Studying How Digital Logic Instructors Solve Canonical Problems

Geoffrey L. Herman
University of Illinois at Urbana-Champaign
Urbana, IL, USA
glherman@illinois.edu

***Abstract*— Sketches and other forms of graphical communication are central to both the practice and learning of engineering. Visual representations play a critical role in helping students learn engineering concepts, socialize them into the engineering discipline, and facilitate or hinder the design process. Despite the importance of graphical communication and visual representations, our understanding of how students learn these representations and use them is poor. This paper describes part of a larger expert-novice comparison study to document how sketches moderate engineering problem solving. We present preliminary findings on how digital logic instructors use sketches to solve canonical problems.**

Keywords*— *digital logic; problem solving; state machines; cognitive interviews; grounded theory

I. Introduction

Sketches and other forms of graphical communication are central to both the practice and learning of engineering [1-4]. They play a critical role in helping students learn engineering concepts, socialize them into the engineering discipline, and facilitate or hinder the design process [5]. Despite the importance of sketching and visual representations more generally, our understanding of how students learn these representations and use them is poor [6, 7].

To better fully understand how students' use of sketching helps or hinders their learning, it can be helpful to juxtapose students' process with those of experts. Consequently, we are conducting a series of novice-expert comparison studies, comparing sketching behaviors and how they intersect with conceptual understanding. Investigations into the differences between experts and novices in their ability to process and recall information have provided a critical foundation in understanding how people learn [8]. Knowledge of these differences have led to the creation of foundational educational theories (e.g., ontological shifts [9, 10]), assessment tools (e.g., concept inventories [11]), and research-based instructional practices suitable for the classroom (e.g., bridging analogies [12]). Studies on how people learn about scientific diagrams have revealed that experts and novices find different elements of diagrams salient and that they chunk visual information differently [8, 13, 14]. Experts emphasize underlying processes and functions in diagrams (e.g., focusing on the cycle of evaporation and rain illustrated in a diagram) while novices focus on surface features (e.g., focusing on the fact that there are clouds, lakes, and the sun in that same diagram) [14] – domain knowledge influences perception of visual representations.

We build on this research, exploring how dynamic diagrams (i.e., sketches) affect problem solving. While this project intends to explore expert-novice differences across a number of engineering disciplines, we have started by exploring these differences in how students and professors solve finite state machine problems from digital logic courses. During the 2014-2015 school year, we interviewed 27 students and 9 instructors of digital logic courses, recording their sketches and their verbal explanations as they solve canonical and open-ended design problems. In this paper, we focus on our preliminary analysis of the instructor interviews through the lens of methodological considerations. We present the design of our study, our analysis methods, and then discuss methodological considerations that arise from preliminary findings.

II. Background

A. Data Set and core STEM classes

Our study was guided by DiSessa's Knowledge in Pieces framework that argues that students' knowledge is fragmented lacking coherence across contexts [15]. Novices construct their knowledge on demand, marshalling pieces of information that are cued by the context of the problem. Prior research on students' conceptual understanding of engineering concepts has revealed that students' ability to articulate their understanding is altered by perturbations in the visual presentation of the content [16, 17]. For example, prior research on students' understanding of state machines revealed that on average students construct four different, often mutually exclusive, conceptions of state [17]. Students fluidly and unconsciously switch between these conceptions across contexts [17]. Similarly, when students discussed shear stresses and strains in mechanics of materials, their reasoning was dependent on the physical orientation of the members undergoing axial loads or the physical orientation of stress elements drawn on the members [18-20]. Students' reasoning overly relied on physical orientation, conceiving of "shear" as a *vertical* construct. Students would routinely replace the word "shear" with "vertical" during reasoning [18-20], but this linguistic shift is only viable in certain circumstances.

