

A Socio-cognitive Framework and Method for Studying Technology-mediated Problem Solving

Darren K. Maczka, Virginia Tech

Darren Maczka is a Ph.D. candidate in Engineering Education at Virginia Tech. His background is in control systems engineering and information systems design and he received his B.S. in Computer Systems Engineering from The University of Massachusetts at Amherst. He has several years of experience teaching and developing curricula in the department of Electrical and Computer Engineering at Virginia Tech.

Dr. Jacob R. Grohs, Virginia Tech

Jacob Grohs is an Assistant Professor in Engineering Education at Virginia Tech with Affiliate Faculty status in Biomedical Engineering and Mechanics and the Learning Sciences and Technologies at Virginia Tech. He holds degrees in Engineering Mechanics (BS, MS) and in Educational Psychology (MAEd, PhD).

A socio-cognitive framework and method for studying technology-mediated problem solving

Introduction

In this theory paper we describe a theoretical framework, data collection strategy, and methodology for studying engineering problem solving. A critical component of this work is the acknowledgment that contemporary engineering problem solving involves working with information systems technology, usually by way of a computer. Computer use is often required in the engineering classroom, and while there has been research on the use of computing technology to augment the learning environment with pedagogical tools,^{1,2,3} we still know little about how the computing tools that are presented as “tools of the trade” impact students’ perception of engineering problem solving.⁴ Understanding how these tools shape the problem solving process is important because familiarity with the tools can have a profound affect on students’ ability to be successful in class, even though tool proficiency may not be an explicit learning objective or pedagogical goal. In other words, by not addressing the intrinsic complexity of the tools themselves, instructors may inadvertently leave students with the impression that tool proficiency is a trivial part of engineering problem solving, which is not the case.

Unlike other engineering tools, the computer mediates interaction between not only the problem solver and the problem, but also the problem solver and socially constructed knowledge that is integral to the solution generation process. We will argue that conventional methods of studying the problem solving process are limited in their efficacy to understand the role information technology has on the problem solving process. Existing methods of studying problem solving tend to either ignore the process itself, looking only at inputs and outputs while situating information technology as an independent variable that might affect the end product,⁵ or extract information about the process via think-aloud protocols which have known limitations associated with the validity of information they can provide.^{6,7}

On the other hand, theoretical frameworks and methods for studying technology situated practice tend to be used to generate systems-level descriptions of interactions, for example, how video conferencing technology shifts power dynamics within a group.⁸ In other words, the unit of analysis is the relationship between a particular technology and social group. This is a useful perspective for understanding how social practice and technology influence each other, but does not immediately lend itself towards understanding how individuals, for example a student in an engineering class, make sense of these interactions and how those interactions shape their beliefs about the world and their ability to act within it.

With a theoretical framework that focuses on mediated interaction situated in a sociomaterial world, coupled with multi-modal data collection of the entire process, we can gain a better understanding of the critical role computing tools play in shaping students understanding of engineering problem solving. This understanding will aid instructional practices that already embed information systems in learning practices by illuminating aspects of technology use that are critical to effective participation, but otherwise go unmentioned. Additionally, a better understanding of the skills associated with computer mediated problem solving may provide insight into how to help close curricular disconnects that inhibit transfer of techniques and tool skills from one setting to another.⁹

Modeling problem solving

We use Jonassen's definition of a problem as an entity with some initial state and a goal state, and in which there is "some social, cultural, or intellectual value" in arriving at the goal state.¹⁰ Problem *solving* then, is the *process* through which the goal state is achieved. Our understanding and modeling of the problem solving process has developed in tandem with changes in learning theories. Several decades ago, information-processing models cast problem solving as an input/output system.⁵ This modeling suggested that to reach the goal state, the inputs must be transformed through a series of operations to produce an output. This suggests that solutions to problems might be implemented as deterministic algorithms. While this rather narrow view of problem solving led some cognitive scientists to view the study of problem solving as a fairly narrow domain, similar to mathematical thinking,¹¹ Newell, recognizing problem solving as symbolic manipulation, argued that in fact, all of human cognition is problem solving.¹² The boundary between what people think of as "problem solving" versus just "regular" cognition is one of degree, not of kind.¹¹

Recognizing the importance of problem solving in engineering work, Jonassen dedicated much of his work to understanding and categorizing problem solving. His taxonomy of problem structure defines 11 types ranging from *puzzles* to *dilemmas* where the former are well-structured with a well-defined goal state, while the latter are ill-structured with no clearly defined goal state. The problems students practice in school tend to be well-structured, while those in the engineering workplace tend to be ill-structured.¹³ While it was long assumed that skills learned in the practice of solving well-structured problems in school would transfer to the ill-structured problems in the workplace, this is in fact not the case.¹³

Recent studies of problem solving tell us that students' epistemological beliefs^{7,14} and problem representation¹⁵ determine problem solving strategy, and thus play a significant role in students' skill at solving engineering problems. Given that technology plays a role in shaping epistemological beliefs,¹⁶ and computer technology is a central tool for problem representation in engineering environments,¹⁷ we can not study the problem solving process without also studying the role technology plays in it.

