about pair programming and the think-aloud method. With the think-aloud method [41], [42], participants were asked to vocalize their thoughts and feelings as they performed their tasks.

Once participants were introduced to the concepts, they were asked to complete a 10 minute Simple Task to promote pair jelling. Pair jelling is an adjustment period when pairs first start working together; after the pair has jelled, they perform tasks considerably more efficiently [12], [43]. Participants used the Java IDE Eclipse [44] and TeamViewer [45]. TeamViewer provided video, voice, and text chat and is commonly used in industry with 1.8+ billion devices connected [45]. The main communication mechanism was video, but text chat was available for participants to use throughout the study.

The participants were then given the Main Task to implement a Tic-Tac-Toe game within a 40 minute time limit. The time limit helped us keep the entire study session below 90 minutes and avoid fatigue for our participants. We video recorded participants' facial reactions, audio, and screen interactions using Morae [46], a screen capture and replay tool.

After the study, we conducted a retrospective interview using pre-determined questions (refer Table II) to get insights about the participants' experiences. These interview sessions were audio recorded. Finally, participants were asked to fill out a post self-efficacy questionnaire and pair programming questionnaire. Study material can be found at [47].

C. Tasks

The **Simple Task** was to look at a password checking implementation in Eclipse and fix the password validation test using pair programming.

The **Main Task** was to implement a Tic-Tac-Toe game for two players, X and O, who take turns marking spaces in a 3x3 grid. Participants needed to add code and respective test cases to finish the task. The participants were given an implementation of the board and a test case. This was selected as a task because (1) it is simple to implement for anyone with Java experience and (2) does not require domain knowledge to understand the requirements and generate test cases.

D. Qualitative Analysis

Videos/Audio from the Main Task and retrospective interviews were transcribed. Transcripts of participants from the Main Task were split into 30-second segments and coded, allowing multiple codes per segment. Three researchers independently coded 20% of the transcripts and reached agreement on 90% of the coded data by calculating inter-rater reliability using the Jaccard measure [48]. Then, the researchers split the remaining code between them.

Code sets used were: (1) Roles - Navigator, driver (2) Leadership style - Democratic, authoritative, laissez-faire, paternalistic [49], [50] (3) Non-Verbal Cues - Smile, laughter, confusion, arms crossed, clasped hands, imitation [51] (4) Task management - Taking control, dividing and conquering, releasing control, instructing partner (derivations of code sets

TABLE II
SAMPLE RETROSPECTIVE INTERVIEW QUESTIONS

You can trust partner better when?...(certain characteristics)
You can collaborate better when?...
Do you feel like your partner interrupted you? Did it interrupt your workflow?
How did you decide when to switch Navigator/Driver roles? When do you want to be navigator? When do you want to be driver?
You can trust your partner better when?..
Would you prefer partnering with a man or woman?
Would you prefer remote pair programming or in-person?

is further discussed in IV). We used thematic analysis to organize our data. Thematic analysis [52] is a method used for organizing qualitative data into themes which relate back to the research question.

E. Limitations

Every study has limitations. One limitation of this study is the small sample size of 6 pairs, although the participants are students and have similar demographics to online CS students. Another limitation is a possibility that students were aware of their partner before the study since they were from the same university. Lastly, we mimicked the remote environment by placing participants in separate rooms instead of geographically different locations. This may not be representative of a real remote environment. Limitations like these can only be addressed through further empirical studies.

IV. RESULTS

Ellis et al. [53] proposed that group interactions are functionally divided into three dimensions: coordination, communication and collaboration. These dimensions are formulated as research questions to identify hindrances faced by same- and mixed-gender pairs of CS students while working remotely on their programming assignments.

A. RQ1: How might different gender pairings of remote CS students coordinate?

In remote pair programming, a pair's coordination may be influenced by individual leadership and interruption styles.

1) Leadership Style: Previous studies [50], [54]–[65] have shown that men and women employ different leadership tactics. For example, many women tend to be more democratic (sharing decision making abilities), whereas men tend to be more authoritative (dominating the interactions) [49], [50]. These gender differences highlight a difficult challenge to overcome for pair programming. In our study, we found more instances (304 of 390 dialogues) of authoritative and democratic leadership styles than laissez-faire (where all decision-making is given to the partner) and paternalistic (when one partner teaches the other). Therefore, we narrowed our analysis to only consider authoritative and democratic leadership styles.

All four same-gender pairs had a democratic leadership style while, in both mixed-gender pairs, one of the participants was authoritative (refer Fig. 1). This suggests that there may be conflict between partners of different genders because of their

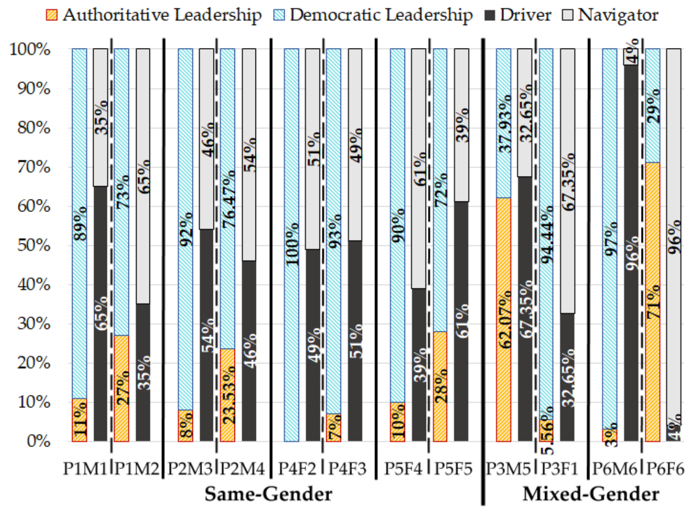

Fig. 1. Percentage same- and mixed-gender pairs spent in leadership styles (Authoritative/Democratic) and pair-programming roles (Driver/Navigator).

leadership styles. Hence, we investigated the effect leadership style has on driver/navigator roles and task management.

Leadership Style vs. Driver/Navigator Roles: Pairs that were more democratic spent roughly equal time in both driver/navigator roles, while pairs with a more authoritative partner had an imbalance of time in each role (refer Fig. 1). For example, in P4, both participants were democratic (P4F2 - 100%, P4F3 - 93%) and spent nearly 50% of the time in both driver/navigator roles. Contrary to this, in P6, one participant was authoritative (P6F6 - 71%) and was the navigator 96% of the time. In pair programming, a navigator and driver work together in a role-based protocol and switch roles regularly, with equal time in each role. These results highlight a potential problem for pair programming – there is an apparent imbalance in the sharing of responsibilities when one partner is authoritative and the other is democratic. This has a direct implication on gender roles.