In digital logic, the context-dependence of students' conceptual understanding was similarly revealed. When

This work was supported by the College of Engineering and the National Science Foundation grant EEC 1429348.

analyzing Boolean logic problems, students adopted dramatically different solution procedures and cued different knowledge based on the visual presence of a truth table [16]. Students were given exactly the same problem twice (except for the presence of a truth table) separated by several other problems. Despite solving the same problem, students adopted different procedures and revealed different conceptions [16]. Without the truth table, students would conflate concepts such as implication and conjunction [16]. With a truth table present, students maintained the distinction between implication and conjunction. These distinctions were observed regardless of presentation order [16].

This prior work revealed that different representations of the same information have different affordances for the problem solver. Different representations make some knowledge explicit while other knowledge is tacitly assumed. For example, in the example of truth tables, the representation affords and explicitly encourages the problem solver to exhaustively explore every case, whereas if the problem solver is simply asked to derive a Boolean logic expression, the need to exhaustively explore cases is implicit. For students with unreliable conceptual understanding, the use of exhaustive search strategies was necessary to remember the cases that distinguished implication and conjunction [16].

These prior studies are limited in that they focused solely on the representation that the researchers presented to the students and not on how students produced their own representations or how students' own representations constrained or enhanced their thinking. To begin studying the differences between how engineering novices and experts use and produce sketches, this following study sought to answer the following research questions: 1) How do digital logic instructors produce sketches when designing sequential circuits? and 2) How do the practices of instructors differ from those of students?

III. Methodology

To explore the importance of the affordances of different representations, we are specifically examining how students and instructors differentially solve problems that explicitly require the use of multiple representation transformations. In this paper, we present initial findings about these processes as they relate to transforming finite state machine diagrams into sequential circuits. Because few engineering disciplines require digital logic, we begin by defining the relevant concepts of state and sequential circuits.

A. Terminology, Concepts, and Diagrams

A finite state machine (FSM) consists of a finite set of *states* S, a finite set of *input* symbols I, a finite set of *output* symbols O, a *transition* function $\delta: S \times I \rightarrow S$, and an *output* function $\omega: S \rightarrow O$ (assuming a Moore model finite state machine). The transition function maps each pair of a state and an input symbol to a *next state* in S. The output function maps each state to an output symbol. These functions can be represented graphically using a *state diagram* as shown in Figure 1 or diagram 1 in Figure 2. During the design of a FSM, the states are typically labeled with some meaningful name to facilitate interpretation of the diagram. When designing a sequential circuit, each state is given an arbitrary and unique binary state encoding. These encodings each have the same number of binary bits (i.e., fixed width) and can be selected to optimize the design of the sequential circuit. In Figure 1, the state on the left is encoded as 00. When the system is in state 00, the system will output the value of 1. An input value of 0 during this state will cause the system to transition back to state 00. An input value of 1 during this state will cause the system to transition to state 01. A diagram key is often included as part of the diagram to indicate which variable names are assigned to the state encodings, input encodings, and output encodings (far right portion of diagram 1 in Figure 2). Encodings are generally indexed in descending order so that the indices represent the power of two assigned to each position, enabling the encodings to be interpreted as unsigned binary numbers (Q_1Q_0 = 10 is read as "state 2"). The current state Q_1Q_0 is distinguished from the next state $Q_1^+Q_0^+$ with the use of a superscript '+' sign.

