ASEE
125 YEARS AT THE HEART OF ENGINEERING EDUCATION
2018 ASEE ANNUAL CONFERENCE & EXPOSITION
JUNE 24–27, 2018
SALT PALACE CONVENTION CENTER
SALT LAKE CITY, UT

Paper ID #23827

Work in Progress: A Case Study Exploring Teaching Strategies Employed in a Cornerstone Engineering Design Course

Mr. Kevin Calabro, University of Maryland, College Park

Kevin Calabro is Keystone Instructor and Associate Director in the Clark School of Engineering at the University of Maryland.

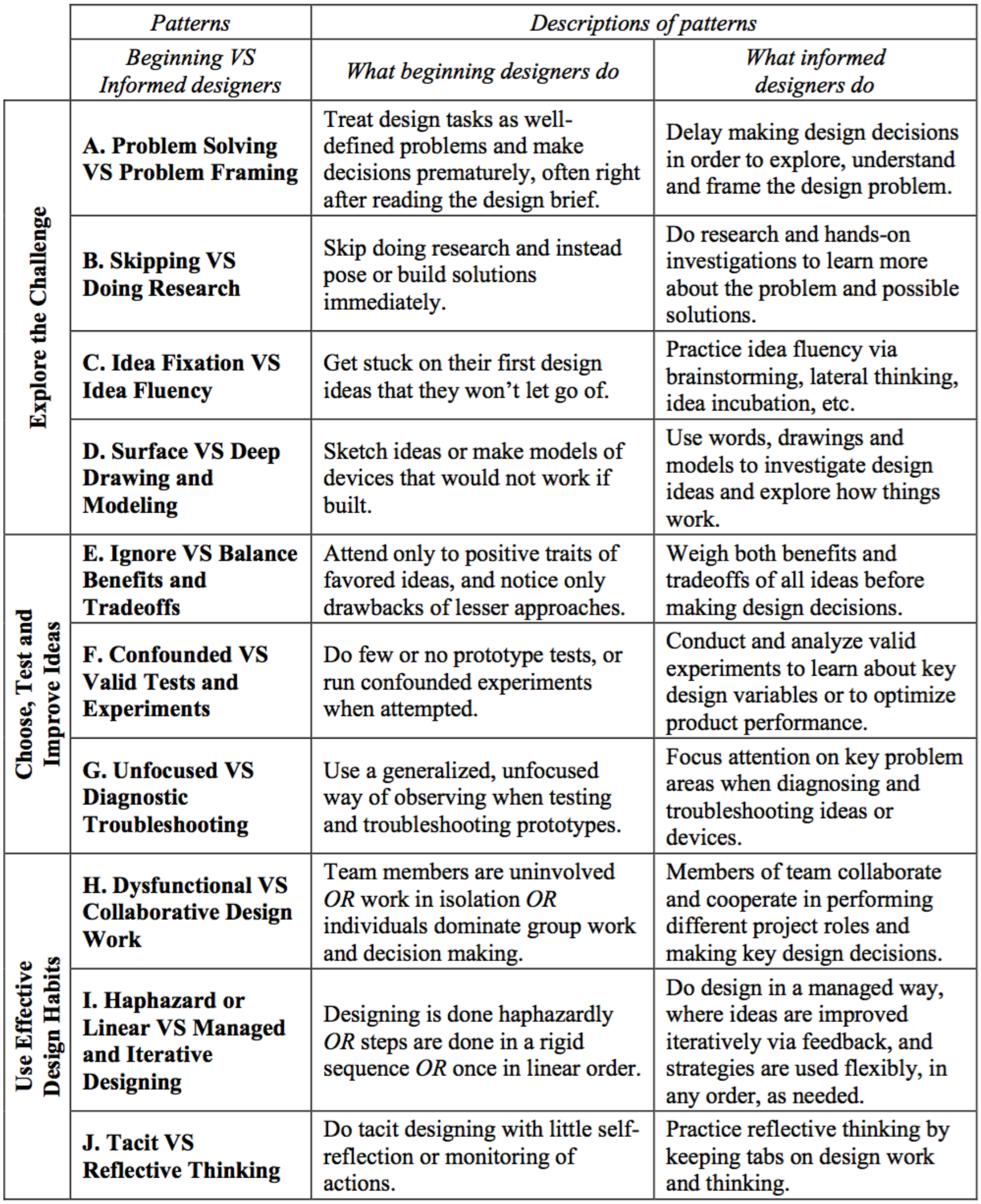
WIP: A Case Study Exploring Teaching Strategies Employed in a Cornerstone Engineering Design Course

