31 A Case Study of Qualitative Methods

Colleen M. Lewis

1 Introduction

In research – and many parts of life – we only see the finished product, a snapshot of calm and certainty even when the reality is chaotic. When people meet me, they might learn that I am a computer science (CS) professor. I assume they would never guess that I nearly failed data structures in college and still struggled in my second attempt. They would never imagine how many interviews I bombed and graduate schools I did not get into. They don't see the inevitable paper and grant rejections or poor teaching evaluations. Those things aren't on my CV, but reflecting back, I see these as some of the most influential elements for my learning.

This chapter is a narrative of the actual doing of a research study, what Roth (2006) calls a *praxis narrative*. I hope to give you a "feel for the game" (Bourdieu, 1992) of doing one type of qualitative research. Ideally, you will gain some insights into qualitative methods, or at least recognition that if it feels chaotic, it is not necessarily wrong. Textbooks about qualitative methods have a burden of providing clarity to the methods. This chapter instead seeks to show all of the mess and ambiguity. In our current context, where computing knowledge is often perceived as only available to the intellectual elite or people with a "geek gene," it is our responsibility to challenge these notions and help others see our humanness. I will attempt to do that while telling the backstory of this paper.

In this chapter, I will share some of what I learned through writing, revising, and now reflecting on a paper that traversed a particularly rocky path. My qualitative analysis was eventually published in a paper at the Association for Computing Machinery (ACM) Special Interest Group on Computer Science Education (SIGCSE) International Computing Education Research (ICER) conference (Lewis, 2012a), but the path there was a bit bumpy. The analysis came from my master's thesis (submitted December 2009), abbreviated to submit to ICER in April of 2010. It was rejected from ICER in 2010 and again in 2011. Despite the suspicion that the manuscript was doomed, I decided to revise and resubmit it again in 2012. Only in this third submission to ICER was it accepted. I received incredibly thoughtful – and harsh – reviews of my first submission to ICER in 2010. At that time, my work was described as preliminary and that the contribution was fairly minimal.

When it was finally accepted in 2012, it was awarded the inaugural Chair's Award at the 2012 ICER conference, "given to the paper that, in the judgment of the organizing committee, best illustrates the highest standards of empirical computing education research, taking into account the quality of its questions asked, methodology, analysis, writing, and contribution to the field." As an eternal imposter, I am distrustful of this characterization, but I can confidently claim that my work got much better and that the story of this growth captures important parts of my learning about qualitative methods.

The feedback I received at each point, even when it seemed unnecessarily harsh, was thoughtful and ultimately helpful. In terms of the volume of new and edited text, each new ICER submission was a significant revision. However, the changes themselves were never burdensome. The ideas were there, I just did not know how to put them into words, or even that I *should* put them into words. The key learning was subtle and led to reframing, restructuring, and ultimately much better analysis.

The following sections will chronicle my insights within the process distilled into *tips*. The tips are sequenced to roughly align with the broad categories of data collection, analysis, and writing. Each tip tells part of the story of the work of qualitative research. In recounting a part of the story, I attempt to capture something I think I learned about these broad, often interwoven categories of doing qualitative research. I have written each tip as a declarative statement, but these statements serve only as labels for the messiness that the real work of qualitative methods involves. Given that messiness, these declarative tips might be improved by adding "when possible" to the end of each. Now that I have written these tips, they seem obvious; they might even appear in qualitative methods textbooks. Perhaps I had even read them with them in one of the two qualitative methods courses I took en route to my PhD in science and mathematics education. However, qualitative methods require juggling multiple goals, and I simply had not developed fluency and consistency in applying these ideas and strategies.

Data Collection

- Check and organize your data as you go (Section 2.1).
- Don't plan to collect data when you are developing a new curriculum (Section 2.2).
- Adapt your data collection as you go (Section 2.3).

Data Analysis

- Keep detailed notes as your analysis evolves (Section 3.1).
- Follow your instincts (Section 3.2).
- Try to cherry-pick some interesting data (Section 3.3).
- Invent terms (i.e., constructs) so that you can work toward more precise analysis (Section 3.4).
- It would be easier if it were systematic (Section 3.5).
- When your analysis is good, it will seem obvious (Section 3.6).

- Acknowledge that your epistemological assumptions will shape the analysis (Section 3.7).

Writing

- Provide structure to help your reader understand your data (Section 4.1).
- Distinguish description and interpretation (Section 4.2).
- Present enough data so that someone could disagree with your interpretation (Section 4.3).
- Explain why you chose qualitative methods (Section 4.4).
- Explain the limitations of the work (Section 4.5).
- Explain how your analysis fits into the ongoing conversation (Section 4.6).
- Tie the motivation to the analysis (Section 4.7).
- Make the implications for teaching explicit (Section 4.8).

I encourage you to read the paper. It reveals how much of the real work is invisible in the final product. However, I will assume you haven't read it and will provide the relevant context as necessary. The paper describes a student that I gave the pseudonym Kevin as he debugs a Scratch program over the course of less than five minutes. Kevin was one of fifty 11–12-year-old students enrolled in a summer enrichment program for academically advanced students. I taught the enrichment course with the help of two other adults. The students were divided between a morning and afternoon section, each with 36 hours of instruction. One class used Scratch and the other used Logo. Each day, students in each class did isomorphic tasks using their given programming language. In this design, I was also generating quantitative data from surveys and daily quizzes to see how students' attitudes and performance differed between the two sections. This quantitative data was described in another paper (Lewis, 2010). Kevin was technically pair programming at the time, but his partner, Rachel, did not say anything during the episodes I analyzed, and from the whole class video she appears to not be looking at their computer screen or Kevin. With that context, you are ready for the tips!