Conventional Methods for studying Problem Solving

Methods for studying the problem solving process have relied heavily on the use of think-aloud protocols, sometimes referred to as verbal protocol analysis.^{18,19,20} The basic principle of a think-aloud protocol is that the participant verbalizes their thought process as a means of making their cognitive process apparent to the researcher.²⁰ Think-aloud protocols may be either concurrent, when participants are asked to verbalize during an activity, or retrospective, when participants are asked to recall an activity shortly after completing it.¹⁸

There are challenges with both concurrent and retrospective methods: concurrent verbalization may interfere with the thought process itself, especially for participants who are not used to verbalizing their thought process, while retrospective methods may miss certain actions and decisions points.²¹ Perhaps unsurprisingly, the two methods produce different kinds of information. Concurrent methods produce information primarily about actions and outcomes, while retrospective methods generate information about beliefs, conditional knowledge, and metacognition.²²

Like early problem solving models, think-aloud protocols were developed using information-processing models of cognition. This had two primary impacts on the use of the method: only information that is present in the participant's short-term memory is accessible, and the protocol must be designed carefully so as not to introduce extraneous inputs that would disrupt the participant's thought process.¹⁹ As a side-effect, conventional think-aloud methods tend to prioritize knowledge and understanding of the researcher, since it is they who design the protocol based on their own understanding of the situation Koro-Ljungberg et al. [23].

Bringing a constructivist perspective to think-aloud methods, Koro-Ljungberg et al. suggest a modified protocol that mitigates some of the shortcomings of traditional methods. Their modified protocol combines a traditional concurrent protocol with follow-up interviews. Using this combination, they were able to encourage participants to generate situated knowledge related to the problem solving task. Remaining consistent with the constructivist perspective while acknowledging the role of the researcher, their knowledge is co-constructed between participant and researcher.²³

Thinking with technology

Often overlooked in problem solving studies is the possible impact of the tools used during the process on the process itself. From a cognitive load perspective, computing tools are ostensibly designed to reduce extraneous load by automating routine tasks,²⁴ but this is assuming the use of the tool itself has become routine. In fact, there is evidence that the cognitive load associated with working with unfamiliar tools can interfere with the problem solving process as a whole,^{16,25,26} so an outcomes-focused method of studying problem solving would have difficulty in determining if a failure to produce a solution was a result of misconception, or simply lack of familiarity with the tools. Process-focused methods would fair better in this regard, though without a suitable theoretical framework and data collection strategy, unraveling the impact of tools in the problem solving process may still be difficult. For example, using only a concurrent think-aloud protocol it

is possible the subject may not verbalize thoughts related to working with the tool unless explicitly prompted to do so.

While the distributed cognition approach described by Hollan, Hutchins, and Kirsh [31] suggests that tools can act in a way that offloads cognitive activity, they suggest this results when the interface itself becomes transparent to the user, or what they describe as a direct manipulation interface.³² One example they use to demonstrate this idea is of the interaction between pilot and the autopilot in the cockpit of a Boeing 747. While distributed cognition provides a useful way to understand how the instruments together with the pilot achieve “the successful completion of a flight”, this presupposes the pilot is already familiar with the interface. One might imagine that a student new to flying would not see the instrumentation as a transparent aid in achieving the goal of successful flight, but rather as a daunting set of dials, knobs, and buttons to learn. New flight students, after all, do not start their studies in a Boeing 747.

In a study by McNeill et al., students identified the Internet and software as “resources that could be consulted to help find information”.⁷ This suggests that understanding *how* students interact with these computer-based resources is important for understanding the entire process. In fact, given that computing technology shapes both practice²⁷ and thinking²⁸ we know that these digital resources are more than just a knowledge repository. Computer use interacts with the problem solving process in at least three ways:

1. By shaping epistemic beliefs and emotion - Jonassen argues that both epistemic belief, and mood affect problem solving strategies and process.¹⁰ For example, the belief that there is a right or wrong way to solve a problem may lead some students to spend a lot of time searching for the “right” way, and a belief that one either has or does not have knowledge necessary to solve a problem may lead some solvers to give up at the first sign of difficulty. We also know that working with computer technology can shape epistemic belief,¹⁶ and emotional state, as when someone experiences frustration when they do not know how to use a particular interface.²⁹
2. As a source of extraneous cognitive load - Studies exploring the difference between expert and novice problem solvers indicate that experts switch between problem-solving stages more frequently than novices.³⁰ If problem-solvers are using a computer as their primary tool, then this switching will likely entail a parallel switching of software and other electronic resources accessible from the computer. Proficiency of computer interaction then, may be a contributing factor to problem-solving success. The user interface itself may be a source of cognitive load too, for example, a study of different programming environments found that some environments are associated with more cognitive load than others.²⁶
3. As a conduit to the social world - We have indicated that the constructivist viewpoint has strongly influenced both theoretical frameworks and methods for understanding the problem-solving process.^{6,23} Under constructivism, the individual actively constructs knowledge and understanding through interacting with the world.^{33,34} Thus, as an interactive tool, the computer will necessarily play a significant role in how students construct knowledge.