This work in progress (WIP) paper presents findings from the early stages of a case study that explores the teaching practices of experienced educators in well-established undergraduate engineering design courses. Our research seeks to answer questions about how instruction is adjusted throughout the duration of an engineering design course to help students sequence an array of engineering design activities into coherent engineering design practices. Our research adopts the informed design teaching and learning matrix as a lens for viewing design activities [1], [2]. This WIP paper is limited to exploring teaching strategies targeting a single focal engineering design pattern – troubleshooting – within a cornerstone engineering design course. Using interview data and classroom observations, we seek to provide rich descriptions of how teaching strategies targeting troubleshooting are deployed. We compare our findings to the framework provided by Crismond and Adams [1] and offer refinements along the troubleshooting pattern. Our longer term expectation is that findings from our research, based on a case study approach, may inform engineering design instruction, curriculum development, and professional development efforts. By closely examining one practice, this paper demonstrates how this may be possible.

Literature Review

Crismond [2] and Crismond and Adams [1] present descriptions of engineering design activities and teaching strategies that we draw upon in our study of engineering design teaching practices. Crismond and Adams' [1] informed design teaching and learning matrix, reproduced below as Table 1, provides a learning progression that contrasts beginner design approaches with more informed design approaches (columns) along nine patterns of activity (rows). Each pattern describes a set of core activities associated with engineering design. The patterns are organized into three focal areas: Exploring the Challenge, Choosing/Testing/Improving Ideas, and Using Effective Design Habits. Using effective design habits such as iteration (Pattern I) or metacognition (Pattern J) are required in conjunction with other design patterns such as problem framing (Pattern A), idea fluency (Pattern C), and/or balancing benefits and tradeoffs (Pattern E). As such, each pattern should not be thought of as an engineering design activity in isolation from the others. The goal of engineering design instruction, then, is for the instructor to help students to effectively enact individual, combinations of, and sequences of engineering design patterns as and when needed to address the specific and unique design challenges students face. Ford [3] defines the appropriate sequencing of activities as a practice and argues that practices should be judged by how the performances (e.g., a step of the engineering design process) that comprise the practice interact with the other performances undertaken. In our opinion, the aim for engineering design instruction is for students to be able to engage in an appropriate sequence of performances (i.e., a practice) instead of simply following a predefined sequence of steps (i.e., what Berland et al. [4] might call a rote performance). The patterns identified in the matrix resonate with our intuitions. In particular, this interpretation of engineering design that permits iteration and highlights sensemaking appears much more accurate to us than many of the linear, step-by-step engineering design process descriptions that are widely available. What remains an open question for us is whether or not instructional episodes can be accurately identified as targeting one or more of these engineering design patterns and whether or not consistent teaching patterns will emerge across multiple case studies.

Table 1. The informed design teaching and learning matrix [1].