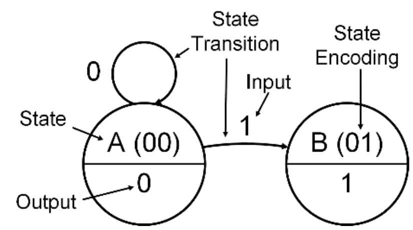
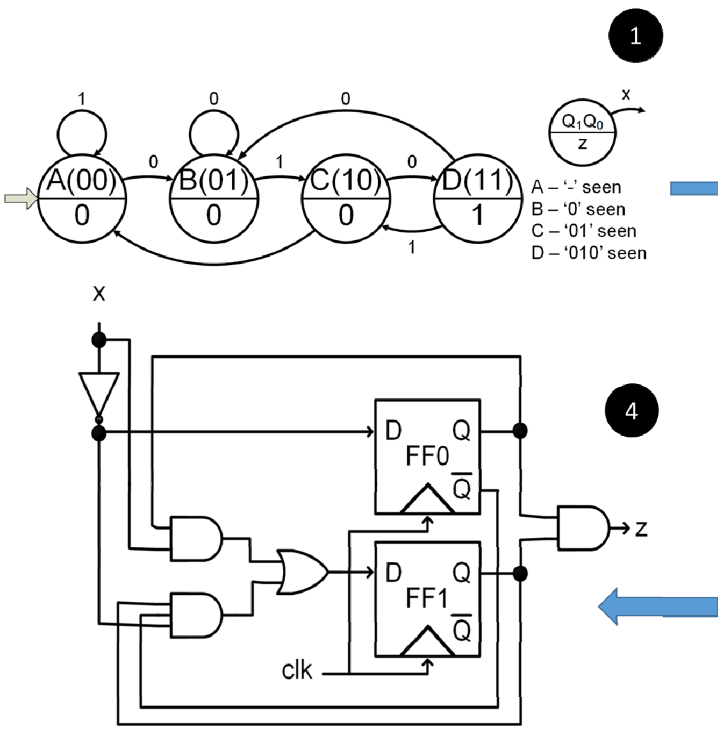


Fig. 1: Partial state diagram with parts of the state machine labeled

To design a sequential circuit, the problem solver generally needs to translate the state diagram into a tabular form, such as a next-state table (see diagram 2 in Figure 2). In the next state table, state encoding variables are typically assigned to the left-most columns with input variables in the next column to the right. The next states and outputs are then listed in the following columns as they are functions of the current state and inputs. The problem solver chooses a state (e.g., state A with state encoding 00 in Figure 2) and determines the output based on the current state (e.g., output 0 for state A) and the next state based on the current state and value of the input (e.g., the top row of the next state is 01 for state A).

The tabular representation is translated into Boolean expressions (see diagram 3 in Figure 2) that determine how the circuit will be implemented. Assuming that D-type flip-flops are used in the circuit, the next-state encoding and the D input of the flip-flops are equivalent, so D is often used in place of Q+ in these equations.

Finally, the Boolean expressions are used to construct a schematic for the sequential circuit. The state variables are stored in circuit devices called flip-flops (the boxes labeled FF) that has a device input (D) and device output (Q). A flip-flop stores its state until the system clock triggers it to change its state to the value being sent to its input D. This structure enables the system to be stable even though it has feedback loops. AND, NOT, and OR operators in the Boolean expression are translated into logic gates (D-shaped, triangle-shaped, and arrowhead-shaped respectively). The device-level outputs of these gates are used to compute the D input of the flip flops and the system output.



2

Current state		Input	Output	Next state (FF input)	
Q_1	Q_0	x	z	Q_1^+ (D_1)	Q_1^+ (D_0)
0	0	0	0	0	1
0	0	1	0	0	0
0	1	0	0	0	1
0	1	1	0	1	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	0	1
1	1	1	1	1	0

3

$$D_1 = Q_0 x + Q_1 \overline{Q_0} \bar{x}$$
$$D_0 = \bar{x}$$
$$z = Q_1 Q_0$$

Fig. 2: The series of transformations that a subject would likely use to design a sequential circuit. Sub-figure 1 is a state diagram, sub-figure 2 is a next-state table, sub-figure 3 is a set of Boolean expressions, and sub-figure 4 is a circuit diagram

B. *Sampling*

In Fall 2014 and Spring 2015, we interviewed nine digital logic instructors from seven institutions. These instructors serve as our expert sample. The instructors' institutions varied in size (from 1,000 to over 40,000 students), institution type (public and private), and in geographical location. Faculty were recruited to participate in interviews while they were in attendance at education conferences such as the ACM Special Interest Group in Computer Science Education conference and the ASEE Annual Conference & Exposition. We required that instructors had to have taught a digital logic course within the past year to participate. In contrast, our novice sample consisted of students who were all from the same institution and had passed a digital logic course within the past month. Participants will henceforth be called *subjects.*