In the mixed-gender pairs of our study, an authoritative man was more often a driver (P3M5) and an authoritative woman was a navigator (P6F6). In contrast, for same-gender pairs the more authoritative man was frequently a navigator while the more authoritative woman was a driver (refer Fig. 1). In interviews, it was revealed that women participants preferred to be the navigator and men participants preferred to be the driver when they knew what to do (we go into depth regarding this in Section IV-B2). Because the men and women in our study preferred different roles, mixed-gender pairs tended not to share driver/navigator roles while same-gender pairs went beyond their own preferences and shared roles and responsibilities with their partners.

Leadership Style vs. Task Management: We observed the effect of leadership styles on how CS students manage task control. Table III shows the frequency of strategies used by each pair. The strategies observed were:

Taking control of the task implicitly or explicitly. In all six pairs, we observed participants take control over the task without consulting their partner. There were two ways this happened: implicitly, in which one participant began working on the task without saying anything, and explicitly, in which one participant said they were going to take control, but still without consulting their partner.

In democratic-democratic pairings, which were mostly same-gender pairs, we found that both participants had the same amount of task control between them even when their self-efficacy scores were vastly different. For example in P5, P5F4 took control 7 times while her partner P5F5 took control 6 times during their study session. P5F5 reported a higher self-efficacy (15) than P4F4 (-10). Similarly, in P2, P2M3 took control 3 times and his partner P2M4 took control 2 times. P3M4 reported a higher self efficacy (14) than P3M3 (8). This indicates that in our study, same-gender pairs were good team players, even with their less confident partners.

For authoritative-democratic pairings in our study, the participant with higher self-efficacy took control more frequently than their less-confident partner. In the mixed-gender pair P3, P3M5 had a higher frequency of both implicit and explicit control (14 instances) than his partner (4 instances). For instance, P3M5 took control explicitly, by saying *"That is a really good idea. Let me do that real quick"* and implicitly, by starting to type after they had finished discussing the task. P3M5 also reported a higher self-efficacy (16) than his partner P3F1 (9). Hence, in mixed-gender pairs, the "free-rider effect" occurred, which is a phenomenon where one partner with low self-efficacy is less motivated to contribute to the task [66].

Dividing and conquering the task. We found that three democratic pairs (P1, P2, P4) divided tasks with explicit discussions to determine who was responsible for the next task. For instance, in P1, both men asked each other what task they would like to do during the assignment. P1M2 said, *"So, do you want to start with the winner class or..."* and P1M1 responded by allocating the task to P1M2 as he said, *"Do you want to take care of test case is a tie?"* This strategy was not used by any authoritative-democratic pairs.

Compromise was also effective in dividing tasks. In P2, a discussion was started by P2M4 to determine who will be the driver by saying, *"Do you want to drive or do you want me to?"*, P2M3 responded, *"Uhm I can drive."*, and after some discussion, P2M3 said, *"Go ahead and drive, sounds like you know."* Even though P2M3 initially took the offer to drive, when he realized that P2M4 might have a better understanding of a portion of the code, he told P2M4 to drive. Because same-gender pairs in our study were democratic, they were able to acknowledge their partner's strengths and divide tasks to benefit the assignment.

Releasing the control of the task. All four democratic, same-gender pairs (P1, P2, P4, P5) had one person release control when they felt it was no longer their turn to manage the task. This was triggered by either politeness or uncertainty about the task. For instance, in P1, P1M2 politely asked P1M1, *"So, do you want to? (P1M1 lowers hands quickly and smiles)"* and P1M1 responds, *"Sure. (P1M2 smiles)."* This action shows that the partners trust one another and respect each other's input. In P4, P4F2 expressed concern that she had *"been typing all this time"* and thus released control to her partner. This shows an awareness towards the pair programming principles of fairness and equal learning.

TABLE III
FREQUENCY OF TASK MANAGEMENT STRATEGIES

	Same-Gender				Mixed-Gender	
	P1	**P2**	**P4**	**P5**	**P3**	**P6**
Task Control	2	5	1	13	18	2
Dividing Task	4	2	2	0	0	0
Release Control	1	1	1	1	1	0
Instruct Other	1	4	3	3	8	21

While with authoritative-democratic, mixed-gender pairs (P3, P6), partners did not share control as effectively as same-gender pairs. In P3, P3M5, who was more authoritative, offered control once by saying, *"Ok. You want to control for a second?"* However, this offer was declined by P3F1 as she was unsure about the methodologies being used, *"I'm just thinking if there's a better way to do this."* Hence, the pair programming principles were not well practiced by mixed-gender pairs.

Instructing the other to do the task. All six pairs preferred to manage tasks by giving their partner instructions or commands on what to do, but this behavior was more common with the authoritative-democratic pairs (P3, P6). For instance, P6F6 explicitly told P6M6 word for word what to type: P6F6: *"Let's create a test to see who won the game. So @test." (P6M6 types code).* P6M6: *"So @test."* P6F6: *"And then void, we can call it something as check if won."* P6M6: *"Check if won." (P6M6 types code)* P6F6: *"And then assert equals. game dot." (P6M6 types code).* This behavior of a "backseat driver," where a navigator works extremely close with the driver and gives specific actions to take was also observed by Jones and Fleming [12]. Hence, for authoritative-democratic or mixed-gender pairs in our study, the more authoritative partner tended to dictate the direction of the task.

2) Interruption Styles: We found that both same- and mixed-gender pairs interrupted equally, although in other studies on gender differences and communication styles, men interrupted women more often than men interrupted each other [67]–[69]. We found that participants of both same- and mixed-gender pairs only interrupted their partners if the interruption was necessary and relevant to the current topic. It was mentioned in the interviews that both men and women participants did not interrupt their partners most of time as they considered it impolite, did not want to disturb their partner, slept on non-important issues, or they had low-self efficacy.

B. RQ2: How might different gender pairings of remote CS students communicate?

In distributed software development, communication has been shown to be key to the success of a project [64], [70]–[72]. The communication between remote pairings may happen non-verbally, for a purpose, and using technology.