2 Data Collection

In preparation for my data collection, I frequently received advice that I was collecting too much data. With student assent and parental consent, I recorded students' screens during each of the 36 hours of instruction in the summer enrichment program. My students, 11–12-year-olds, worked in a total of 25 pairs, with 12 pairs in the first offering and 13 pairs in the second offering. Every hour of instruction was also recorded on three video cameras. Therefore, I had over 900 hours or over 38 days of video recordings. "Too much data" seems like a reasonable critique and an understatement. I am nearly certain that I will deserve this critique again on future projects. However, the following three tips attempt to capture what I have learned and hope to apply in data collection for future projects.

2.1 Tip – Check and Organize Your Data As You Go

I am dismissive of the "too much data" critique, but it is not wrong. As my mentors and peers predicted, I was totally overwhelmed during data collection, and as a result I did not sufficiently check and organize the data during the course. That is a mistake I hope to avoid repeating. I had done plenty in advance of the summer course, but it was still insufficient.

For a class project, I had collected and analyzed data from a one-hour pilot study with four students. This short intervention did not pilot the whole curriculum, but I began the summer having written a paper using the same type.

That summer, I had prepared by installing the screen recording software on all of the computers in my classroom and confirmed that it recorded using each of the external microphones that were required. I determined that the battery in the external microphones would need to be replaced daily to ensure audio recordings were made. I had developed a labeling system for the physical tapes recorded by each of the three video cameras in the room. I made elaborate checklists that we used to ensure we collected and stored the data. I owe a huge debt of gratitude to my co-teachers Brittany Murlas and Christa Henderson because they did not know what they had signed up for! We had very little data lost during the summer; given the complexity of the data collection, I think this was impressive.

There was just one catch. When the screen recordings expanded, the audio and the video file got out of sync. When I opened the file with video editing software, the audio component was about two-thirds of the length of the video. Only when the audio component was stretched to match the length of the video did the audio appear to sync up with the video. This easily could have been detected before the end of the summer if I had been checking the data throughout. Instead, it required that I use a computer in my research lab to stretch the audio, which took more time to process than the length of the original video. It seems likely that if I had caught this earlier I could have avoided this additional step. Will this same problem repeat itself? No, but surely something else will go wrong, and I might as well set aside the time during data collection to figure that out.

2.2 Tip – Don't Plan to Collect Data When You Are Developing a New Curriculum

The course in which I collected data was a course I had never taught before. I had prepared all of the curriculum in advance, but students completed the content I had planned for the three-week course within the first week. I spent the remaining two weeks sleep-deprived as I tried to keep up with the students and generate more activities for them. How were my results changed because the curriculum was far from perfect? Recall that I taught computing using Scratch and Logo to address quantitative questions related to the strengths and weaknesses of each. The curriculum was likely equally bad between the Scratch and Logo

offerings of the course, and I hope did not affect my results comparing Scratch and Logo (Lewis, 2010).

From that experience, I now strongly believe that you should not collect data in the first offering of a course. This seems like another tip that I will ignore sooner than is wise, but the flaws in a first offering of a course most likely will reduce the interpretability of results. Perhaps only collecting data with polished curricula is unrealistic, but describing the weaknesses of the curriculum is necessary. If I could go back and revise that publication (Lewis, 2010), I would include more information about the ways in which the class was rocky. I think I owe that to my readers and the research community generally.

2.3 Tip – Adapt Your Data Collection As You Go

One of my reviewers in 2010 argued that I should standardize my data collection and appeared to imply that qualitative work requires that all participants be given the same set of problems. While I reject the larger argument that the participants need a consistent experience, there are definitely practical considerations that would justify this. First, qualitative research requires a lot of space to explain the methods and analysis. If participants are each answering different questions, a lot of real estate could be lost to the description of the relevant problems that appear in the analyses. Second, asking the same question multiple times may provide the opportunity to follow up on hypotheses that are developed from an earlier participant's answer. This was helpful in my dissertation (Lewis, 2012b), which involved one-on-one interviews with students solving programming problems during a clinical interview. Often a student's answers puzzled me, and their responses to my follow-up questions provided no illumination. If another student expressed a similar idea, follow-up questions with that second student might be more fruitful. This opportunity was only available because I asked all students a common set of questions. However, these follow-up questions are examples where each student may be asked different questions. These unplanned follow-up questions allowed me to explore topics I did not know to plan interview questions for.

3 Data Analysis

I have frequently received the advice to be more systematic in my analysis. I think an assumption in the word "systematic" is that there are prescriptive steps that should be followed to ensure that qualitative methods are done correctly. That has not been my experience. I will concede in the first tip below that I should be more systematic in keeping detailed notes of my analysis decisions. However, the remaining tips embrace the openness of qualitative methods. Qualitative methods do not guarantee that another researcher would find the same result. My paper focused on less than five minutes of Kevin's interaction with Scratch. With over 900 hours of video recordings, there is seemingly

no chance that another researcher would have identified this same five-minute excerpt! The first analysis task I had was to narrow the focus of the research and data analysis. This task feels anything but systematic, and I have gotten more and more comfortable with that. Section 4 describes the ways I try to build trust with my reader because I am not relying on following a predetermined set of steps.