C. *Interview Process*

Subjects were interviewed for one hour. Interviews were conducted in a modified "think-aloud" format: Subjects were instructed to vocalize their thoughts as they solved problems and responded to questions. Prior to the interview, subjects were briefed on the study's goal of understanding how students and instructors design sequential circuits. They were told not to expect feedback during the interviews about whether their designs were valid, but to expect frequent requests to elaborate on what they were doing.

All interviews were conducted on a tablet computer in Microsoft OneNote. The tablet provided a digitizer stylus with an eraser to create a near pencil-and-paper experience in the digital environment. Interviews were recorded using Camtasia for screen capture and audio.

Subjects were paid for their participation, and all subjects gave written consent to be interviewed under IRB approval (University of Illinois at Urbana–Champaign number 15065).

D. *Interview Questions*

All subjects were interviewed using the same protocol. The interview consisted of two portions: the training tasks and the problem solving tasks.

The subjects were given a training task to familiarize them with the digital pen and tablet environment, to the process of vocalizing their thoughts, and to collect baseline data on subjects' drawing abilities and patterns. Subjects were informed that the task was a training task. They began by drawing shapes (lines, arrows, squares, etc.), the alphabet, and numbers 1 to 10. Subjects then copied a cartoon drawing of a banana. The final training task asked subjects to copy a simple, computer-generated state diagram (See Fig. 3).

For the problem solving tasks, subjects were given four different design tasks of increasing difficulty. The first three design problems can be considered canonical design problems with known strategies for their completion. The fourth design problem is a complex, ambiguous design problem that while solvable is not intended to be solvable during the interview. The goal of this problem was to observe what the problem solvers did when they did not know what to do.

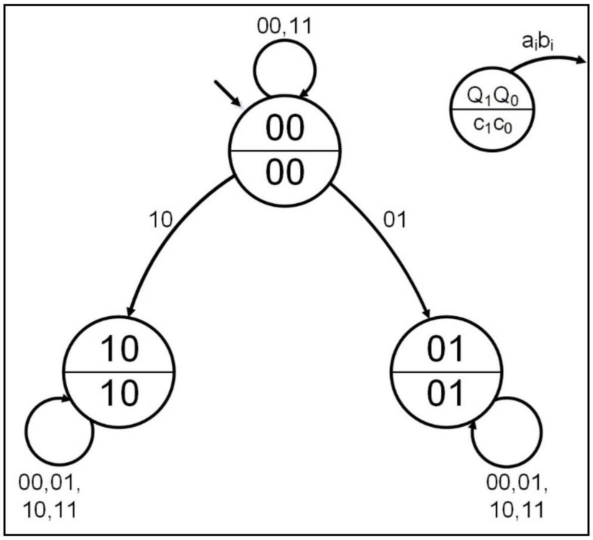


Fig. 3: State diagram that subjects copied during the training task

E. Analysis Method

Interviews were analyzed using the *constant comparative method [21, 22]* without an a priori coding scheme, but with an awareness of prior research on conceptual understanding, problem solving, and comprehension of diagrams. Analysis was performed directly from the video data using MaxQDA so that verbal statements could be linked to sketching behaviors. Trustworthiness of the analysis is promoted through rigorous documentation of the comparisons that are being made and maintaining an audit trail of emergent codes and themes [21]. Analysis and comparisons were made along four units of analysis of different granularities [21]: 1) the subject, 2) the problem, 3) the transformation and 4) the statement.

Subject unit of analysis: Each subject was categorized as being uninformed, a mid novice, an advanced novice, or an expert based on the accuracy and quality of their solutions across the interview. This categorization was made by consensus between the researchers.