These examples demonstrate that interacting with computer technology affects a number of factors that are also associated with problem-solving. This suggests that to effectively study how

a student engages with the problem-solving process, we need not only a theoretical framework of problem-solving, but also of computer technology-in-use.

The Subjectivity of the human-machine interface

Imagine sitting at your computer writing a conference paper. If you are an experienced typist you are likely not consciously thinking of the keyboard, the individual keys, or the circuitry and software in the computer that allows for mechanical key presses to result in the change of pixel colors that you recognize as letters on the screen. It is *as if* your words flow freely from your mind onto the representation of the page. In that moment, it is fair to say the perceived boundary between you and the material representation of your words is blurry at best. However, if something unexpected occurs, you hit a wrong letter, your word processor crashes, your cat jumps on the keyboard, the boundary between you and the machine snaps into sharp relief. These changes in perceived boundary between us and the material world could provide key sites of inquiry for the purposes of studying problem solving research. As this example demonstrates, interaction with computer technology can not simply be split into studying what the human is doing plus what the computer is doing: the relationship between the two is in constant negotiation throughout the process. Scholars have suggested a number of theoretical frameworks to make sense of the subjective experience of interacting with the material world.^{35,36,37} Of these, sociomateriality has become somewhat common in both engineering education³⁸ and organizational research.³⁹

The fundamental principal of sociomateriality is that the human subject can not be cleanly separated from the material world for purposes of analysis.³⁸ In this context, “material” means both the physical “stuff” that we may interact with, for example the computer keyboard, but also non-physical artifacts like software, and in particular the qualities of those artifacts that persist across space and time.⁴⁰ The researcher must take care not to give primacy to either the human subject, or material world. The unit of analysis then becomes not one or the other, but the relationship between the two.⁴¹

Existing work that makes use of the sociomaterial framework seeks to describe sociomaterial systems and the relationships that define them. As educators, our primary goal is to help students learn. While we acknowledge that learning takes place within a sociomaterial context, it will take more than just a systems-level understanding to inform pedagogical and curricular decisions. We must understand how students’ epistemic beliefs and prior knowledge interact with the world, the process through which they make sense of the world, and how that influences certain decisions, such as problem solving strategies, or deciding to give up on a problem. Thus, while we still must consider the system as a whole, our goal is not to simply describe the social and material interactions, but to describe the cognitive sociomaterial interactions. We visualize this cognitive sociomaterial system in Figure 1. Our data collection and inquiry will be centered around the subject (head silhouette) and their cognition (gears), but we recognize this process to be situated in a world with social and material components, in which they interact. The sociomaterial world shapes our students’ cognition (red arrows directed at subject), and then as part of their cognitive process, they act in the world, reflecting their understanding back onto it (blue arrows).

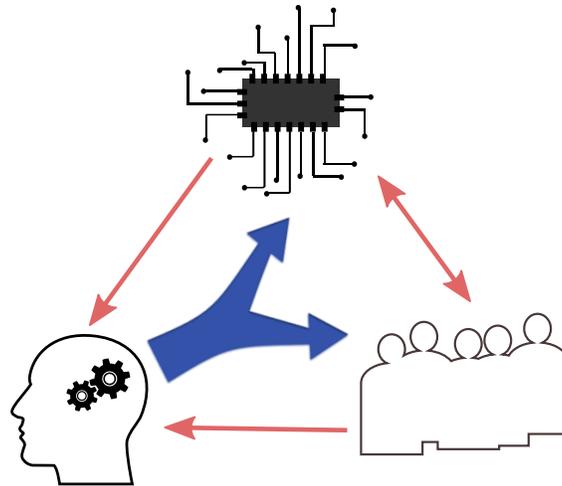


Figure 1: Situated cognition in a sociomaterial world. The subject, indicated by the head silhouette, is situated within and interacts with social and material agents within the world. The social and material agents are intertwined such that they must be studied as a system, rather than individually. As one of those social agents, the subject thinks, indicated by gears, about the world they are in, and the problem they wish to solve. This cognitive process, which is shaped by the influences of the world, shapes their action in the world, indicated by the blue arrows. We desire a method that will help illuminate this cognitive process of experiencing and action in the world.