3.1 Tip – Keep Detailed Notes As Your Analysis Evolves

One way to be systematic is to document the analysis process as you go. In qualitative methods, this documentation is typically referred to as "memos." I think about it as a lab notebook or journal describing the analysis. I have good systems for implementing this on my collaborative projects, but not my individual ones.

In all of my collaborative projects, we coedit a shared Google doc during each meeting, and those notes nearly form a transcript of the meeting. In one project that spanned multiple years (Lewis, Anderson, & Yasuhara, 2016; Lewis, Yasuhara, & Anderson, 2011), we made it easier to refer back to these meeting notes by labeling each document with a title and including a one-sentence summary of the discussion at the top of the document. This was invaluable when we later wrote up our methods, and it helped us during the analysis to avoid inadvertently backtracking.

A weakness of the methods in my individual analysis efforts is that I do not have comprehensive notes about the path of my analysis. When I watch video data, I take notes in what is often called a "content log." This helps me document my observations and initial hypotheses, but it does not capture the continual snapshots of my analysis process like I have done in my collaborative projects. In my future individual analyses, I need to learn to incorporate the generation of this type of artifact. For this aspect of being systematic, I see only advantages.

3.2 Tip – Follow Your Instincts

To sift through 900 hours of video, I began my analysis focused on a particular piece of the curriculum: where students draw a brick wall, as shown in Figure 31.1. For those not familiar with Scratch, characters, like the cat shown in Figure 31.1, can draw lines as they move around the screen. The task of drawing the brick wall, which was designed by Guy Haas (n.d.), involves a relatively complex set of movements of the character that involve alternating whether the character is drawing a line (i.e., to draw a brick) or is not drawing a line (i.e., to create a space). It is pretty tricky to pick lengths for the bricks, partial bricks, and spaces that allow each row of bricks to line up. Additionally, it is tricky to navigate the character between these lines. If you are dying to see the code, you might try it directly in Scratch or read the published paper (Lewis, 2012a).

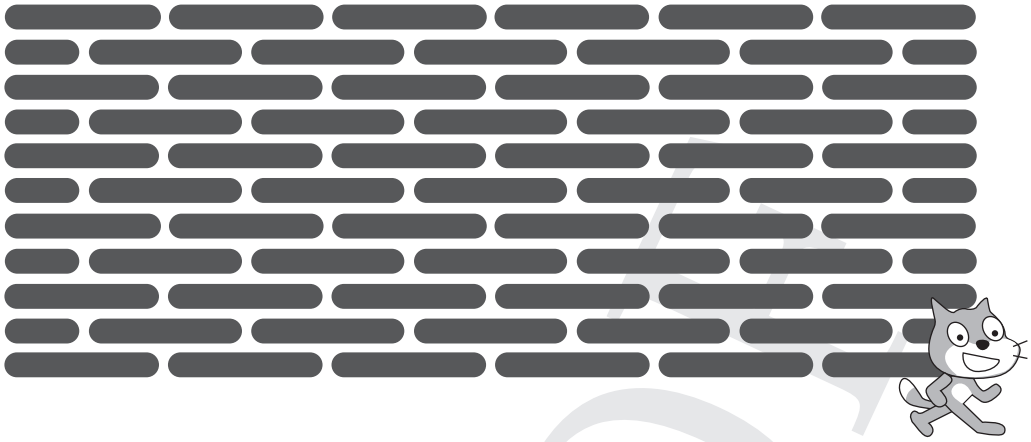

Figure 31.1 *Completed drawing of a brick wall in Scratch.*

I started with the brick wall activity based upon my experience of students completing the activity in class. Students often appeared to productively engage with the activity for between one and two hours.[1] Students hit predictable bugs, which appeared to be conceptually rich. Best of all, these bugs were immediately recognizable to me from what was drawn on the screen. For example, students often hit a point where the rows were skewed like a staircase. From having taught the class in which data were collected, my gut instinct was to focus on this problem in order to observe debugging behavior.

Perhaps my description of why I picked the brick wall sounds systematic. I certainly had reasons to pick the brick wall task. However, these reasons evolved as I started the analysis, and I could have picked another task and generated compelling reasons for focusing on that task instead. Narrowing the focus of the analysis is fundamentally open, and I have begun to come to grips with the fact that my plan is to trust my instincts. Luckily, as a CS instructor, I am constantly able to refine and test my instincts about how people learn CS, which increases my confidence in my instincts. However, focusing on one thing means there is less time to focus on other things, and there is nothing other than my fallible instincts on which to rely.

3.3 Tip – Try to Cherry-Pick Some Interesting Data

Once I had picked the task and I had started watching videos, one might describe my process of picking specific video excerpts as cherry-picking. While cherry-picking is used as a critique, I now think about this work as choosing the *right* cherries to pick! In other projects, I have cherry-picked quotes to illustrate a point in the completed analysis (e.g., Lewis, Yasuhara, & Anderson, 2011), but in this project, cherry-picking was an early step in the analysis.