	Patterns	*Descriptions of patterns*	
	Beginning VS Informed designers	*What beginning designers do*	*What informed designers do*
Explore the Challenge	**A. Problem Solving VS Problem Framing**	Treat design tasks as well-defined problems and make decisions prematurely, often right after reading the design brief.	Delay making design decisions in order to explore, understand and frame the design problem.
	B. Skipping VS Doing Research	Skip doing research and instead pose or build solutions immediately.	Do research and hands-on investigations to learn more about the problem and possible solutions.
	C. Idea Fixation VS Idea Fluency	Get stuck on their first design ideas that they won't let go of.	Practice idea fluency via brainstorming, lateral thinking, idea incubation, etc.
	D. Surface VS Deep Drawing and Modeling	Sketch ideas or make models of devices that would not work if built.	Use words, drawings and models to investigate design ideas and explore how things work.
Choose, Test and Improve Ideas	**E. Ignore VS Balance Benefits and Tradeoffs**	Attend only to positive traits of favored ideas, and notice only drawbacks of lesser approaches.	Weigh both benefits and tradeoffs of all ideas before making design decisions.
	F. Confounded VS Valid Tests and Experiments	Do few or no prototype tests, or run confounded experiments when attempted.	Conduct and analyze valid experiments to learn about key design variables or to optimize product performance.
	G. Unfocused VS Diagnostic Troubleshooting	Use a generalized, unfocused way of observing when testing and troubleshooting prototypes.	Focus attention on key problem areas when diagnosing and troubleshooting ideas or devices.
Use Effective Design Habits	**H. Dysfunctional VS Collaborative Design Work**	Team members are uninvolved *OR* work in isolation *OR* individuals dominate group work and decision making.	Members of team collaborate and cooperate in performing different project roles and making key design decisions.
	I. Haphazard or Linear VS Managed and Iterative Designing	Designing is done haphazardly *OR* steps are done in a rigid sequence *OR* once in linear order.	Do design in a managed way, where ideas are improved iteratively via feedback, and strategies are used flexibly, in any order, as needed.
	J. Tacit VS Reflective Thinking	Do tacit designing with little self-reflection or monitoring of actions.	Practice reflective thinking by keeping tabs on design work and thinking.

Crismond and Adams [1] describe how each of the engineering design patterns may be mapped onto learning goals and teaching strategies. Table 2 shows this mapping for Pattern G, Troubleshooting, which is the focus that we chose for this WIP paper. The Learning Goals and Teaching Strategies columns in Table 2 provide potential entry points for understanding how an instructional episode observed may map onto a design pattern.

Table 2. Analytical framework used to guide this case study WIP paper (modified from [1]).

Design Strategies Beginning vs. Informed Pattern	**Learning Goals Where Students…**	**Teaching Strategies Where Students…**
G. Troubleshoot Unfocused vs. Diagnostic Troubleshooting	Diagnose and troubleshoot ideas or prototypes based on simulations or tests.	Follow troubleshooting steps: observe, name, explain, and remedy; Do troubleshooting stations/videos; Do modeling or cognitive training in troubleshooting

Crismond and Adams describe informed designers as focusing "*their attention on problematic areas of their potential solutions and products*" [1, p. 766-7, italics in original] and indicate that informed designers use troubleshooting to identify and eliminate flaws in their designs. Examples of troubleshooting are described for both conceptual design stages where mental simulations are required and for prototyping stages of design where physical testing of components and systems are often required. Four possible teaching strategies suggested by Crismond and Adams [1] include:

- **Diagnostic troubleshooting:** a four-step process that requires students to observe, diagnose, explain, and then remedy observed faults in designs or systems.
- **Cognitive training:** instruction aimed to help students understand how systems and devices work, what principles govern the operation of these components, and describing case studies of prototypical failures that students may latter draw analogies from.
- **Troubleshooting stations:** instructional method where students are intentionally provided poor performing designs and scaffolded in identifying the cause(s) of the problems and asked to improve the performance of the component.
- **Teacher modeling:** a form of coaching in which a teacher demonstrates for students how they analyze a component that is not performing well.