Discourse and materiality

The focus on relationships and interactions provided by a sociomaterial view of the world naturally suggests an idea of discourse, which in the Foucaultian sense means the use of both language and practice to produce knowledge.⁴² Gupta and Elby note that novice students tend to view mathematical knowledge and concrete ideas as two disjoint types of knowledge that do not “speak” to one another.⁴³ Thus the concept of a discourse between different knowledge domains as well as between human and computer is useful for understanding problem solving.

O’Connor et al. suggest a mediated discourse analysis to explore the process of becoming an engineer.⁴⁴ In their study they analyze student presentations in a project based class. Since project presentations are a common practice in many engineering curricula, they seek to understand how the practice plays a role in reproducing a certain aspect of engineering culture, the belief that engineering design is isolated from the social and political realm.⁴⁵ In their study the mediating objects included presentation technology such as the projector and desktop computer along with software such as PowerPoint and SolidWorks. The mediational means will depend on the study, though because we are interested in how students engage with computing tools during problem solving, generally the computer along with any software or Internet resource used during problem solving will become a mediational means, though there may be others.

Influenced by critical discourse analysis,⁴⁶ mediated discourse analysis encourages the research to view the discourse within a larger social and political world. It is not enough to simply analyze the subject’s words and actions, to understand the discourse the researcher must seek to understand the sociomaterial process that shape discourse. Thus, if a participant is observed to access an

online repository to aid in problem solving this is not a simple act of information retrieval. Rather the researcher must consider what beliefs the participant has about socially constructed knowledge to know to access the resource in the first place, what social practices were necessary to create that resource, how is knowledge in that resource cataloged and validated, etc.

Our Multi-modal method for studying problem solving

Unlike purely cognitive models of problem solving,¹¹ we conceive of problem solving as process that unfolds through interaction of cognitive skills and situated practice.²⁷ Our research framework and method should allow us to interrogate the problem solving process itself, and bring to light interactions between problem solver and tools that may warrant further investigation. In particular, we are interested in learning how the problem solver themselves makes meaning out of their interactions, and how that shapes future interactions, perhaps in different but similar contexts.

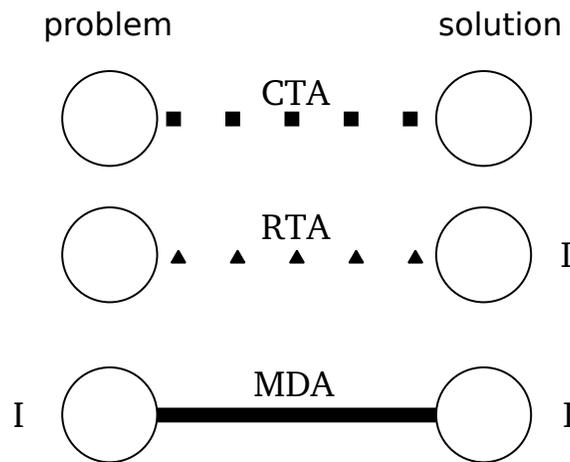


Figure 2: Three study designs for problem solving research. Dotted line indicates information about specific points in time provided by a concurrent think-aloud protocol (CTA) and retrospective think aloud (RTA), though each generates different kinds of information. The solid line indicates collecting multiple continuous data streams which can be used with mediated discourse analysis. Interviews are indicated by a capital “I” where interviews after the problem solving session may be structured as retrospective think-alouds.

We suggest a research design informed by constructivist and sociomaterial theoretical perspectives. In this combination we take knowledge to be actively constructed by the subject through inner dialog as well as interaction with the material world (Figure 1). The subjectivity of the boundary between subject and material world serves as a guide for inquiry: we are interested in what actions and behaviors as well as cognitive states are associated with apparent movement of this boundary. This may be extracted from data in different ways. For example, perhaps sudden frustration seen in facial expressions, or moving of hands off the keyboard indicates a shift in the boundary as experienced by the participant. These may serve as time points to ask about in follow up interviews, or lacking time to analyze screen-capture data before hand, the followup interviews

themselves may be structured to ask about times during the process the participant was frustrated or something unexpected threw them off.

Because different data streams provide different kinds of information, we must utilize multiple streams in each study if we want to be able to reconstruct the interactions between participant's cognitive behavior and interaction with the problem space.

Since concurrent think-aloud protocols generate data predominantly about action and consequence,²² we instead propose to extract this information through analysis of screen-capture video, thus eliminating some of the concerns associated with concurrent think-aloud protocols. By viewing the relationship between subject and computer as a discourse, we can view screen-capture video and watch for actions initiated by the user, and response from the computer. For instance, if we are studying computer programming, we may identify times in the screen-capture data when the programmer initiated a compile, and how the computer responded (e.g. with a message of success, or with an compile error).