1 This difficult a task might have been risky in a classroom with only a single teacher, but we had three adult instructors with class sizes of 24 and 26.

As I recall, I had watched hours of video before finding Kevin's excerpt. I was looking for times where I could reasonably claim that the students were debugging. There were plenty of things that might be debugging. The video showed the students' screens, which meant in watching the videos I could observe a bug in their code and then could observe them making changes to their code. However, it was not clear that the students and I were seeing the same thing. Had they noticed the bug? Did they think that the changes they made to the code would fix the bug? This was a pretty deep rabbit hole.

I eventually stumbled upon the video of Kevin and Rachel. Kevin and Rachel's statements were unique in that they made their debugging explicit. Sometimes this was from Kevin and Rachel's dialogue and sometimes this was when Kevin seemed to be talking to himself or perhaps to the computer. The episode in the ICER paper began with Kevin stating, "What? Wait." It seemed clear that Kevin had recognized a bug.

Stumbling upon these data was like finding a gold mine! The ambiguity in every other video I had watched was disambiguated here by their statements. In the explicitness of their debugging, they were unlike other pairs. However, they did not seem to be outliers in other ways. In focusing on a single pair, I was not going to be sure of the exact extent to which the patterns I observed would generalize. Finding examples where I could observe the things I was interested in for my research (i.e., finding some cherries to pick) was the most important thing at this stage.

3.4 Tip – Invent Terms (i.e., Constructs) So That You Can Work toward More Precise Analysis

My analysis revolves around Kevin's understanding of program state. This idea of "state" is my primary *construct*. Methods books may provide a more comprehensive definition of construct, but I think of it is as a term that I use in my research for which I provide a specific definition. Across my projects, I have developed constructs that help me think about the analysis. Often I will develop constructs that do not appear in the final paper, but the process of iteratively refining my definition for the construct helps me refine my ideas and analysis.

Unfortunately, my construct of state is imperfect. The following definition of state from my paper has at least two *big* problems:

> State represents the idea, present in all programming environments, of a set of temporary or permanent variables that completely describe the current environment on which a program can act. This includes programmer-defined variables as well as other aspects of the runtime environment such as the current stack frame. Program commands change aspects of the computer program's state and the process of writing programs involves developing sequences of state change operations to achieve a particular goal.
>
> (Lewis, 2012a, p. 127)

First, my definition is inconsistent with a way that computer scientists talk about state. Specifically, computer scientists talk about functional programming

languages as being stateless. This relies on the idea that a function, from the mathematical definition, has a single output for each input. That is, if you pass a function the same input multiple times, you are guaranteed to get the same output. They describe functions as "stateless" because the output only depends on the input. This contrasts with imperative programming where a function that adds two to the variable x will each time result in a different value of x; that is, it is stateful.

Second, based on my definition, absolutely everything in programming is state – everything. What is not state? Nothing. This is not good. Hopefully, in the future, I will be able to narrow in on something well-defined and more specific. With a construct too broadly defined, it seems less likely to be helpful.

The first big problem with my construct tends to make some computer scientists mad, but I think the "everything is state" problem is much worse. A similar critique could likely be made of the notional machine construct (see Chapter 1), which encompasses all of how programs are executed by computers. This is not a new revelation that my construct of state has big problems. I have spent a lot of time thinking about it, and I have decided that the construct is still useful. In particular, I knew that my use of state was inconsistent with another usage, but – for people without that context – "state" seemed intuitive to them, and for people with that context, it was never difficult to clarify my meaning.

3.5 Tip – It Would Be Easier if It Were Systematic

I think a jigsaw puzzle analogy is helpful for qualitative research. There appear to be endless puzzle pieces to choose from and I am trying to pick out and connect enough pieces to provide a clear, even if small, picture. Unfortunately, there is no box lid to guide the process and no edges of the puzzle to help me pick a place to start. There are just puzzle pieces, and I have to continually convince myself that I will be able to put some of them together if I spend time considering the possible connections.

In my experience, qualitative analysis takes a long time before it seems like the pieces fit together. Along the way, I will often figure out that the pieces do not fit in the way I had believed or that a piece I had barely noticed is probably more important than some of the other connections I had made. Sometimes hours, weeks, and months can go by with seemingly no measurable progress. Qualitative analysis takes time, and it does not follow a recipe of steps where it is possible to make continual forward progress.

Each of my qualitative projects has involved multiple and sometimes prolonged periods of doubt about the direction and content of the analysis. Without a set of steps to follow, this openness can be overwhelming. Two things have been helpful for me in managing this doubt and uncertainty. First, I track my time and I give myself a sticker for every 45 minutes of work.[2] There might

2 I am not kidding: see https://medium.com/@colleenlewis/intrinsic-motivation-is-overrated-dc1cbd4a7b7c

not be measurable progress, but there are no shortcuts for this work, and I need to get myself to put in the time. Second, I frequently share my analysis with colleagues. In graduate school, we had faculty-run research groups where one or two students would sign up to present their work and lean on the group to help them chart their next steps. As a faculty member, I no longer had access to these groups, and in 2014, I started hosting the Work in Progress workshop at ICER in order to provide a similar venue of support. Many times, new perspectives can help us to find a path forward.