1) Communicating non-verbally: The meaning of non-verbal communication depends not only on the words being exchanged, but also on contextual information: the sender (encoder) of the non-verbal behavior, the receiver (decoder) of

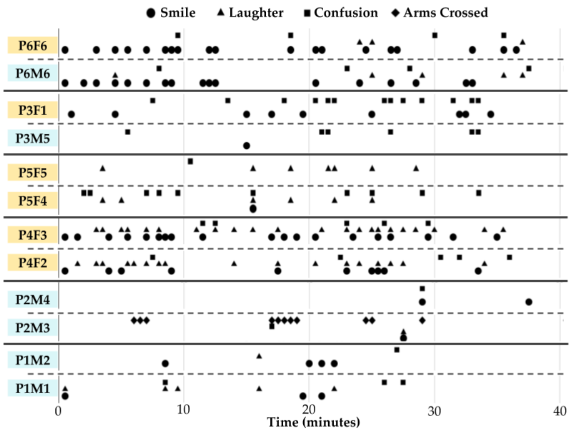

Fig. 2. Frequencies of non-verbal cues over time for all pairs. Some cues may be overlapping in the 30-second segments of videos.

that non-verbal behavior, the relationship between the sender and receiver, and the arrangement of other non-verbal cues [73]. With today's tools available to remote CS students, communication can be done via Instant Messengers (IMs) and audio or video calls, which may eliminate or greatly reduce the non-verbal communication that would be present co-located [73].

We observed gender differences in non-verbal communication as women participants used more nonverbal cues (avg 27.83) than men participants (avg 13.5). This data is consistent with previous studies on gender communication [73], [74]. Fig. 2 clearly shows women participants used more non-verbal cues than men when they were in either same- or mixed-gender pairs. Research on gender communication also suggests that women, compared to men, are more expressive non-verbally, more involved in non-verbal behaviors, and more skilled at sending and receiving non-verbal messages [74], [75].

Men and women use various non-verbal cues in different ways for different reasons [76]. Reasons that pairs utilized non-verbal cues were:

Empathy. Often, participants used smiling or laughing to empathize with their partner. We found both same- and mixed-gender pairs used non-verbal cues to demonstrate empathy, but additionally men empathized with verbal cues. For example, in same-gender pair P1, P1M1 laughed and said *"Its ugly but it should work* and P1M2 responded with a laugh and said *"Yeah, thats how I feel sometimes"*. This shows P1M2 empathizing with P1M1 using both non-verbal and verbal cues. While, in mixed-gender pair P3, P3M5 said, *"I feel like this is spaghetti code right here. (smiles)"*, his partner, P3F1, responds with *"(smiles) Just go ahead first."*. Unlike the man-man pair, P3F1 did not use verbal cues to empathize with her male partner, but did smile as a non-verbal cue. This highlights a potential difficulty for mixed-gender pairs in that their communication could be hard to understand by the other gender.

Uncertainty. All pairs used non-verbal cues (confusion and laughter) to indicate when they were uncertain either about their knowledge of the task or quality of their work. Both women participants (avg 6.5) and men participants (avg 3.0) showed confused facial emotions when they were uncertain regarding the task. P3F1, who was democratic, had shown more confusion than her authoritative partner, most likely because he did most of the code without consulting her as discussed before (in Leadership Style). A trend unique to men participants was laughing at uncertainty about their code. For instance, P1M2 said *"(laughing) I'm not going to say it looks good, but it looks working."*

Politeness. Participants often smiled or laughed to accompany a polite action such as changing roles. For example, P1M2 asked P1M1 to be driver, *"So, do you want to? (smiles)"* and P1M1 smiled and accepted. In this example, both smiled to recognize the polite action. There was no difference in the amount of politeness between genders.

Lack of interest. In the same-gender P2, there was a lot of arms crossing, a behavior that can indicate a person is not paying full attention. P2M3 often crossed their arms while P2M4 was explaining something to him or simply talking to himself. In this pair, P2M3 crossed his arms 13 times, which is much greater than any other pair. This is most likely due to the "free riders effect" as P2M3 reported lower self-efficacy scores (8) than P2M4 (14).

Imitating behavior. Mimicking body language and facial expressions is often associated with trying to establish connection and rapport with the partner [77]. In our study, men and women alike copied their partner's body language or facial expressions, but this phenomena was more common with woman-woman pairs. Consider the following exchange in P4: P4F3 said *"But how do we keep track of the first consecutive moves? (P4F3 laughs)"* and then *(P4F2 laughs).* In Fig. 2, imitation can be seen when two participants use the same non verbal cue in one time segment. The P4 pair imitated each others' smiles or laughter 34 times during the study session. This behavior was common in the woman-woman pairs, probably due to the gender dynamic of a woman-woman relationship. In contrast, when P2M4 says, *"Yeah, we don't have access to it"*, and then *(P2M3 laughs)* P2M4 doesn't laugh along with P2M3, but instead continues to think of the solution, suggesting that the man-man dynamic relies less on imitating behavior.

These results are similar to one recent study that found that more men approached communication with a clear purpose and problem to be solved, while women communicated to make rapport with their partner [78].

Solving the task. Most participants used facial expressions along with verbal communication when they successfully solved a task. In P4, both women smiled and laughed when they successfully solved parts of their task. For instance, P4F2 said, *"Yeah, yeah. Definitely. That works. Yay! (smiles)" (P4F3 laughs)*. This is in line with previous research that women tend to use more facial expressions than men [73], [74].

2) Communicating for What Purpose: One potential gender difference is what men and women are trying to accomplish with their communication. In our study, there was a difference in how men and women viewed the purpose of the driver and navigator roles.

Women participants viewed navigator as the primary communicator role. Women participants preferred to be navigator when they knew what to do so they could explain the code to the driver. As P6F6 commented, she *"would want to be the navigator if I knew more about what we were doing than my partner that way I could help them learn more..."* On the other hand, women participants preferred to be driver when they wanted someone else to lead. This was evident in an interview when P6F6 said *"... if I didn't know what I was doing I would want someone else to tell me what to do that way I could get that hands-on experience, but also have someone instructing me and teaching me how a certain thing works."* Similarly, P3F1 said she viewed driver as a passive role. Therefore, women participants viewed the navigator as the primary communicator role when they knew how to solve the problem.