In addition to describing four teaching strategies that may address troubleshooting, Crismond and Adams [1] identify troubleshooting as being closely related to conducting experiments (Pattern F) and indirectly highlight that troubleshooting is frequently used as a means to revise/iterate (Pattern I) on a design concept. The four teaching strategies described above, along with anticipated interactions between troubleshooting and conducting experiments and/or revising/iterating design concepts, provides a more nuanced lens for viewing teaching strategies targeting the troubleshooting pattern.

Methods

We selected case study research (CSR), a form of qualitative research that is used to study contemporary phenomena in which the investigator has little or no control and where context matters, due to its ability to produce analytic generalizations [5]. An analytic generalization seeks to generalize findings back to theory by either confirming, rejecting or extending theoretical propositions. We adopt the matrix as an analytical lens in hopes to contribute to the research communities theoretical understanding of engineering design instructional practices.

Crismond and Adams acknowledge that “while some of these instructional approaches have undergone empirical testing, others have not, which highlights a critical need for future educational research” [1, p. 747]. Our tacit assumption is that by studying experienced engineering design educators in well established design course that when we observe instructional interventions they will be in the service of improving students’ development along one or more of the patterns. Additionally, we believe CSR is a particularly appropriate method for this study because the method permits teaching practices to be studied in the context of a real classroom. The classroom setting within our case study contrasts the laboratory setting used by a large number of studies that have informed the development of the matrix (e.g., [6]-[9]). The controlled conditions of these research studies do not accurately reflect engineering practice which often requires engineers to work on teams over long durations to solve complex problems. Additionally, the clinical setting does not reflect an educational setting in which a teacher is available to help guide and scaffold students’ learning. This last point is particularly germane to us as we are concerned with how students can be supported by teachers in an effort to help them develop into more competent engineering designers.

A pilot case study of instructor Holmes (pseudonym) was completed during a three-week long undergraduate engineering design course offered in the summer of 2017 to a select group of students interested in pursuing engineering upon graduation from high school. We collected interview data, observed all classroom instructional periods, and collected course documents. In total, three hours of formal interview data was recorded, field notes covering nearly 60 hours of classroom instruction were generated, over 30 course documents were collected, and three hours of audio memos were recorded. The field notes generated are informed by our direct observation of instructional interventions as well as by informal conversations with Holmes during the observation. It is important to note that the data collection effort focused exclusively on instructor Holmes and not on the ~40 students in the classroom. We chose to focus on the instructor only in an effort to better understand teaching practices and instructional/curricular decisions an experienced instructor makes when teaching an engineering design course. In this paper we focus on instruction targeting troubleshooting in the pilot case because this case/pattern provides the richest data we have collected and thus serves as a test to evaluate the utility of our methodological approach. The two research questions that this paper attempts to answer are:

1. How does an instructor employ instructional interventions to support student learning along the troubleshooting pattern?
2. How does an instructor employ instructional interventions to help students sequence performances related to troubleshooting into coherent engineering design practices?

Findings

In our first interview, Holmes – a full professor in civil engineering with an active engineering research program, but only limited exposure to scholarly work in engineering education – described engineering design as a set of activities very similar to the activities included in the informed design teaching and learning matrix. Holmes described engineering design learning outcomes in terms of five activities. The activities he described are establishing requirements / understanding the goals of the design, modeling (applying physics concepts to generate mathematical representations of the system to predict performance), prototyping, debugging, and functioning on teams. In addition, Holmes describes student engagement with these activities as being a cyclic (iterative) process in which students often revisit one or more activities as new information is learned. This paper focuses on Holmes' instructional efforts towards the troubleshooting pattern or what he terms debugging. This is an area where Holmes feels particularly comfortable and competent. The working definition of debugging that Holmes provided during our pre-course interview is:

> But debugging any- debugging the mechanics, the physics, the electronics, the computer, all of that. Being able to observe that the thing is not doing exactly what you hoped it would do and to figure out in a logical and fruitful way where the source of that problem was likely to originate. So, you know, like sorta winnowing things down, ruling out this, ruling out this, ruling out something else until ultimately you are left with the only thing left that could possibly be the problem. And part of that is designing experiments that aren't necessarily the actual use case for the final product. They are strange little experiments that only exercise maybe one dimension of how the product works and holds everything else fixed but that's how you start to rule things out. If you don't see the problem manifest itself in that particular experiment then it's probably not there and you can rule that thing out and focus on the other thing and work your way down. And, just, I don't know a better way to teach that sorta stuff then by example and just kinda making sure they know the logical process of starting with well it could be this, it could be this, it could be this, and don't rule anything out. And then, then when you have to start ruling things out there's some economies involved when you try to think about well what's the most likely thing for it to be and maybe chase after that first, what's the most likely thing for it not to be so rule that out quickly so that you don't have to worry about it, and, but also, what's the most economical thing to rule out.

The working definition that Holmes provides for troubleshooting confirms a number of theoretical propositions identified in the matrix [1], but also provides areas where these descriptions may benefit from refinement. First, Holmes describes a process that is very similar to diagnostic troubleshooting in that it requires observing a flaw and finding a remedy or solution to the problem. However, the description Crismond and Adams [1] provide include only two intermediate steps of diagnosing and explaining the flaw. These steps are presented in an unproblematic way leading the reader to believe that a correct diagnosis is possible through careful observation. The description Holmes provides indicates that iteration is often necessary in troubleshooting and that it may not be possible to attribute an observed flaw to a single explanation through a linear four-step process. Instead, Holmes' definition makes clear that many possible explanations may exist and the task of troubleshooting requires one to identify which explanation is correct in order to remedy the problem. In addition, the definition Holmes provides highlights the importance of what he describes as economical considerations that are part of the troubleshooting process. This viewpoint indicates that a prioritization must be made regarding the order in which different possible faults get tested. This prioritization is established to find the fault in the shortest amount of time and/or at the lowest cost.

Holmes' view is that troubleshooting and conducting valid experiments are inseparable in all but the simplest cases. This provides evidence to support Crismond and Adams' [1] claim conducting experiments is highly germane to development along the troubleshooting pattern. In addition, Holmes suggests that demonstrating by example the diagnostic troubleshooting process is the best way he has found to help students develop into more informed designers along this pattern. Crismond and Adams [1] termed this teaching method as teacher modeling.

The commitment that Holmes expressed in the first interview at getting students to develop into more informed engineering designers along the troubleshooting pattern clearly extended to the classroom instructional activities he offered to students. During one classroom observation, Holmes began class with a mental simulation related to the quintessential introductory Arduino "Blink" sketch presented as an activity in class the previous day. This activity has students turn an LED connected through an external circuit on for one second and off for one second with the LED light continuing to blink for as long as the Arduino remains powered. In this mental simulation that lasted approximately 30 minutes, Holmes asked his students to recall the "Blink" exercise completed the previous day and to imagine that they have the exact same setup as yesterday but that it no longer works. He then asked students to list possible explanations for why the setup no longer works. Students offered more than a dozen explanations which Holmes listed on the whiteboard ranging from the LED was plugged in backwards (one of many possible hardware issues identified) to a different Arduino code was uploaded onto the Arduino (one of a few possible software issues identified). Holmes took this list that he wrote on the whiteboard and systematically described the troubleshooting process he would use to find the flaw. He carefully went through the list with a philosophy to rule out the most likely and/or easiest to diagnose problems first and to leave the more difficult or costly possible faults to be examined only if necessary later on. Holmes was very careful when describing what experiments he would use to test for a given fault and what the limitations of those experiments were. For example, he cautioned that an easy way to test if an LED might be burned out is to switch it with another LED, but that in doing so there are no guarantees that the new LED may not also be burned out… or that the new LED may have been plugged in with the incorrect polarity… or the new LED may be plugged in correctly and may be working, but may emit IR light which is not visible to the human eye. This 30-minute lesson that was focused on troubleshooting included clear examples of how the troubleshooting and conducting experiments patterns may be mutually supportive in engineering design activities. The lesson ended with a few takeaways related to troubleshooting that were often returned to later in the course. These key takeaways include realizing the possibility that more than one problem may exist when something does not work, that new problems may be created when attempting to solve old problems, and that it is essential to systematically make a list of all possible explanations for why something is occurring and then to logically cross of one item after another, starting with the low hanging fruit, until a sensible explanation is reached.