3.6 Tip – When Your Analysis Is Good, It Will Seem Obvious

My greatest frustration in doing qualitative research is that when I think I'm done, it all seems obvious. Why had that taken me so long? I have put some puzzle pieces together and now there is a clear picture. I think that this feeling of obviousness is a feature of well-done qualitative work, but I think this leads to me (and others) underestimating it. Consider the key finding in my paper:

> I hypothesize that a key competence in debugging is learning to identify what elements of program state are important to pay attention to and that this attention, and not only domain knowledge, mediates the debugging process.
> (Lewis, 2012a, p. 127)

Of course it is important for students to pay attention to the right elements of program state. However, even if the finding seems obvious, I think there is something important here. In the paper, I argue that once students pay attention to the right element of state, the bug feels like a simple mistake. By eventually dismissing many – or potentially most – bugs as simple mistakes, students may miss that part of learning to program is learning what elements of state they should pay attention to. The connection between qualitative results being obvious and students discounting bugs as obvious is just a coincidence, but one I appreciate.

I might not convince readers of this chapter that my key finding in my paper is meaningful, but I hope to share my experience of frustration when the final result seems obvious because the pieces have come together so well. While I find this deeply frustrating, I now see obviousness as the goal for my analysis and perhaps as something I can help my readers value.

3.7 Tip – Acknowledge That Your Epistemological Assumptions Will Shape the Analysis

In the end, my analysis connected directly to the theoretical work of my advisor, Andrea (Andy) diSessa. Is that a coincidence? No. My advisor works in the general area of epistemology. He focuses on physics, but his theories have been applied to other domains. By the time I started working on this analysis, I had – consistent with my advisor's work – started to think of learning as a process of developing the ability to solve problems across contexts. In Andy's work, he talks about how students learn to pay attention to the right things in order

to solve problems in physics (e.g., diSessa & Sherin, 1998). In my dissertation, I argued that students' thinking and learning about state are consistent with his model, which was developed around students' thinking and learning about force in physics. Building upon his work, in my dissertation, I proposed changes to his theory and argued why CS content motivated these changes, but these changes were also applicable to physics.

Connections to existing theories can be important for moving the field forward. These epistemological connections were helpful in connecting my work to the broader body of literature about education or, more specifically, "conceptual change."

While these connections to my advisor's work can be seen as biases, I think they are, to some extent, unavoidable. In educational research, our epistemological assumptions are necessarily going to shape how we make sense of students' thinking and learning. Like our instincts, these shape how we navigate the openness of qualitative research. Again, qualitative methods don't guarantee us that different readers would find the same thing. The point is to focus on phenomena where prescriptive steps would be insufficient, and we can attempt to make these assumptions that guide our work explicit.

4 Writing

I have frequently received the advice to "write clearly." Probably nothing could be less helpful. I certainly have had mixed results in my writing. Even from my accepted 2012 submission, I received a disparaging comment about my writing and a suggestion to do a "careful revision." I had certainly engaged in "careful revision," but that does not mean that my careful revision is sufficient to produce clear writing. Instead, I have settled on the following tips, which I hope steer me toward writing clearly.

4.1 Tip – Provide Structure to Help Your Reader Understand Your Data

After years of working with this data set, I found the lines of transcript, and Kevin's accompanying actions, crystal clear. This was not the case for my formal and informal readers. In early drafts, I was surprised by the things that my readers found opaque, and their feedback forced me to be explicit and structured in my writing.

I believe that the most consequential change that I made across the revisions to improve clarity was breaking the analysis up into short, sequential excerpts that each included subsections titled "Summary," "Data," and "Analysis." However, this structure was present in my 2011 submission, which notably was still rejected. Even if this structure did not unlock the elusive acceptance, it was foundational in how I now think about describing my analyses.

Each excerpt began with a "Summary," which provided a brief overview of what happened in the excerpt to follow. I tried to write this in a way that no one

would dispute and that would prepare my readers to read the transcript that followed. For example, beginning in 2011 and present in the published version, I wrote, "In excerpt three, Kevin executes the program to draw the first two rows and retraces the first and second row before accidentally tracing over the first row with a copy of the second row." I think it would be reasonable to quibble with my use of "accidentally," which I think implies an interpretation of Kevin's actions. However, the rest seems a cut-and-dried description.

The next subsection, "Data," provided the transcript for that portion and would likely be nearly impossible for a reader to read without having first read the summary. Parentheticals within the transcript described Kevin's actions of executing or modifying code. I used sequential line numbers between excerpts to show that the content progressed sequentially. I think that helping the reader to understand the overall chronology is important, and therefore I never use ellipses to indicate pauses because they can be misinterpreted as removed text. Instead, I note pauses as a parenthetical note, as in "(pause)." Realistically, using ellipses is probably fine provided that the convention is stated.

The third section, "Analysis," walked through the episode chronologically, alternating description and interpretation. The structure of these subsections is likely less important for the reader than in drawing my attention, as I am writing and revising, to the fact that I am engaged in interpretation. This might seem rather simplistic, but analysis tasks are difficult enough that this metacognition is still difficult and relevant for me. The next tip is dedicated to this difficulty.

4.2 Tip – Distinguish Description and Interpretation

In this work, I want to understand Kevin's thinking. Unfortunately, I can only make inferences or assumptions about his thinking based upon his statements and actions. Therefore, whenever I make a statement about Kevin's thinking, I need to include a hedging term such as "appears" or "I hypothesize."