Problem unit of analysis: Analysis of each subjects' solution was described holistically. For example, a student might reveal a robust method for designing FSMs but a poor conceptual understanding of counter circuits.

Transformation unit of analysis: Analysis focused on how a subject revealed their conceptual knowledge within a specific transformation task. For example, a subject might reveal a correct understanding of state while drawing a state diagram, but reveal an incorrect understanding of state while drawing a circuit diagram.

Statement unit of analysis: Each statement a student made or figure they drew was analyzed to document a students' conceptual understanding as revealed in the moment. For example, a subject might refer to the output of the circuit (O) as the next state (Q+) revealing a contextual conflation of the two concepts.

The goal of the constant comparative method is to make comparisons within and across these units of analysis to collect evidence for and against emergent themes and theories[21]. In this analysis approach, the researchers focus on maintaining thick descriptions of observations to facilitate comparisons rather than on reducing these observations to a strict coding scheme[21]. In the description that follows, a theme is a holistic description of an observed behavior that held true across multiple units of analysis and across multiple subjects. A theory is a framework that organized recorded themes into an interpretive narrative.

IV. Methodological Considerations

The most immediate methodological challenge to interviewing experts has been recruitment. Despite offering financial incentives to faculty, response rates to interview requests were extremely low (less than 1%). While we hoped that faculty attending engineering education conferences would have a higher interest in participating in the study, it seems that busy conference schedules were prohibitive for participation.

Preliminary analysis of the expert interviews has revealed methodological challenges for analysis. First, notational standards varied between institutions, even when both institutions used the same textbook. During the training activities, all faculty spent time negotiating the meaning of the notation in Fig. 3. Notably, though, while novices who did not understand the notation were content to call symbols by symbolic names (e.g., 1s, 0s, Qs, Cs, arrows), instructors sought to identify meaningful names for the symbols (e.g., state, inputs, outputs). Second, some instructors were not familiar with the term "mod-counter" that appeared in one problem specification. Consequently, those instructors were not able to parse the circuit specifications and struggled to make sense or meaning of that problem. Third, instructors often vocalized their thoughts from the position of a teacher justifying their teaching methods, saying things like, "I would tell my students to…" or "Should I explain this like I would to my students?" This difference in metacognition was not part of the original coding scheme or research questions, but these types of comments call into question the different ways that experts and novices experience the think-aloud interview.

Analysis has revealed significant differences in the types of codes generated from the data. Instructors began their problem solving with sense making, verbally describing the behavior of the circuit before sketching. This type of sense making frequently enabled the instructors to bypass the structured problem solving approach illustrated in Fig. 2 to arrive at a solution. These type of approaches makes it difficult to make one-to-one comparisons between experts and students who always used the structured approach in Fig. 2.

Additionally, instructors engaged with problems in a dialogue, "correcting" or modifying the problem specification when they misread or failed to understand the problem. In contrast, students never questioned or modified the problem statements and rarely made assumptions during their problem solving activities.

These findings suggest fundamentally different meta-cognitive and epistemological approaches between instructors and students. In future work, we will more fully explore the details of instructor and student cognition as it pertains to specific representations.

REFERENCES

[1] J. Juhl and H. Lindegaard, "Representations and visual synthesis in engineering design," Journal Of Engineering Education, vol. 102, pp. 2-19, 2013.

[2] D. Jonassen, J. Strobel, and C. B. Lee, "Everyday problem solving in engineering: Lesson for engineering educators," Journal of Engineering Education, vol. 95, pp. 139-151, 2006.

[3] T. Litzinger, P. Vanmeter, C. Firetto, L. Passmore, C. Masters, S. Turns, et al., "A cognitive study of problem solving in statics," Journal of Engineering Education, vol. 99, pp. 337-353, 2010.

[4] K. Henderson, On line and on paper: Visual representations, visual culture, and computer graphics in design engineering. Cambridge, MA: MIT Press, 1999.