The above excerpt provides evidence that Holmes has established a learning outcome for students to develop into more informed engineering designers by improving their ability to function along the troubleshooting pattern of the framework. Holmes leads a classroom activity that requires students to conduct a mental simulation of one possible fault and then describes a process by which the fault can be debugged. This classroom activity provides further evidence to support the notion that teacher modeling of the diagnostic troubleshooting process may be an appropriate instructional intervention to include within an engineering design class. However, the diagnostic troubleshooting procedure employed provides more nuance and less certainty than the version described within the informed design teaching and learning matrix. Specifically, Holmes' first interview and our classroom observations combined suggest a possible alternative diagnostic troubleshooting approach that may be more appropriate for undergraduate engineering students. This process may be re-written as:

1. Observe a flaw
2. Generate a list of all possible explanations for the flaw
3. Select an appropriate order to begin testing each potential explanation
4. Conduct controlled experiments to test each potential explanation until all flaws are found (iterating and revising explanations and the testing order as needed)
5. Remedy the flaw

Approximately half of the time that students spent in Holmes' class was dedicated to working in small teams to develop a prototype of a robotic platform capable of achieving a number of predefined performance goals. We have numerous examples within our field notes of instruction aimed at helping students to troubleshoot design flaws during the prototyping phase of the course. These observations were all examples of teacher modeling of the diagnostic troubleshooting process. During these teacher modeling episodes Holmes frequently referenced the aforementioned class discussion related to the process used to troubleshoot the fictitious Arduino blink exercise while he actively modeled his troubleshooting procedure to the unique problem students approached him with. Most of these instructional interventions had durations of under ten minutes and frequently occurred after students attempted to solve the problem on their own for some period of time. These interactions occurred throughout the entirety of the prototyping process with no obvious phase of the project where these episodes were most frequent. With that said, these episodes most often revolved around electronics and programming faults.

In our observations, Holmes put in considerable effort to be prepared to help students with the flaws they would likely encounter within the project. Holmes had first-hand experience with many of the obstacles students encountered that required some degree of troubleshooting in order to proceed forward. He also required that his teaching assistants have first-hand experience with essentially all of the hardware that students may choose to use on their design projects. This teaching practice – ensuring proper preparedness by gaining first-hand content knowledge of the troubleshooting problems students are likely to encounter – appears important to Holmes. We speculate that Holmes' preparedness permitted him to have more instructive teacher interactions with students using the teacher modeling instructional approach. This is evident by the fact that he was frequently able to verbally narrate his actions and the process he was employing when solving any number of different and authentic problems that cropped up on student projects. In essence, Holmes was able to remedy the design flaw for students while narrating how he is using a diagnostic troubleshooting process to solve the problem.

Holmes described the teacher modeling strategy that he employed in a second interview conducted immediately after the course concluded. In his words:

> If the students comes to me with, you know, Dr. Holmes blah blah blah this isn't working can you help us? A small part of what I'm trying to do is to go sit down with them and to solve the problem. But to be perfectly honest, I think I go into each one of those with the understanding that these students are watching me and they're watching- their not just hoping that I solv- they don't just want to run away and I solve the problem and they come back and its problem solved. They are gonna watch the process and so I have to think about, just myself, demonstrating the proper process. And so I ask a lot of questions, I want to see everything as clean as possible, I'll figure out, you know, all of these different things and you know what's this doing and what's that doing and let's comment that out and let's comment this out, let's get rid of this, let's get rid of that, let's unplug this, let's unplug that. And I'm trying to do both at the same time, I'm trying to demonstrate to them this idea of isolated experiments that can identify what we're looking for. And at the same time I'd really love to solve their problem. I have more than one time uttered under my breath please let it be this as I'm sticking something back together. Because if it is then then not only will I have shown them a process but I will have shown them a process that ultimately works and they'll have confidence in what I showed them. So, I think that's important and I think it's important for us to teach and that's a very direct way, but a somewhat indirect way, is, that since we do spend so much time sitting down with them solving problem I think it's incumbent on the instructors to actually demonstrate this process as they solve students problems. In many cases it would be very easy to say ‘ah trust me take this it'll work’, but that's not really the right way to go about it.

The above description that Holmes provides was highly consistent with what we observed during multiple episodes within the prototyping phase of the design project. We witnessed teacher modeling of troubleshooting essentially every class meeting period, and sometimes multiple times during a given class, throughout the prototyping phase. The description of the teacher modeling approach that Holmes provides clearly shows the inseparable link that he sees between troubleshooting and conducting valid experiments. His description and our observations frequently highlighted instances where he modeled troubleshooting processes in which isolated and controlled tests were required to evaluate whether or not a hypothetical fault was the actual cause of the design flaw. In some of these cases, theatrics were included to highlight just how important it is to conduct isolated experiments. Holmes frequently ripped all wires out from a team’s breadboard or Arduino to isolate possible faults and in doing so highlighted the importance to students of running controlled experiments. This provides further evidence that instruction related to the troubleshooting and conducting experiments are highly coupled and that teacher modeling of troubleshooting should include thoughtful inclusion of how conducting isolated experiments are a necessary component of troubleshooting.

Discussion

Our case study provides evidence that two of the teaching strategies included in the matrix [1] were employed by Holmes. The first method was diagnostic troubleshooting. Our data suggests that Holmes had a well defined procedure in mind to help students develop troubleshooting processes to correct faults to improve the performance of their designs. Our data suggests, however, that the diagnostic troubleshooting procedure included in the matrix is too simplistic. Our findings suggest that diagnosing a fault is not always a simple process that can be achieved by observing the behavior of the system. Instead, effort must be made at the start of a troubleshooting episode to list all possible explanations for the fault and prioritization efforts are required to establish an order for testing each explanation. Our findings are supported by literature related to developing competency with troubleshooting in other disciplines. Specifically, others who have studied troubleshooting have suggested that strategic knowledge is an essential element of troubleshooting and that this knowledge refers to prioritizing which hypotheses to test or choosing the most expedient testing methods for successfully discovering the faults within a system [10]-[12]. Our findings suggest that helping students develop strategic knowledge of troubleshooting is a necessary component of this pattern. Taken together, our findings and literature from other disciplines on troubleshooting suggest that diagnostic troubleshooting within engineering should include consideration of multiple viable explanations for a fault and a thoughtful prioritization for how/where to begin testing for the fault.

The data we provide also suggest that the teacher modeling instructional strategy was frequently employed, particularly during the prototyping portion of the course when physical faults in design hardware occurred. More recent scholarly efforts outside of engineering design contexts have also pointed towards modeling as an effective instructional method for helping students to develop into more informed troubleshooters [10]. Dounas-Frazer and Lewandowski [10] draw on the notion of cognitive apprenticeship [13] to describe teacher modeling through their research related to troubleshooting in electronics lab courses. They break down cognitive apprenticeship into five teaching methods applicable to troubleshooting instruction:
(1) modeling (having an expert troubleshoot a fault while narrating the mental process), (2) coaching (offering students tips/hints from the sidelines), (3) scaffolding / (4) fading (providing supports to aid students early on and then slowly reducing the number of supports available), and (5) articulation (having students verbalize their approach). Our field notes suggest that the two instructional methods most frequently used by Holmes in an undergraduate engineering design context were modeling and coaching. Our findings lend support to the inclusion of these two methods within the informed design teaching and learning framework [1].