For example, the first subsection of analysis began: "Kevin retraced the top row three times and appeared unsure why a third line was not drawn." The first half of the sentence constitutes my description of the events and the second half constitutes an interpretation. Another example appears in the second episode's analysis: "From Kevin's statement 'why isn't this working?' (line 10) I assume that he identified a problem, but had not identified the cause" (Lewis, 2012a, p. 131). The content of my interpretation and its source are explicit.

As an author, I might describe my task as "describing what I think happened." However, this includes relatively distinct elements such as: (1) the events that are part of the overall sequence of events and context; (2) the specific events that shape my opinion; (3) my interpretation based upon those events; and (4) any technical content knowledge a reader might not have. Again, I find this challenging. Exploring the following two sentences in the final paper can capture some of this challenge:

> By experimenting with the rotation of the character, he appeared to be appropriately attending to the direction of the character. However, for a complete

> understanding of the bug he also needed to attend to the position of the character.
>
> (Lewis, 2012a, p. 131)

The first phrase presents my description: "By experimenting with the rotation of the character." Unfortunately, even in this simple phrase, the line between description and interpretation is blurry. Specifically, my use of "experimenting with" is not purely descriptive. Instead, "experimenting with" likely evokes a goal-driven action, and I do not provide support for this interpretation. This description could likely be improved by removing "experimenting with" and replacing it with "modifying," which better fits the goal of description rather than interpretation.

Moving on from this imperfection, the second phrase presents my interpretation: "he appeared to be appropriately attending to the direction of the character." With the hedging phrase "appeared to," I signal the claim as interpretive. The third phrase presents technical content: "However, for a complete understanding of the bug he also needed to attend to the position of the character." Here, I attempt to treat the technical details of the bug objectively, without hedging, to clearly communicate an aspect of the episode that relies on content knowledge that a reader might not have. Again, there is a lot to think about in writing and analysis.

4.3 Tip – Present Enough Data That Someone Could Disagree with Your Interpretation

In each submission, my reviewers have provided alternative interpretations of the data. Their sometimes forceful presentation of alternative interpretations made me question my work and my interpretations. While initially frustrating, these were sometimes some of the most helpful comments that I received because they helped me identify specific strengths and weaknesses of my analysis.

First and foremost, providing enough data that someone could form an alternative hypothesis is helpful for ensuring that you can get feedback that will help improve the work. This can be challenging because of page limits, but earlier stages of formative feedback can provide the opportunity to expose more of the data to alternative hypotheses. Additionally, I could expose to them how I had chosen to select particular data (see Section 3.2). Before my first ICER submission, I had received multiple rounds of feedback, most notably from peers and advisors at the University of California, Berkeley. Second, transparency in the analysis is important because other researchers might not have focused on the same thing if they analyzed your data. I figure it cannot be stated too many times: qualitative methods do not provide a set of prescriptive steps that will produce a particular analysis. As discussed above, the knowledge and perspectives of the researcher are naturally embedded within qualitative analysis. By providing enough data that someone could disagree with you, your reader has the opportunity to critique your interpretations.

Developing alternative interpretations (or "rival hypotheses"; Yin, 2013) has become an essential part of my process of analysis, and I believe it is helpful if

this can be made visible to my readers. In a more recent ICER paper (Lewis & Shah, 2015), my collaborator and I compiled a list of alternative hypotheses and discussed the strengths and weaknesses of each in the paper.

4.4 Tip – Explain Why You Chose Qualitative Methods

In the process of writing this chapter, I can now see how comments that I had dismissed from my 2010 reviewers may have actually been a symptom of not explaining why I chose qualitative methods. Across my reviews, I think that reviewers' comments requesting more standard data collection and more data analysis suggest that I had not done the work of explaining why qualitative data collection and analysis were appropriate. My 2010 submission did little more than describe the work as "a fine grain analysis." The "Data Collection" section in my 2011 submission said:

> The data collection was designed to capture students engaged in programming and debugging … Rather than gathering information such as performance on quizzes, we attempted to capture students' process of solving problems.

This specified that qualitative data were collected, but not why. The text removed between the two sentences above further described the content of that data. However, my study design had reasons for both qualitative data collection and qualitative analysis. Only in the final draft, after the 2012 submission, did I explain the reason for my use of qualitative data and analysis within the methods section:

> There are a number of challenges in studying students' debugging behavior. Observing students debugging their own buggy code does not provide any consistency across research participants because the bugs they identify and fix will be unique. However, observing students debugging uniform bugs in code they did not write may be an unfamiliar experience for students and not representative of their behavior debugging their own code. The methods used in this study prioritized observing natural debugging behavior rather than documenting behavior that could easily be compared across research participants.
>
> (Lewis, 2012a, p. 128)

Here, I situate my goal of "observing natural debugging behavior" as central to the research question of understanding students' debugging behavior. I try to make the point that no other methods would be sensible, and I remind the reader of these points in the conclusion.