Video recording of the participant's face and hand placement can provide information regarding affective state, e.g. frustration or surprise. Times during which the participant appears to have a change in emotional state may be checked with screen-capture times to observe what interaction was taking place with the computer at that time. A follow up interview conducted as a retrospective think-aloud protocol for key moments (e.g. "how did you get started?", "walk me through what happened that lead up to the point you become frustrated?") provides additional context for these critical interaction moments. Likewise, the follow-up interview can provide information about critical moments that might have otherwise gone un-noticed in either video stream, for example if a participant mentions getting stuck on a particular aspect of the process.

We compare our method against other common methods in Figure 2. During concurrent think aloud protocol we can gather information about specific points in time, those which the participant verbalizes, as indicated by the dotted line between problem and solution. Retrospective think-alouds (RTA) also generate information about specific points in time, but a different kind of information than CTA. This difference is indicated by different shapes at the sample points. In contrast to both CTA and RTA, by recording screen-capture and video data of the participant, we have multiple continuous streams of data, indicated by the solid line, that may be analyzed as part of a mediated discourse analysis. Interviews are indicated by the letter "I". An intake interview provides contextual information about each participant's typical strategies, believed proficiency as a problem solver, etc. The video streams of computer screen and participant face provide information about interaction and emotional state, while a follow up interview as part of the retroactive protocol, allows the researcher to ask the participant to elaborate their thought process at critical decision points. All together, these data allow the researcher to reconstruct a participant's problem solving process and explore how their thought process was shaped by the environment and was subsequently transformed into actions within the environment.

Problem solving session

In any research study it is important to define the unit of analysis. Most situated engineering work transcends convenient delineations of time or space and as researchers we must implicitly if not explicitly define bounds on what we study. We define a problem solving session as a continuous block of time during which someone is engaged in a problem solving activity. This construction mirrors Bhatt and Roock's (2014) term *digital literacy event* as a self-contained unit of analysis. Unfortunately, by their nature, many real-world problems take multiple sessions and the researcher may only have access to one. However, the larger scale the problem, the more likely it can be conveniently decomposed into smaller sub-problems. For example, a programming project may span days or weeks, but within each session the programmer may have specific sub-goals (e.g. add a feature) and engage in multiple sub-problems (e.g. debugging, feature testing). By taking a discourse perspective, we can view the relationship between subject and problem as a negotiation between multiple sub-problems, each which may take the focus of the participant at different times, e.g. while implementing a new feature, the participant may notice a bug and engage in a debugging process before returning back to feature implementation.

Example case

In this section we describe an ongoing study that is utilizing these methods. While this study is conducted in a laboratory setting which restricted participants' range of options, it provides a convenient example of how the theoretical framework of sociomateriality might be combined with mediated discourse analysis to study problem solving with a focus on student meaning-making.

In this study we seek to understand the variety of strategies and skills experienced programmers rely on while working on a real programming project. We leverage the sociomaterial perspective to try bring to the surface interactions or assumptions that might otherwise go unnoticed. An intake interview is designed to provide information about the participant's epistemic beliefs about programming, for example during the intake interview, one participant said that the programming he was asked to do in his introductory programming course was not "real" programming because he considered programming to be problem solving and "if they give you the solution to the problem, that's not problem solving, then I would argue that's not really programming either." This helps us interpret interactions between the participant and the computer while he is programming.

We collect screen-capture video of the participant's computer screen and video of their face while they work. This provides a continuous record of interactions throughout the process. Based on information gained in the intake and outtake interviews, we may look for certain features in the video recordings. For example, in an outtake interview one participant mentioned he was stuck for a period of time trying to decide what data type and size to make a particular array. We can use this information to find times within the video data when he changed the parameters of the array and explore interactions immediately before and after to try and understand how that he understood that particular array in the context of the larger problem.

Outtake interviews, like we mentioned, provide both a means to collect retrospective think-aloud data about certain decision points, for example “why did you decide to change the array size?”, as well as help identify critical points in the video stream to explore in more detail, e.g. “describe a time that you were stuck.” We can then look for those times in the video stream and explore interactions leading up to the point where the participant indicated getting stuck.

While ultimately we hope that this methodology proves useful for understanding problem solving practice situated in “real” environments such as the classroom or workplace, the current state of this work applies to recreating problem-solving experiences in a lab environment. This is a result of the study being part of a larger study to explore the utility of functional near-infrared spectroscopy (fNIRS) brain imaging in engineering education research. This involves fitting an fNIRS cap onto the participant, connected to a machine. While this setup allows full range of motion for tasks such as programming at a computer, it does limit the location of the study to a particular lab on campus.