Men participants viewed driver as the primary communicator role. Men participants preferred to be driver when they knew what to do so they could explain to their partner as they type. For example, P3M3 presented the driver as being the one with control as he preferred to be driver when *"you have a really good idea of what's going on and you just want to be able to pump the code out."* When they were unsure of what to do, men participants preferred the navigator role. For example, P3M5 preferred to be navigator when *"... the other person has a good idea of what's going on and you're kind of at a loss because it'll give you a little bit of time to kind of catch up to what's happening."* This highlights a possible difference between men and women: men participants viewed the driver as the primary communicator role when they knew how to solve the problem.

These results, when applied to mixed-gender pairs, seem likely to lead to frustration and a lack of trust from contrasting understandings of the driver/navigator roles.

3) Communicating Using Remote Technology: Another gender difference is in preference for remote technology when communicating in pairs. In interviews, most (5/6) of the women participants preferred in-person collaboration over remote collaboration while most men participants either preferred remote collaboration (2/6) or had no preference (3/6).

Women participants would have preferred in-person to feel connected. The reason for women participants preferring in-person collaboration was because they felt distant or disconnected from their partner during remote collaboration and found that communication in-person would be easier. As P4F2 commented, *"I would have liked to see her like next to me...I don't do well with like virtual...(be)cause it feels really weirdly disconnected...I can gauge their reactions a little better."* She also mentioned that with in-person collaboration, it is easier to explain ideas and show mistakes by directly pointing at the screen, which could save time.

Men participants preferred remote technology or had no preference. Men participants preferred remote collaboration because it was easier to switch driver/navigator roles and they felt comfortable communicating through remote technology. As P2M3 said, *"It's easier this way (remote) considering the fact that we're both, you know, stationed at the keyboard ready to go rather than having ... to switch seats, so maybe that was helpful."* Similarly, P6M6 commented *"...it was pretty I guess comfortable to work, pretty laid back and I felt like we both ... were in our ... natural coding environment ... I've done that (remote) more in my life now ... I've used like team viewer like, you know, playing video games."* Other men participants had no preference. P1M2 mentioned that he doesn't think there is much difference between remote and in-person collaboration as he feels just as comfortable talking over the internet as he does in person.

These findings suggest that men CS students may be more comfortable with online education, while women CS students may need more support from remote technology tool builders.

C. RQ3: How might different gender pairings of remote CS students collaborate?

Research about remote collaboration has shown that productive remote interactions among team members depends on creating trust and awareness [79]. Drawing from this background, we investigated how these two factors affect collaboration between CS students when they engage in remote pair programming in same- and mixed-gender pairs.

1) Trust: According to several global software development studies, trust is a key factor in the success or failure of distributed software teams [80]–[82]. Building trust between individuals is extensively studied in remote collaborations [e.g., [83], [84]]. Our participants in interviews indicated the following ways they build trust with a partner:

Built over time. Participants, both men and women, mentioned that trust was built over time as they got familiar with the coding style and habits of their partners. P3M5 said about trusting their partner that *"at the very beginning I didn't (trust) just cause it was a stranger, but as it (time) went on I was like hey... we have the same like level of coding skill and experience, I guess, so I was like, okay I can just kind of watch for a second see if she's doing what I was planning and then...I didn't have to steer so much while I was watching her."* His partner had similar thoughts, P3F1 said, *"I don't know my partner's condition and at times so I am kind of like worried like nervous but then as time goes through, I think we are fine."* This highlights the need for pair jelling [12] for both men and women to build trust.

With an open-minded and polite partner. Both men and women participants wanted to work with partners who were open to listening to their ideas. P6F6 said she would collaborate better with *"someone that's a good leader, someone that can take the reigns, but also knows when to listen and like is open to these ideas from another person."* The participants also preferred polite partners as P3M5 mentioned, *"I just want them to like be the semi polite about it (corrections), I guess. Just like as professional as you can be, you know, because we are working on a project together."*

With a confident and knowledgeable partner. Participants, both men and women, mentioned that they would easily trust a partner that is both confident and knowledgeable. P4F2 said,

TABLE IV
AVERAGE AND STANDARD DEVIATION OF SELF-EFFICACY SCORES

	Same-Gender		**Mixed-Gender**	
	Men	**Women**	**Men**	**Women**
Before Task	14.75 [4.99]	7.75 [13.18]	14.5 [2.12]	14.5 [7.78]
After Task	17.25 [2.63]	12.5 [11.12]	18.5 [1.41]	14.0 [2.83]

"If they can explain themselves and they're confident in their approach to the problem, then I trust them...If they're confident and they're wrong...then my trust in them goes down like very fast."

Hence, both men and women participants trust a partner who is open-minded, polite, confident, and knowledgeable. However, these characteristics can be gender dependent, such as less confidence due to lower self-efficacy in women [85]–[94], which can affect trust with their partner. Further research studies may reveal trust issues correlated with gender.

2) Awareness: Research into distributed software development teams shows that group awareness is challenged by temporal, geographical, and socio-cultural differences [95]–[98]. However, awareness of gender differences may also affect how CS students collaborate. We found the following awareness factors for remote pairs:

Men participants were unaware of gender differences. We found that men participants were not as gender aware as women participants. This result is consistent with a past study on same- and mixed-gender pairs in co-located pair programming by Choi [38]. The two men participants in mixed-gender pairs indicated that collaboration would be similar for a man or a woman partner. *"I think it's about the same"* and P6M6 said, *"only differences would just come from the person or like their personality...I guess the gender doesn't really matter just as long as how they think or what their ideas are..."*

Women participants preferred women partners. In contrast, women participants preferred a partner of their own gender. P4F2 indicated her preference in her partner's gender by saying, *"Definitely with a girl is easier cause with a guy I'd be like ok you can write everything, I'll just watch...well first of all it's a slightly more awkward situation when it's with a guy (be)cause there's that like disbalance already there...of skill level even though he might be like a beginner or whatever, I just implicitly might feel like he would know more for some reason. But if we started working and I realize like he doesn't understand what's going on then I would take over the control, but like I would have less time like asserting...myself at the beginning. I definitely liked working with a girl better also it's just easier for me to like talk to girls and work with girls cause I am a girl, it's just like a thing."* Similar views were shown by P5F4 as she said *"I think with girls is better because I think the guy has too much confidence and they do not allow me to have an idea, yeah. Giving an example, my husband, I'm never going to code in front of him."* This suggests that women may be more aware of how their partner's gender impacts their self-efficacy (confidence in the one's own ability to perform tasks), leading them to prefer working with another woman.