4.5 Tip – Explain the Limitations of the Work

In the CS education community, much of the research is quantitative. At times, it seems that reviewers are critiquing my work for not being quantitative or are implicitly critiquing the fact that I am not making quantitative claims. I am sympathetic to slipping into the dominant, quantitative frame. In the CS education research course that I teach, some students initially propose a quantitative research question for their qualitative research project. Quantitative studies are

so common that warning students about this pitfall does not seem to particularly help. Instead, I have found that the prompt, "Would this question be better answered with thousands of research participants?" can help students evaluate their research question once it is written. A quantitative research question might motivate the study, but the selection of qualitative methods implies that we are not yet able to conduct a quantitative study about the topic.

I believe that in addition to motivating the use of qualitative methods, it is necessary to specify what the analysis does not claim. It is not sufficient merely not to include unsupported quantitative claims. It is necessary to draw the reader's attention to this. For example, in the final version, I included an explanation of these bounds: "The case study is intended to illustrate details of this model, but is not intended to establish the prevalence of this pattern" (Lewis, 2012a, p. 127). Additionally, before presenting the body of the analysis, I reminded the reader about what the analysis does not show: "Case studies like this one are not intended to prove that a particular pattern of behavior exists within a population. Instead, the data serve to inform and exemplify hypotheses regarding features of learning within a domain" (Lewis, 2012a, p. 129). I think these reminders are helpful even if, in principle, they should not be required.

4.6 Tip – Explain How Your Analysis Fits into the Ongoing Conversation

I have had a difficult time situating the work within the existing literature. Reviewer 2 in 2010 critiqued my submission for not referencing relevant papers and particularly pressed me to reference relevant papers that had appeared at ICER. My initial response to this comment was righteous indignation; perhaps the reviewer was upset that I had not referenced one of their publications. After years, I have a better understanding of this comment, and I feel a bit of sheepishness for my initial response. In submitting my paper to ICER, I was attempting to join an ongoing conversation. However, by omitting relevant work published at ICER, I appeared ill informed about the current state of the conversation. This was not a self-centered reviewer – this was a reviewer helping me to understand the norms of academic research, and I now appreciate it if reviewers point me to relevant work even if I had made the conscious decision to exclude it.

Beyond including relevant references, the paper's "Previous Research" section should highlight the gap in the extant literature that I am attempting to address and should summarize the extant literature in order to contextualize that gap and bring my reader into the conversation. I have had trouble applying this idea when I think I have a novel insight about the literature or when the literature that inspired me to conduct the study does not actually relate to the contribution I am trying to make in the paper. For example, in my 2010 submission, I thought I was particularly clever for sharing how state was central across different bodies of research:

> This section reviews a diverse set of work that helps emphasize the importance of state. The first category focuses on a reframing of prior misconceptions research with an emphasis on state. The second category focuses on researchers who either explicitly or implicitly argue for the

> importance of state. The third category focuses on programming environments that make state visible for novice programmers. (Lewis, 2010)

However, it probably is not that difficult to convince a reader that state is interesting. And these connections did not relate to my analysis beyond the fact that they were about state. In later versions, I cut these connections entirely. If I had kept them, they would probably be best placed in a discussion section. They could possibly fit into a previous research section if I did not repeat the mistake of omitting how the previous research actually motivates the current analysis.

Similarly, in my master's thesis, I foolishly included literature that motivated me in pursuing this line of work. Unrelated to the analysis or results, the abstract of my master's thesis began, "The paper presents a critique of a line of computer science education research that focuses on identifying predictors of programming aptitude and assumes a static view of intelligence." My master's thesis included basically all of the content from the paper I have been describing, but also included a diatribe about arguments claiming that some students could never learn CS. This frame from my master's thesis still provides the underlying motivation for my work, but is utterly disconnected from the analytic work and research questions that I address. It seems silly now, but this previous research was connected for me, and it seemed intuitive that it should be included. I assume that I will continue to make this mistake and hope that I can remember to remove it in later drafts.

4.7 Tip – Tie the Motivation to the Analysis

Similar to my instinct to include irrelevant previous literature, I have had difficulty describing the gap in the literature that I am attempting to address. Since qualitative methods do not progress along a prescriptive path, it is not possible to target a narrow gap in the research. Instead, the contribution of the work is emergent. The work forms a picture with many possible connections to the existing literature. Now I see the need to more narrowly motivate the work. Excerpts from the abstracts of each of my ICER submissions show how I was attempting to motivate or otherwise frame the work.

I began with an absurdly huge framing, and in each submission refined the focus. The abstract for my first submission to ICER started to narrow in on the construct of state: "State is a technical computer science term for the current environment of a computer program. I argue that the concept of state is as difficult to learn and central to programming as the concept of force within physics education." This is unrelated to the analysis, and the second sentence ends with a bold and unsupported claim. What support do I have for it? None. It is a manifestation of the connections I was starting to make between the coordination class of force and my developing ideas of state as a coordination class. However, that theoretical connection did not come close to providing support for the bold claim I made. The absurdity of this unsupported claim did not

appear to be lost on my reviewers. I only mentioned the argument about force in the abstract, which I do not recommend, but because of the absurdity of this claim, I probably mitigated some of the damage.

The abstract for my second submission to ICER continued to narrow my focus, and I wisely opted to not lead with an unsupported claim: "To better understand the challenges in learning to program, we conducted a qualitative analysis of young students engaged in debugging computer programs they had written in Scratch." This goal of "understand[ing] the challenges in learning to program" is certainly central to the work. However, this is still far removed from the analytic work of the paper. This likely motivates the majority of CS education research, but is not specific to my project or work.