Towards a holistic understanding of programming skill

The aim of this ongoing study is to better understand the range of skills and strategies people use when programming in realistic environments. Motivation for this study came from the informal observation that during group work in introductory engineering courses it was generally the person with prior programming experience who completed programming tasks, despite the tasks ostensibly being designed to require only those skills taught in class. To be eligible for participation, subjects were required to use their own laptop and have their own project to work on. The rationale for these restrictions were that we wanted to as much as possible recreate a realistic experience for the subjects despite them coming into a lab environment for the study. Additionally, asking participants to come with their own projects to work on alleviated certain challenges seen in other problem-solving studies. For example, Douglas et al. [6] note that some participants in their study felt researchers expected them to complete the problems they were given.⁶ By bringing their own projects, participants in our study brought their own expectations as well, which we asked them about in the intake interview.

Preliminary analysis has revealed interesting events that warrant further exploration. Though participants were not instructed to verbalize their thought process, some said they naturally talked to themselves while working. One participant explicitly verbalized “this should work” when initiating a compile and then “no, no it doesn’t why?” when the compiler responded with an error message. This marks an event to explore further. By reviewing at the screen-capture data leading up to the surprise result to reconstruct what the participant thought would happen. Additionally, most participants make heavy use of the online crowd-sourced site StackOverflow for solving compile errors. A quick literature search revealed that making sense of these search results and applying them to solve a particular specific error is a complex task.⁴⁸ Thus, that our experienced programmers were doing this with apparent ease indicates a certain level of practice with this process. In subsequent studies we may use this information to inform the intake or outtake interviews, such as by specifically asking how a participant how they use a sociomaterial tool such as StackOverflow in their typical problem solving process.

Discussion and limitations

Admittedly, this method involves more work than conventional think-aloud protocols. While other problem-solving studies have utilized video recording to aid in retrospective think aloud, we are advocating for a more in-depth analysis of the video recordings *in addition to* retrospective think-aloud. We believe that the trade off is worth it in certain problem solving studies. In particular, when problem solving is done with the aid of technology that interacts with the problem and subject in complex ways. Computer hardware and software does this by shaping epistemic beliefs, contributing to cognitive load, and acting as a conduit to the social world. If we hope to understand how a student's epistemic beliefs impact their strategies, how extraneous cognitive load changes impacts their process, and how they access and utilize social resources during the problem solving process, we must explicitly include these steps in our analysis.

Conclusion

In this paper we have outlined how key factors in problem solving: epistemic belief, cognitive load associated with tool use, and accessing socially constructed knowledge, are also heavily mediated by computer use in today's engineering classrooms. For these reasons, we believe that any study of problem solving behavior in the context of computer use must be informed by a suitable framework for understanding the complex relationship between human cognition, behavior, and interaction with the material world. Additionally, appropriate data must be collected and analysis used such that the effects of computer use may be distinguished from other factors, such as conceptual understanding. We think leveraging a sociomaterial framework that has already proved useful in engineering education research is a good first step, however, we must push the framework a bit to center inquiry around student cognition and meaning making so that we are better poised to use results to inform pedagogical and curricular decisions.

Likewise, we argue that multi-modal data must be collected to allow the researcher to reconstruct the interaction between subject, problem, and computer during a problem solving session. Intake interviews can provide information on epistemic beliefs and give researchers a sense of the kinds of problem solving strategies a participant is aware of. This can inform analysis of other data collected from the same participant. Video data of the computer screen and the participant's face provide information about the interaction between participant and computer, as well as emotional state of the participant through the process. This allows for a mediated discourse analysis of participant interacting with aspects of the problem via the computer. Finally, outtake interviews serve as both retrospective think-aloud protocols to gather information about metacognition and decision points, as well as suggesting interesting times to study more closely in the video data. Utilizing all three will help researchers reconstruct the problem solving process of the participant and be better poised to identify specific sites of interest that may lead to further study, such as the complex process involved in making sense of social knowledge repositories in the context of specific problems.

We hope building on this framework and method will help educators identify aspects of computer use as part of problem solving activities that may need special instructional attention, as well as

help students better manage cognitive load (e.g. by isolating certain computer skills and encouraging practice until they become automated). This will become increasingly necessary as complex computer tools continue to be integrated into engineering practice and subsequently included in the engineering curriculum. Rather than present the tools to students as objective “tools of the trade”, we must help them make sense of them, and learn to leverage them in their own problem solving process.