This is further supported by the average self-efficacy of men and women in same- and mixed-gender pairs as shown in Table IV. The average for men and women participants in same-gender pairs and men participants in mixed-gender pairs increased, but the average self-efficacy score of women participants in mixed-gender pairs remained nearly constant. Hence, in mixed-gender pairs, women may not take the benefit of increased individual self-efficacy from remote pair programming.

V. DISCUSSIONS

A. Co-located vs. Remote Studies

Most co-located studies [20], [21], [37], [38] conducted prior to our remote study were done quantitatively, using test scores or pass/fail percentages to determine gender differences. In contrast, we analyzed qualitative data by conducting a controlled lab study using the think-aloud method.

Two co-located studies, by Choi [38] and B. Zhong et. al [37], used qualitative surveys in their approach, and since our study is the first to use a qualitative approach for a remote setting, we chose those two studies for comparison. The following are differences and similarities we observed:

Confidence. In the co-located studies, Choi [38] found no difference in confidence between same- and mixed-gender pairs, while B. Zhong et. al [37] found that elementary school girls in same- and mixed-gender pairs, had increased confidence from pair programming. In our remote study, participants in same-gender pairs had increased confidence, while only women participants in mixed-gender pairs did not have increased confidence.

Compatibility. In post-test surveys conducted by Choi [38], there was no difference between same- and mixed-gender pairs' compatibility, while our study (in interviews) and the study by Zhong et al. [37] (in post-questionnaires) found differences in preferences to work with the other gender.

Communication. Zhong et al. [37] found that same-gender pairs exhibited a higher perceived communication than mixed-gender pairs through quantitative post-questionnaires, while in our study we analyzed communication styles in-depth qualitatively.

Partnership. In both co-located studies by Zhong et al. [37] and Choi [38] and our remote study, same-gender pairs had closer partnerships than mixed-gender pairs.

B. Implications for Gender HCI

Several commercial and open-source software products have used GenderMag (Gender Inclusiveness Magnifier) to find gender-specific issues in their software [99]–[102]. GenderMag is an inspection method that integrates a specialized cognitive walkthrough with research-based personas to statistically cluster gender differences into facets i.e. an individual's problem solving techniques and usage of software features [103], [104].

But GenderMag was created primarily to support one-user problem-solving activities with technology. Studies are needed to investigate building upon GenderMag to include new facets for collaborative activities, particularly for remote pair programming. Extending GenderMag would enable researchers and educators to evaluate remote education tools for CS students. The following discussion includes facets we observed in our study that may improve GenderMag's utility for remote pair programming:

Persona of a remote CS student. GenderMag consists of three personas: "Abby" - a stereotypical woman, "Pat(ricia)/Pat(rick)" - a gender neutral person, and "Tim" - a stereotypical man. The current persona in GenderMag uses customizable characteristics like backgrounds, photos, job titles, education, and fixed facets based on gender research. These facets are: (1) Learning styles (2) Information processing styles (3) Motivation to use technology (4) Risk-averseness (5) Self-efficacy.

We recommend larger follow-up studies be conducted to investigate the impact of adding the following characteristics to a GenderMag persona: (1) Non-verbal communication ("Abby" uses more facial expressions than "Tim") and (2) Preference for remote technology ("Tim" prefers remote technology more than "Abby"). These personas could also include facets of pair programming such as leadership style ("Tim" authoritative vs. "Abby" democratic) and primary communicator role preference ("Tim" prefers driver when he knows how to solve the problem vs. "Abby" prefers navigator when she knows how to solve the problem).

Cognitive walkthrough for pairs of CS students. There may be a need to expand the cognitive walkthrough for pair programming, such as the "goals" aspect of the walkthrough because in pair programming one individual's goal may be affected by another individual's goal.

For pair programming, we recommend follow-up studies explore replacing the cognitive walkthrough of GenderMag with group walkthrough. Group walkthrough is used for collaboration and includes two types of works: Taskwork - actions to accomplish the task and Teamwork - actions that group members must perform as a group to accomplish a task [105], [106]. Taskwork needs to be specifically designed for pair programming and Teamwork should consider same- and mixed-gender pairs because they may be affected by the combination of their leadership styles, pair programming roles (navigator/driver), and self-efficacy.

C. Implications for CS Educational Tools

Based on our results, we recommend tool builders develop the following features for CS educational tools:

Recommending pairs based on pre-questionnaires. To facilitate an optimal, gender-inclusive learning experience for CS students, a recommendation system with a set of pre-questionnaires to identify characteristics like leadership style, self-efficacy, programming style, and information processing style could be created. This system could automatically recommend pairings of CS students to course instructors, which would be beneficial especially in classes of larger sizes.

Integrating "floor passing" strategy. One way to facilitate structured coordination between two CS students is to utilize the "floor passing" strategy. A system could grant exclusive access to the current "floor holder"/driver and prompt users to change roles. This method could help especially for mixed-gender pairs to promote fairness and equal learning.

Supporting holographic video-conferencing. For better communication among remote pairs, we suspect that augmented reality, such as projecting the holographic image of the partner in the real environment, could help CS students, especially women, utilize their non-verbal cues while communicating in a way that 2D video-conference calls cannot.

Idea management and note-taking. Idea management and note-taking can be supported by adding the feature of a shared, interactive "to-do" list, a form of information enrichment that has been shown to be effective for and well-liked by both men and women [107]–[109]. The "to-do list" could help facilitate dividing and conquering a task by allowing pairs to assign a new task to themselves or their partner. This may help reduce "free riders effect" especially in mixed-gender pairs.

VI. Conclusions

This is the first study to investigate the gender differences in pairs of CS students programming remotely. Based on our analysis, through a gender lens, of same- and mixed-gender pairs of CS students we observed the following:

- *RQ1: Coordination* – Same-gender pairs tended to be democratic, put aside their own preferences, and divided tasks fairly, while mixed-gender pairs had one authoritative partner who more often took control of the task and instructed their partner.
- *RQ2: Communication* – Women utilized more non-verbal cues than men. Both genders had a different understanding of the driver and navigator roles: (a) When they knew how to solve the task, women preferred to be the navigator while men preferred to be the driver (b) When they did not know what to do, men preferred to be navigator and women preferred to be driver. Furthermore, men were more comfortable using remote pair programming while women preferred co-located pair programming.
- *RQ3: Collaboration* – Women were more aware of gender differences, and preferred to work with other women than men, who did not have a preference. Additionally, in mixed-gender pairs, women's self-efficacy did not increase; hence women did not benefit from mixed-gender pairings.