In conclusion, our case study of instructional interventions targeting design patterns affords us with an opportunity to closely examine teaching practices aimed at helping students to develop into more informed engineering designers. By studying instructional practices within the context of real classrooms we provide evidence to support theoretical propositions regarding teaching methods appropriate for inclusion within engineering design courses. In doing so, we provide descriptions of how and when within an engineering design course different teaching strategies get employed and how different teaching strategies may target one or more engineering design pattern. Our research design allows us to offer nuanced revisions to the highly concise descriptions of teaching strategies currently available within the engineering education literature.

Acknowledgments

The author would like to thank the university, the teacher, and the students who participated in this study. Thank you for allowing me to observe your design activities throughout the course. In particular, I would like to extend a special thank you to Holmes for opening his classroom for observation and for freely sharing his expertise on teaching engineering design with me.

References

[1] D. Crismond and R. Adams, “The informed design teaching and learning matrix”, *Journal of Engineering Education*, vol. 101, no. 4, pp. 738-797, 2012.

[2] D. Crismond, “Scaffolding strategies for integrating engineering design and scientific inquiry in project-based learning environments,” in *Fostering human development through Engineering and Technology Education*, pp. 235-255, SensePublishers, 2011.

[3] M. Ford, “Educational Implications of Choosing “Practice” to Describe Science in the Next Generation Science Standards”, *Science Education*, vol. *99*, no. 6, pp. 1041-1048, 2015.

[4] L. Berland, C. Schwarz, C. Krist, L. Kenyon, A. Lo and B. Reiser, “Epistemologies in practice: Making scientific practices meaningful for students”, *Journal of Research in Science Teaching*, vol. 53, no. 7, pp. 1082-1112, 2015.

[5] R. Yin, *Case study research: Design and methods*. Sage publication, 2013.

[6] R. Adams and C. Atman, “Characterizing Engineering Student Design Processes: An Illustration of Iteration”, *ASEE Annual Conference Proceedings*, pp. 1271-1281, 2000.

[7] C. Atman, K. Chimka, K. Bursic and H. Nachtmann, “A comparison of freshman and senior engineering design processes”, *Design Studies*, vol. 20, no. 2, pp. 131–152, 1999.

[8] C. Atman, R. Adams, M. Cardella, J. Turns, S. Mosborg and J. Saleem, “Engineering design processes: A comparison of students and expert practitioners”, *Journal of Engineering Education*, vol. 96, no. 4, pp. 359-379, 2007.

[9] C. Atman, K. Deibel and J. Borgford-Parnell, “The Process of Engineering Design: A Comparison of Three Representations”, in *DS 58-1: Proceedings of ICED 09, the 17th International Conference on Engineering Design*, vol. 1, 2009.

[10] D. Dounas-Frazer and H. Lewandowski, “Electronics lab instructors’ approaches to troubleshooting instruction”, *Physical Review Physics Education Research*, vol. 13, no. 1, 2017.

[11] D. Jonassen and W. Hung, “Learning to Troubleshoot: A New Theory-Based Design Architecture”, *Educational Psychology Review*, vol. 18, no. 1, pp. 77-114, 2006.

[12] A. Schaafstal and J. Schraagen, “Training of troubleshooting: A structured, task analytical approach”, in Schraagen, J. M., Chipman, S. F., and Shalin, V. L. (eds.), *Cognitive Task Analysis*, Lawrence Erlbaum Associates, Mahwah, NJ, pp. 57–70, 2000.

[13] A. Collins, J. Brown and S. Newman, “Cognitive Apprenticeship: Teaching the crafts of reading, writing, and mathematics”, *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, pp. 32-42, 1989.