References

- [1] Krishna Madhavan and Euan D. Lindsay. “Use of Information Technology in Engineering Education”. In: *Cambridge Handbook of Engineering Education Research*. New York, NY: Cambridge University Press, 2014, pp. 633–654.
- [2] David Bowman and Lisa Benson. “Museink: Seeing and Hearing a Freshman Engineering Student Ink and Think”. In: *117th Annual Conference and Exposition of the American Society for Engineering Education (ASEE 2010)*. June 20, 2010.
- [3] David Jonassen. “Transforming Learning with Technology: Beyond Modernism and Post-Modernism, or Whoever Controls the Technology Creates the Reality”. In: *The Nature of Technology: Implications for Learning and Teaching*. Ed. by Michael P. Clough, Joanne K. Olson, and Dale S. Niederhauser. Springer Science & Business Media, 2013, pp. 101–110.
- [4] Aditya Johri and Barbara M Olds. “Situated Engineering Learning: Bridging Engineering Education Research and the Learning Sciences”. In: *Journal of Engineering Education* 100.1 (2011), pp. 151–185.
- [5] Herbert A. Simon and Allen Newell. “Human Problem Solving: The State of the Theory in 1970.” In: *American Psychologist* 26.2 (1971), p. 145.
- [6] Elliot P. Douglas et al. “Moving beyond Formulas and Fixations: Solving Open-Ended Engineering Problems”. In: *European Journal of Engineering Education* 37.6 (Dec. 1, 2012), pp. 627–651.
- [7] Nathan J. McNeill et al. “Undergraduate Students’ Beliefs about Engineering Problem Solving”. In: *Journal of Engineering Education* 105.4 (Oct. 1, 2016), pp. 560–584.
- [8] Suzie Wong Scollon. *Nexus Analysis: Discourse and the Emerging Internet*. Routledge, 2004.
- [9] Ann F. McKenna. “Adaptive Expertise and Knowledge Fluency in Design and Innovation”. In: *Cambridge Handbook of Engineering Education Research*. Cambridge University Press, 2015.
- [10] David H. Jonassen. “Toward a Design Theory of Problem Solving”. In: *Educational Technology Research and Development* 48.4 (Dec. 2000), pp. 63–85.
- [11] John R. Anderson. “Problem Solving and Learning”. In: *American Psychologist* 48.1 (1993), pp. 35–44.
- [12] Allen Newell. “Reasoning, Problem Solving and Decision Processes: The Problem Space as a Fundamental Category”. In: (1979).
- [13] David H. Jonassen. “Engineers as Problem Solvers”. In: *Cambridge Handbook of Engineering Education Research*. New York, NY: Cambridge University Press, 2014.
- [14] Gregory Schraw, Michael E. Dunkle, and Lisa D. Bendixen. “Cognitive Processes in Well-Defined and Ill-Defined Problem Solving”. In: *Applied Cognitive Psychology* 9.6 (Dec. 1995), pp. 523–538.
- [15] Aditya Johri, Wolff-Michael Roth, and Barbara M. Olds. “The Role of Representations in Engineering Practices: Taking a Turn towards Inscriptions”. In: *Journal of Engineering Education* 102.1 (Jan. 1, 2013), pp. 2–19.
- [16] Michael P. Clough, Joanne K. Olson, and Dale S. Niederhauser. *The Nature of Technology: Implications for Learning and Teaching*. Springer Science & Business Media, 2013.
- [17] D. Jonassen, J. Strobel, and CB Lee. “Everyday Problem Solving in Engineering: Lessons for Engineering Educators”. In: *Journal of Engineering Education* 95.2 (2006), pp. 139–151.
- [18] K. Anders Ericsson and Herbert A Simon. “Verbal Reports as Data”. In: *Psychological Review* 87.3 (May 1980), pp. 215–251.
- [19] K. Anders Ericsson and Herbert A. Simon. *Protocol Analysis: Verbal Reports as Data*. Rev. ed. Cambridge, Mass: MIT Press, 1993.