The abstract for my third submission to ICER further narrowed in on debugging: "To develop a model of students' debugging processes, we conducted a qualitative analysis of young students engaged in debugging computer programs they had written in Scratch." This is almost identical to the previous submission, except that my goal "To better understand the challenges in learning to program" became "To develop a model of students' debugging processes." My analysis is about understanding Kevin's debugging behavior. That is certainly a component of his learning to program, but the narrower focus seems to connect directly to the analysis, and yet is still broad enough to be perceived as generally important. In the future, I will probably only refine my understanding of my primary contribution at the end, but hope to remember to move it to the beginning of the paper.

4.8 Tip – Make the Implications for Teaching Explicit

I think that a strength and a weakness of the CS education research community is that many of us are also computing educators. This is a strength because computing educators can readily evaluate claims based upon their own experience, and my reviewers frequently noted that the importance of state was clear to them from their teaching. This is a weakness because, as computing educators, we need to make decisions about what and how to teach, and we might devalue the incremental work of education research. Education research is frequently incremental and may sometimes require years or decades to mature to the point of having clear educational implications. For those of us who are also CS educators, this may be exceptionally frustrating, but it is likely important to accept this in order to make progress in the field.

While I value the incremental work of computing education, in response to the audience of computing educators, of which I am a member, I try to explicitly connect to educators' intuitions and provide concrete recommendations. Without providing recommendations, readers might believe that the research simply documents a deficiency of students. I believe that documenting students' difficulties is productive for improving pedagogy. However, I believe that the responsibility for change rests with the educators and educational institutions. It is far too tempting to write students off as "bad students" without finding

ways that we can change the structures in small and large ways to support their success. The recommendations from my research are essentially always tentative, but by making my tentative recommendations explicit, I can hopefully prevent some misunderstandings that we can let ourselves off the hook by blaming our students. As an educator, I am frequently making small tweaks to my teaching, and if they are really just small tweaks, the risk of doing so before there is research support is probably low.

5 Conclusion

Your mileage with these tips may vary. Some of them are likely very general and would be applicable for qualitative, quantitative, and mixed-methods studies. It seems like "check and organize your data as you go" might be universally necessary. However, these tips came from particular challenges that I have faced in using and writing about qualitative methods. I hope that this will motivate me to return to these tips as a checklist throughout my later projects. I imagine that I might be the person who benefits most from the existence of this chapter, but I hope that it provides some benefit for others.

Beyond the content of the tips, I hope that this chapter helps you see how much is hidden in the unpublished backstories of research papers. Because these stories are unpublished, many people – particularly while pursuing a PhD – come to falsely believe that they are alone in wrestling with the messiness of research and the disappointment of rejection.

Acknowledgments

This chapter was originally a joint venture between myself and Josh Tenenberg. Josh provided the encouragement and coaching to help me recognize and articulate these tips. I honestly couldn't have done it without that support! Thanks, Josh! In addition to the helpful comments of the editors, I appreciate that Michael Clancy, Michael Loui, Mark Guzdial, Sebastian Dziallas, Shriram Krishnamurthi, Kathi Fisler, and Andy Ko provided helpful comments and feedback!

References

Bourdieu, P. (1992). The practice of reflexive sociology (The Paris Workshop). In P. Bourdieu & L. J. D. Wacquant (Eds.), *An Invitation to Reflexive Sociology* (pp. 216–260). Chicago, IL: Chicago University Press.

diSessa, A. A., & Sherin, B. L. (1998). What changes in conceptual change? *International Journal of Science Education*, 20(10), 1155–1191.

Haas, G. (n.d.) BFOIT Introduction to Computer Programming. Retrieved from http://guyhaas.com/bfoit/itp/Iteration.html#more_projects

Lewis, C. M. (2010). How programming environment shapes perception, learning and goals: Logo vs. Scratch. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 346–350). New York: ACM Press.

Lewis, C. M. (2012a). The importance of students' attention to program state: A case study of debugging behavior. In *Proceedings of the International Computing Education Research Conference (ICER 2012)* (pp. 127–134). New York: ACM Press.

Lewis, C. M. (2012b). *Applications of Out-of-Domain Knowledge in Students' Reasoning about Computer Program State* (PhD dissertation). University of California, Berkeley.

Lewis, C. M., Anderson, R. E., & Yasuhara, K. (2016). I don't code all day: Fitting in computer science when the stereotypes don't fit. In *Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16)* (pp. 23–32). New York: ACM Press.

Lewis, C. M., & Shah, N. (2015). How equity and inequity can emerge in pair programming. In *Proceedings of the International Computing Education Research Conference (ICER 2015)* (pp. 41–50). New York: ACM Press.

Lewis, C. M., Yasuhara, K., & Anderson, R. E. (2011). Deciding to major in computer science: A grounded theory of students' self-assessment of ability. In *Proceedings of the Seventh International Workshop on Computing Education Research* (pp. 3–10). New York: ACM Press.

Roth, W.-M. (2006). Textbooks on qualitative research and method/methodology: Toward a praxis of method. *Forum: Qualitative Social Research*, 7(1), 11.

Yin, R. K. (2013). *Case Study Research: Design and Methods*, 5th edn. Thousand Oaks, CA: Sage Publications.