[5] T. J. Moore, R. L. Miller, R. A. Lesh, M. S. Stohlmann, and Y. R. Kim, "Modeling in engineering: The role of representational fluency in students' conceptual understanding," Journal Of Engineering Education, vol. 102, pp. 141-178, 2013.

[6] A. Johri and V. K. Lohani, "Framework for Improving Engineering Representational Literacy by Using Pen-based Computing," International Journal of Engineering Education, vol. 5, pp. 958-867, 2011.

[7] A. Johri, W.-M. Roth, and B. M. Olds, "The role of representations in engineering practice: Taking a turn towards inscriptions," Journal of Engineering Education, vol. 102, pp. 2-19, 2013.

[8] J. D. Bransford, A. L. Brown, and R. R. Cocking, How People Learn. Washington, D. C.: National Academy Press, 2000.

[9] J. D. Slotta and M. T. H. Chi, "Helping students understand challenging science topics through ontology training," Cognition and Instruction, vol. 24, pp. 261-289, 2006.

[10] D. Yang, A. S. Roman, R. A. Streveler, R. L. Miller, J. Slotta, and M. Chi, "Repairing student misconceptions using ontology training: A study with junior and senior undergraduate engineering students," presented at the Proceedings of the 2010 American Society for Engineering Education Annual Conference and Exposition, 2010.

[11] D. Hestenes, M. Wells, and G. Swackhamer, "Force concept inventory," The Physics Teacher, vol. 30, pp. 141-158, 1992.

[12] J. Clement, "Using bridging analogies and anchoring intuitions to deal with students' preconceptions in physics," Journal of Research in Science Teaching, vol. 30, pp. 1241-1257, 1993.

[13] M. T. H. Chi, P. J. Feltovich, and R. Glaser, "Categorization and representation of physics problem by experts and novices," Cognitive Science, vol. 5, pp. 121-152, 1981.

[14] B. D. Jee, D. Gentner, K. Forbus, B. Sageman, and D. H. Uttal, "Drawing on experience: Use of sketching to evaluate knowledge of spatial scientific concepts," presented at the Proceedings of the 31st Annual Conference of the Cognitive Science Society, 2009.

[15] A. A. diSessa, "A bird's eye view of 'pieces' and 'coherence' controversy," in Handbook of Conceptual Change Research, S. Vosniadou, Ed., ed Mahawha, NJ: Lawrence Erlbaum, 2008, pp. 35-60.

[16] G. L. Herman, M. C. Loui, L. Kaczmarczyk, and C. Zilles, "Describing the what and why of students' difficulties in Boolean logic," ACM Transactions on Computing Education, vol. 12, pp. 3:1-28, 2012.

[17] G. L. Herman, C. Zilles, and M. C. Loui, "Flip-flops in students' conceptions of state," IEEE Transactions on Education, vol. 55, pp. 88-98, 2012.

[18] D. B. Montfort, G. L. Herman, S. A. Brown, H. M. Matusovich, R. A. Streveler, and O. Adesope, "Patterns of student conceptual understanding across engineering content areas," International Journal of Engineering Education, vol. 31, pp. 1587-1604, 2015.

[19] D. Montfort, G. L. Herman, R. Streveler, and S. Brown, "Assessing the application of three theories of conceptual change to interdisciplinary data sets," presented at the Proceedings of the Forty-Second ASEE/IEEE Frontiers in Education Conference, Seattle, WA, 2012.

[20] G. L. Herman, D. Montfort, S. Brown, and R. Streveler, "Work-in-progress: Do students need to learn to speak engineering-ese?? Conceptual change as language acquisition in engineering," presented at the Proceedings of the Forty-Second ASEE/IEEE Frontiers in Education Conference, Seattle, WA, 2012.

[21] H. Boeije, "A purposeful approach to the constant comparative method in the analysis of qualitative interviews," Quality & Quantity, vol. 36, pp. 391-409, 2002.

[22] J. Corbin and A. Strauss, Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Thousand Oaks, CA: Sage, 2007.