The gender differences reported in this study can inform the design of the theory-based method GenderMag. We recommended extending the GenderMag method by informing the design of personas and cognitive walkthroughs to include gender-specific leadership styles and driver/navigator preferences. Furthermore, we recommended features for CS educational tools to bring the gender-inclusivity benefits of pair programming to remote CS education. Our study shows that online CS education still needs further development to fulfill its promise of equalizing life-circumstance barriers for women.

References

[1] Diversity in computer science.
URL https://code.org/diversity

[2] A. Belshee, Promiscuous pairing and beginner's mind: Embrace inexperience, in: Proceedings of the Agile Development Conference, IEEE Computer Society, 2005, pp. 125–131.

[3] T. Dybå, E. Arisholm, D. I. K. Sjøberg, J. E. Hannay, F. Shull, Are two heads better than one? on the effectiveness of pair programming, IEEE Softw. 24 (6) (2007) 12–15.

[4] A. Cockburn, L. Williams, Extreme programming examined, Addison-Wesley Longman Publishing Co., Inc., 2001, Ch. The Costs and Benefits of Pair Programming, pp. 223–243.

[5] L. Williams, E. Wiebe, K. Yang, M. Ferzli, C. Miller, In support of pair programming in the introductory computer science course, Computer Science Education 12 (3) (2002) 197–212.

[6] D. W. Palmieri, Knowledge management through pair programming, NY: Morrow, 2002.

[7] E. Arisholm, H. Gallis, T. Dyba, D. I. K. Sjoberg, Evaluating pair programming with respect to system complexity and programmer expertise, IEEE Transactions on Software Engineering 33 (2) (2007) 65–86.

[8] M. Celepkolu, K. E. Boyer, Thematic analysis of students' reflections on pair programming in cs1, in: Proceedings of the 49th ACM Technical Symposium on Computer Science Education, ACM, 2018, pp. 771–776.

[9] J. Chong, T. Hurlbutt, The social dynamics of pair programming, in: 29th International Conference on Software Engineering (ICSE'07), 2007, pp. 354–363.

[10] H. Gallis, E. Arisholm, T. Dyba, A transition from partner programming to pair programming- an industrial case study, in: Proceedings of Pair programming workshop in 17th annual ACM conference on Object-Oriented Programming, Systems, Languages, and Applications, 2002, pp. 1–8.

[11] K. Han, E. Lee, Y. Lee, The impact of a peer-learning agent based on pair programming in a programming course, IEEE Transactions on Education 53 (2) (2010) 318–327.

[12] D. L. Jones, S. D. Fleming, What use is a backseat driver? a qualitative investigation of pair programming, 2013, pp. 103–110.

[13] Z. Li, E. Kraemer, Social effects of pair programming mitigate impact of bounded rationality, in: Proceedings of the 45th ACM Technical Symposium on Computer Science Education, ACM, 2014, pp. 385–390.

[14] N. Nagappan, L. Williams, M. Ferzli, E. Wiebe, K. Yang, C. Miller, S. Balik, Improving the cs1 experience with pair programming, in: Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, ACM, 2003, pp. 359–362.

[15] C. O'Donnell, J. Buckley, A. Mahdi, J. Nelson, M. English, Evaluating pair-programming for non-computer science major students, in: Proceedings of the 46th ACM Technical Symposium on Computer Science Education, ACM, 2015, pp. 569–574.

[16] L. Williams, D. S. McCrickard, L. Layman, K. Hussein, Eleven guidelines for implementing pair programming in the classroom, in: Agile 2008 Conference, 2008, pp. 445–452.

[17] L. Williams, R. Kessler, Pair Programming Illuminated, Addison-Wesley Longman Publishing Co., Inc., 2002.

[18] C. M. Lewis, N. Shah, How equity and inequity can emerge in pair programming, in: Proceedings of the Eleventh Annual International Conference on International Computing Education Research, ACM, 2015, pp. 41–50.

[19] C. McDowell, L. Werner, H. Bullock, J. Fernald, The effects of pair-programming on performance in an introductory programming course, in: Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education, ACM, 2002, pp. 38–42.

[20] C. Mcdowell, L. Werner, H. E. Bullock, J. Fernald, The impact of pair programming on student performance, perception and persistence, in: 25th International Conference on Software Engineering, 2003. Proceedings., 2003, pp. 602–607.

[21] C. McDowell, L. Werner, H. E. Bullock, J. Fernald, Pair programming improves student retention, confidence, and program quality, Commun. ACM 49 (8) (2006) 90–95.

[22] K. W. Han, E. Lee, Y. Lee, The effects of pair programming on achievement and motivated strategies in programming course 9 (6) (2006) 11–28.

[23] F. J. Rodríguez, K. M. Price, K. E. Boyer, Exploring the pair programming process: Characteristics of effective collaboration, in: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, ACM, 2017, pp. 507–512.

[24] O. Ruvalcaba, L. Werner, J. Denner, Observations of pair programming: Variations in collaboration across demographic groups, in: Proceedings of the 47th ACM Technical Symposium on Computing Science Education, ACM, 2016, pp. 90–95.

[25] L. L. Werner, B. Hanks, C. McDowell, Pair-programming helps female computer science students, J. Educ. Resour. Comput. 4 (1).

[26] D. Damian, Global software development: growing opportunities, ongoing challenges, Software Process: Improvement and Practice 8 (4) (2003) 179–182.

[27] J. D. Herbsleb, D. Moitra, Global software development, 2001.

[28] J. D. Herbsleb, D. J. Paulish, M. Bass, Global software development at siemens: experience from nine projects, in: Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005., 2005, pp. 524–533.

[29] S. D. Teasley, L. A. Covi, M. S. Krishnan, J. S. Olson, Rapid software development through team collocation, IEEE Trans. Softw. Eng. 28 (7) (2002) 671–683.

[30] P. Baheti, E. F. Gehringer, P. D. Stotts, Exploring the efficacy of distributed pair programming, in: Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods - XP/Agile Universe 2002, Springer-Verlag, 2002, pp. 208–220.

[31] R. Duque, C. Bravo, Analyzing work productivity and program quality in collaborative programming, in: 2008 The Third International Conference on Software Engineering Advances, 2008, pp. 270–276.

[32] B. Hanks, Empirical evaluation of distributed pair programming, Int. J. Hum.-Comput. Stud. 66 (7) (2008) 530–544.