- [20] E Krahmer and N Ummelen. "Thinking about Thinking Aloud: A Comparison of Two Verbal Protocols for Usability Testing". In: *IEEE Transactions on Professional Communication* 47.2 (2004), pp. 105–117.
- [21] Tamara van Gog et al. "Uncovering the Problem-Solving Process: Cued Retrospective Reporting Versus Concurrent and Retrospective Reporting." In: *Journal of Experimental Psychology: Applied* 11.4 (Dec. 2005), p. 237.
- [22] K. Lynn Taylor and Jean-Paul Dionne. "Accessing Problem-Solving Strategy Knowledge: The Complementary Use of Concurrent Verbal Protocols and Retrospective Debriefing." In: *Journal of Educational Psychology*. American Educational Research Association Annual Meeting., Apr, 1994, New Orleans, LA, US; A draft of this article was presented at the aforementioned meeting. Vol. 92. American Psychological Association, Sept. 2000, p. 413.
- [23] Mirka Koro-Ljungberg et al. "Reconceptualizing and Decentering Think-Aloud Methodology in Qualitative Research". In: *Qualitative Research* 13.6 (2012), pp. 735–753.
- [24] Fred Paas, Alexander Renkl, and John Sweller. "Cognitive Load Theory and Instructional Design: Recent Developments". In: *Educational psychologist* 38.1 (2003), pp. 1–4.
- [25] Slava Kalyuga. "Enhancing Instructional Efficiency of Interactive E-Learning Environments: A Cognitive Load Perspective". In: *Educational Psychology Review* 19.3 (Aug. 24, 2007), pp. 387–399.
- [26] Murat Pasa Uysal. "Evaluation of Learning Environments for Object-Oriented Programming: Measuring Cognitive Load with a Novel Measurement Technique". In: *Interactive Learning Environments* (May 18, 2015), pp. 1–20.
- [27] Sylvia Scribner and Ethel Tobach. *Mind and Social Practice: Selected Writings of Sylvia Scribner*. Google-Books-ID: ppTiqXHfhAYC. Cambridge University Press, Jan. 13, 1997. 408 pp.
- [28] Patricia A. Tun and Margie E. Lachman. "The Association between Computer Use and Cognition across Adulthood: Use It so You Won't Lose It?" In: *Psychology and Aging* 25.3 (2010), pp. 560–568.
- [29] Robin H. Kay and Sharon Loverock. "Assessing Emotions Related to Learning New Software: The Computer Emotion Scale". In: *Computers in Human Behavior*. Including the Special Issue: Integration of Human Factors in Networked Computing 24.4 (July 2008), pp. 1605–1623.
- [30] Robin S. Adams, Jennifer Turns, and Cynthia J. Atman. "Educating Effective Engineering Designers: The Role of Reflective Practice". In: *Design Studies* 24.3 (2003), pp. 275–294.
- [31] James Hollan, Edwin Hutchins, and David Kirsh. "Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research". In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 7.2 (2000), pp. 174–196.
- [32] Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. "Direct Manipulation Interfaces". In: *Human-Computer Interaction* 1.4 (Dec. 1, 1985), pp. 311–338.
- [33] Peter L. Berger and Thomas Luckmann. *The Social Construction of Reality*. New York: Anchor Books, 1966. 219 pp.
- [34] Mordechai Gordon. "Toward A Pragmatic Discourse of Constructivism: Reflections on Lessons from Practice". In: *Educational Studies* 45.1 (Feb. 3, 2009), pp. 39–58.
- [35] Karen Barad. "Posthumanist Performativity: Toward an Understanding of How Matter Comes to Matter". In: *Signs* 28.3 (2003), pp. 801–831.
- [36] Wanda J. Orlikowski. "Sociomaterial Practices: Exploring Technology at Work". In: *Organization Studies* 28.9 (Sept. 1, 2007), pp. 1435–1448.
- [37] Gilles Deleuze and Félix Guattari. *A Thousand Plateaus: Capitalism and Schizophrenia*. Bloomsbury Publishing, 1988.
- [38] Aditya Johri. "The Socio-Materiality of Learning Practices and Implications for the Field of Learning Technology". In: *Research in Learning Technology* 19.3 (Sept. 2011), pp. 207–217.
- [39] Dubravka Cecez-Kecmanovic et al. "The Sociomateriality of Information Systems: Current Status, Future Directions". In: *Mis Quarterly* 38.3 (2014), pp. 809–830.
- [40] Paul M. Leonardi. "Materiality, Sociomateriality, and Socio-Technical Systems: What Do These Terms Mean? How Are They Related? Do We Need Them?" In: *Materiality and Organizing: Social Interaction in a Technological World*. Rochester, NY: Social Science Research Network, 2012.
- [41] Paul M. Leonardi. "Theoretical Foundations for the Study of Sociomateriality". In: *Information and Organization* 23.2 (Apr. 2013), pp. 59–76.
- [42] Stuart Hall. "Foucault: Power, Knowledge and Discourse". In: *Discourse theory and practice: A reader* 72 (2001), p. 81.

- [43] Ayush Gupta and Andy Elby. “Beyond Deficit-Based Models of Learners’ Cognition: Interpreting Engineering Students’ Difficulties with Sense-Making in Terms of Fine-Grained Epistemological and Conceptual Dynamics”. In: (2010).
- [44] Kevin O’Connor et al. “Ideologies of Depoliticization in Engineering Education: A Mediated Discourse Analysis of Student Presentations in a First-Year Projects Course”. In: ASEE Conferences, 2015, pp. 26.880.1–26.880.17.
- [45] Donna Riley, Amy E. Slaton, and Alice L. Pawley. “Social Justice and Inclusion”. In: *Cambridge Handbook of Engineering Education Research*. New York, NY: Cambridge University Press, 2014, pp. 335–356.
- [46] Jan Blommaert and Chris Bulcaen. “Critical Discourse Analysis”. In: *Annual Review of Anthropology* 29 (2000), pp. 447–466.
- [47] Ibrar Bhatt and Roberto de Roock. “Capturing the Sociomateriality of Digital Literacy Events”. In: *Research in Learning Technology* 21 (Jan. 2014).
- [48] Mohammad Masudur Rahman and Chanchal K. Roy. “Recommending Relevant Sections from a Webpage about Programming Errors and Exceptions”. In: *Proceedings of the 25th Annual International Conference on Computer Science and Software Engineering*. IBM Corp., 2015, pp. 181–190.