[33] D. Tsompanoudi, M. Satratzemi, S. Xinogalos, L. Karamitopoulos, An empirical study on factors related to distributed pair programming, International Journal of Engineering Pedagogy 9 (2) (2019) 65–81.

[34] B. J. da Silva Estácio, R. Prikladnicki, Distributed pair programming, Inf. Softw. Technol. 63 (50) (2015) 1–10.

[35] B. A. Becker, K. Quille, 50 years of cs1 at sigcse: A review of the evolution of introductory programming education research, in: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, ACM, 2019, pp. 338–344.

[36] K. M. Ying, L. G. Pezzullo, M. Ahmed, K. Crompton, J. Blanchard, K. E. Boyer, In their own words: Gender differences in student perceptions of pair programming, in: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, ACM, 2019, pp. 1053–1059.

[37] B. Zhong, Q. Wang, J. Chen, The impact of social factors on pair programming in a primary school, Comput. Hum. Behav. 64 (2016) 423–431.

[38] K. S. Choi, A comparative analysis of different gender pair combinations in pair programming, Behaviour & Information Technology 34 (8) (2015) 825–837.

[39] J. Butler, Revisiting bodies and pleasures, Theory Culture & Society - THEOR CULT SOC 16 (1999) 11–20.

[40] C. West, D. H. Zimmerman, Doing gender, Gender and Society 1 (2) (1987) 125–151.

[41] C. H. Lewis, Using the "thinking aloud" method in cognitive interface design, RC 9265, IBM (1982).

[42] C. B. Seaman, Qualitative methods in empirical studies of software engineering, in: IEEE Transactions on Software Engineering, 1999, pp. 557–572.

[43] L. Williams, R. R. Kessler, W. Cunningham, R. Jeffries, Strengthening the case for pair programming, IEEE Softw. 17 (4) (2000) 19–25.

[44] Eclipse IDE (2019).
URL https://www.eclipse.org/

[45] TeamViewer (2019).
URL https://www.teamviewer.com/en-us/

[46] Morae.
URL http://www.techsmith.com/morae.asp

[47] Study material.
URL https://github.com/RainyAtom/HHPairProgStudy

[48] P. Jaccard, Etude de la distribution florale dans une portion des alpes et du jura, Bulletin de la Societe Vaudoise des Sciences Naturelles 37 (1901) 547–579.

[49] I. Cuadrado, M. M. D. Navas, F. Molero, E. Ferrer, J. F. Morales, Gender differences in leadership styles as a function of leader and subordinates sex and type of organization, 2012.

[50] T. Yang, H. E. Aldrich, Whos the boss? explaining gender inequality in entrepreneurial teams, American Sociological Review 79 (2) (2014) 303–327.

[51] P. E. Gallaher, Individual differences in nonverbal behavior: Dimensions of style, Journal of Personality and Social Psychology 63 (1992) 133–145.

[52] V. Braun, V. Clarke, Using thematic analysis in psychology, Qualitative research in psychology 3 (2006) 77–101.

[53] C. A. Ellis, S. J. Gibbs, G. Rein, Groupware: Some issues and experiences, Commun. ACM 34 (1) (1991) 39–58.

[54] F. L. Denmark, Women, leadership, and empowerment, Psychology of Women Quarterly 17 (3) (1993) 343–356.

[55] A. Eagly, S. Karau, Gender and the emergence of leaders: A meta-analysis, Journal of Personality and Social Psychology 60 (5) (1991) 685–710.

[56] A. Eagly, S. Karau, Role congruity theory of prejudice toward female leaders, Psychological review 109 (2002) 573–98.

[57] A. Eagly, B. Johnson, Gender and leadership style: A meta-analysis, CHIP Documents 108.

[58] A. Eagly, S. Sczesny, Stereotypes about women, men, and leaders: Have times changed?, American Psychological Association, 2009.

[59] C. L. Hoyt, J. L. Burnette, Gender bias in leader evaluations: Merging implicit theories and role congruity perspectives, Personality and Social Psychology Bulletin 39 (10) (2013) 1306–1319.

[60] H. Ibarra, J. L. Petriglieri, Impossible selves : Image strategies and identity threat in professional women's career transitions by, 2007.

[61] C. Kubu, Who does she think she is? women, leadership and the b(ias) word, The Clinical Neuropsychologist 32 (2017) 1–17.

[62] K. Lanaj, J. R. Hollenbeck, Leadership over-emergence in self-managing teams: The role of gender and countervailing biases, Academy of Management Journal 58 (5) (2015) 1476–1494.

[63] L. H. Liang, D. Brown, D. Lance Ferris, S. Hanig, H. Lian, L. Keeping, The dimensions and mechanisms of mindfulness in regulating aggressive behaviors, Journal of Applied Psychology 103.

[64] D. E. Perry, N. A. Staudenmayer, L. G. Votta, People, organizations, and process improvement, IEEE Software 11 (4) (1994) 36–45.

[65] B. J. Tepper, Consequences of abusive supervision, Academy of Management Journal 43 (2) (2000) 178–190.

[66] N. L. Kerr, S. E. Bruun, Dispensability of member effort and group motivation losses: Free-rider effects, Journal of Personality and Social Psychology 44 (1983) 78–94.

[67] D. James, S. Clarke, Women, men, and interruptions: A critical review, in: D. Tannen (Ed.), Oxford studies in sociolinguistics. Gender and conversational interaction, 1998, pp. 231–280.

[68] C. W. [160] Don H Zimmerman, West, sex roles, interruptions and silences in conversation, in: Communications of the ACM, 1988, pp. 1268–1287.

[69] K. J. Anderson, C. Leaper, Meta-analyses of gender effects on conversational interruption: Who, what, when, where, and how, 1998.

[70] B. Curtis, H. Krasner, N. Iscoe, A field study of the software design process for large systems, Commun. ACM 31 (11) (1988) 1268–1287.

[71] R. E. Grinter, Recomposition: Putting it all back together again, in: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work, ACM, 1998, pp. 393–402.

[72] R. E. Kraut, L. A. Streeter, Coordination in software development, Commun. ACM 38 (3) (1995) 69–81.

[73] M. L. Knapp, J. A. Hall, Non-verbal communication in human interaction.

[74] J. A. Briton, Nancy J.and Hall, Beliefs about female and male nonverbal communication, Sex Roles 32 (1) (1995) 79–90.

[75] E. Leigh, Men & women communicating in the workplace, in: The Center for Healthcare Communications, 2010.

[76] D. Tannen, You just don't understand: women and men in conversation, 1990.

[77] J. L Lakin, T. L Chartrand, Using nonconscious behavioral mimicry to create affiliation and rapport, Psychological science 14 (2003) 334–339.

[78] M. Ubando, Gender differences in intimacy, emotional expressivity, and relationship satisfaction, Pepperdine Journal of Communication Research 4 (13).

[79] J. D. Herbsleb, A. Mockus, T. A. Finholt, R. E. Grinter, An empirical study of global software development: distance and speed, in: Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001, 2001, pp. 81–90.

[80] S. L. Jarvenpaa, K. Knoll, D. E. Leidner, Is anybody out there?: Antecedents of trust in global virtual teams, J. Manage. Inf. Syst. 14 (4) (1998) 29–64.

[81] S. L. Jarvenpaa, D. E. Leidner, Communication and trust in global virtual teams, Organization Science 10 (6) (1999) 791–815.

[82] P. Kanawattanachai, Y. Yoo, Dynamic nature of trust in virtual teams, Journal of Strategic Information Systems 11 (2002) 187–213.

[83] S. L. Jarvenpaa, T. R. Shaw, D. S. Staples, Toward contextualized theories of trust: The role of trust in global virtual teams, Information Systems Research 15 (3) (2004) 215–310.

[84] N. B. Moe, D. Šmite, Understanding a lack of trust in global software teams: A multiple-case study, Softw. Process 13 (3) (2008) 217–231.

[85] L. Beckwith, M. Burnett, S. Wiedenbeck, C. Cook, S. Sorte, M. Hastings, Effectiveness of end-user debugging software features: Are there gender issues?, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2005, pp. 869–878.

[86] L. Beckwith, C. Kissinger, M. Burnett, S. Wiedenbeck, J. Lawrance, A. Blackwell, C. Cook, Tinkering and gender in end-user programmers' debugging, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2006, pp. 231–240.

[87] M. Burnett, S. D. Fleming, S. Iqbal, G. Venolia, V. Rajaram, U. Farooq, V. Grigoreanu, M. Czerwinski, Gender differences and programming environments: Across programming populations, in: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ACM, 2010, pp. 28:1–28:10.

[88] M. M. Burnett, L. Beckwith, S. Wiedenbeck, S. D. Fleming, J. Cao, T. H. Park, V. Grigoreanu, K. Rector, Gender pluralism in problem-solving software, Interact. Comput. 23 (5) (2011) 450–460.

[89] A. Durndell, Z. Haag, Computer self efficacy, computer anxiety, attitudes towards the internet and reported experience with the internet, by gender, in an east european sample, Computers in Human Behavior 18 (2002) 521–535.

[90] K. Hartzel, How self-efficacy and gender issues affect software adoption and use, Commun. ACM 46 (9) (2003) 167–171.

[91] A. Fisher, J. Margolis, Unlocking the clubhouse: Women in computing, SIGCSE Bull. 35 (1) (2003) .23–.

[92] P. NSF ITWF & ITR/EWF Principal Investigator Conference, Philadelphia, The influence of professional identification on the retention of women and racial minorities in the it workforce, 2004, pp. 65–69.

[93] Piazza, blog. stem confidence gap (2005).
URL `http://blog. piazza.com/stem-confidence-gap/`

[94] A. Singh, V. Bhadauria, A. Jain, A. Gurung, Role of gender, self-efficacy, anxiety and testing formats in learning spreadsheets, Comput. Hum. Behav. 29 (3) (2013) 739–746.

[95] P. Dourish, V. Bellotti, Awareness and coordination in shared workspaces, in: Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work, ACM, 1992, pp. 107–114.

[96] D. Nutter, C. Boldyreff, Historical awareness support and its evaluation in collaborative software engineering, 2003, pp. 171–176.

[97] H. H. Olsson, E. Conchir, P. J. Agerfalk, B. Fitzgerald, Global software development challenges: A case study on temporal, geographical and socio-cultural distance, 2006, pp. 3–11.

[98] I. Steinmacher, C. Steinmacher, A. Paula, M. Aurelio, Awareness support in global software development: A systematic review based on the 3c collaboration model, Collaboration and Technology 6257.

[99] A. Shekhar, N. Marsden, Cognitive walkthrough of a learning management system with gendered personas, in: Proceedings of the 4th Conference on Gender & IT, ACM, 2018, pp. 191–198.

[100] C. Mendez, H. S. Padala, Z. Steine-Hanson, C. Hilderbrand, A. Horvath, C. Hill, L. Simpson, N. Patil, A. Sarma, M. Burnett, Open source barriers to entry, revisited: A sociotechnical perspective, in: Proceedings of the 40th International Conference on Software Engineering, ACM, 2018, pp. 1004–1015.

[101] G. Charness, U. Gneezy, Strong evidence for gender differences in risk taking, Journal of Economic Behavior & Organization 83 (1) (2012) 50–58.

[102] M. Burnett, A. Peters, C. Hill, N. Elarief, Finding gender-inclusiveness software issues with gendermag: A field investigation, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ACM, 2016, pp. 2586–2598.

[103] M. Burnett, S. Stumpf, J. Macbeth, S. Makri, L. Beckwith, I. Kwan, A. Peters, W. Jernigan, Gendermag: A method for evaluating software's gender inclusiveness, Interacting with Computers 28 (6) (2016) 760–787.
[104] Gendermag.
URL http://gendermag.org/
[105] D. Pinelle, C. Gutwin, S. Greenberg, Collaboration usability analysis: Task analysis for groupware usability evaluations, Interactions 11 (2) (2004) 7–8.
[106] T. Mahatody, M. Sagar, C. Kolski, State of the art on the cognitive walkthrough method, its variants and evolutions, Int. J. Hum. Comput. Interaction 26 (2010) 741–785.
[107] V. I. Grigoreanu, M. M. Burnett, G. G. Robertson, A strategy-centric approach to the design of end-user debugging tools, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2010, pp. 713–722.
[108] S. Kuttal, A. Sarma, G. Rothermel, Predator behavior in the wild web world of bugs: An information foraging theory perspective, 2013, pp. 59–66.
[109] S. K. Kuttal, A. Sarma, G. Rothermel, Debugging support for end user mashup programming, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2013, pp. 1609–1618.