Augmented Silkscreen: Designing AR Interactions for Debugging Printed Circuit Boards

Ishan Chatterjee
ichat@cs.washington.edu
Paul G. Allen School of Computer Science & Engineering, University of Washington
Seattle, WA

Olga Khvan
khvanolga.v@gmail.com
Human-Computer Interaction + Design, University of Washington
Seattle, WA

Tadeusz Pforte
pfortet@cs.washington.edu
Paul G. Allen School of Computer Science & Engineering, University of Washington
Seattle, WA

Richard Li
lichard@cs.washington.edu
Paul G. Allen School of Computer Science & Engineering, University of Washington
Seattle, WA

Shwetak Patel
shwetak@cs.washington.edu
Paul G. Allen School of Computer Science & Engineering, University of Washington
Seattle, WA

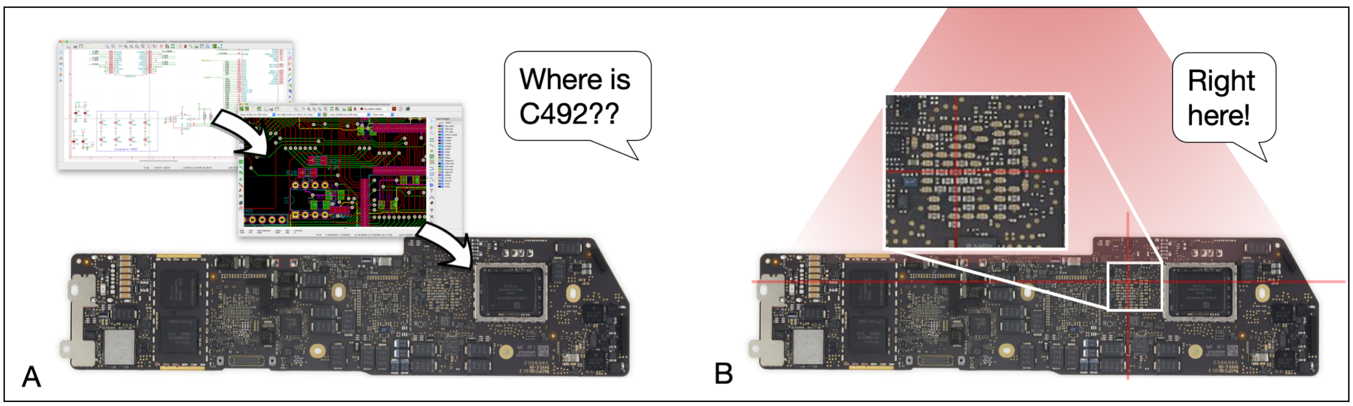

Figure 1: (A) PCBs, such as the motherboard above, can contain thousands of components, most smaller than a grain of rice. Among other metadata, each component has a location, orientation, reference designator, and set of pins. Pins are connected via metallic traces buried in the board called nets. Basic tasks involved in debugging PCBs (such as finding a given component, pin, or net) typically involve flipping through multiple software files, such as the schematic ((A), top) and layout ((A), middle). (B) Augmented Silkscreen explores augmented reality interaction techniques to make this and other PCB debugging workflows more seamless and efficient.

ABSTRACT

Debugging printed circuit boards (PCBs) requires frequent context switching and spatial pattern matching between software design files and physical boards. To reduce this overhead, we conduct a series of interviews with electrical engineers to understand their workflows, around which we design a set of AR interaction techniques, we call Augmented Silkscreen, to streamline identification, localization, annotation, and measurement tasks. We then run a set of remote user studies with illustrative video sketches and simulated PCB tasks to compare our interactions with current practices, finding that our techniques reduce completion times. Based on these quantitative results, as well as qualitative feedback from our participants, we offer design recommendations for the implementation of these interactions on a future, deployable AR system.

DIS '21, June 28-July 2, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8476-6/21/06…$15.00
https://doi.org/10.1145/3461778.3462091

CCS CONCEPTS

• **Human-centered computing → Mixed / augmented reality**; • **Hardware → Printed circuit boards**; *Board- and system-level test.*

KEYWORDS

Augmented Reality, Printed Circuit Board, Hardware, Debugging

ACM Reference Format:
Ishan Chatterjee, Olga Khvan, Tadeusz Pforte, Richard Li, and Shwetak Patel. 2021. Augmented Silkscreen: Designing AR Interactions for Debugging Printed Circuit Boards. In *Designing Interactive Systems Conference 2021 (DIS '21), June 28-July 2, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3461778.3462091

1 INTRODUCTION

By 2030, the number of smart devices in the world is projected to reach 50 billion [20]. The proliferation of these devices can largely be attributed to the increasingly integrated nature of electronics and silicon, as per Moore's law. Just as the number of transistors in an integrated circuit (IC) have increased exponentially, so too have printed circuit boards (PCBs) become increasingly dense with electronic components. These denser and more complex PCBs pose greater challenges for electrical engineers during the debugging process. However, the tools used to support these engineers in debugging faulty PCBs during design and development remain largely unchanged. During the process of debugging a new PCB design, electrical engineers must constantly move between circuit diagrams, board layout diagrams, and the physical circuit board itself in order to validate their design or understand the nature of a design failure. The design and the layout might be distributed across both physical (i.e. paper) and virtual (i.e. software tool) mediums as well. The constant context switching, as well as manually looking for corresponding components across the different representations of the circuit, lends this process to be extremely time-consuming and error-prone, such that the smallest optimizations to this process can have significant compounding benefits.

Augmented reality (AR) has been cited as an effective paradigm for reducing the overhead of tasks with repeated context switching, particularly those with spatial associations and affordances [10, 12, 16, 21, 26]. While there has been some amount of prior work exploring ways of using AR for debugging breadboards and PCBs, the primary focus has been on supporting hobbyist makers and students, in particular taking an educational perspective [15, 19, 24, 29]. In this work, we conduct a design exploration, investigating how this paradigm can be effectively applied to support the existing PCB debugging workflows of industry professionals through a series of needs finding and evaluative user studies. We design a set of AR interactions to enhance workflow productivity, and evaluate their utility via feedback interviews, illustrative video sketches, and remote simulation of certain PCB tasks. The scope of our work does not include the complete implementation of an AR tool, instead focusing our exploration on understanding user needs and designing AR interactions agnostic to AR implementation (head mounted device, projective AR, camera pass-through AR, etc.).

The goal of this work is to highlight and demonstrate to the HCI community the design considerations and research opportunities in this space.

The main contributions of this work include:

(1) An initial, formative study identifying challenges in PCB assembly, bring-up, and debugging workflows to inform interaction design.
(2) A set of augmented reality interaction techniques supporting workflows related to localization, annotation, and measurement operations of components, pins, and nets across the design files and the physical PCB.
(3) A user feedback evaluation (n=6) for:
  (a) assessing the interaction techniques for user preference, usage, and likelihood of adoption, and
  (b) evaluating completion time and user confidence in component identification tasks.

2 TECHNICAL BACKGROUND

In this section, we will briefly describe how PCBs are designed, define terminology, and discuss the tools used to design PCBs. Please see the supplementary material and online tutorials[1,2] for visual descriptions.

The electrical engineering design process typically starts with designing a circuit to meet a set of functional requirements. Using an electrical computer-aided design (ECAD) tool, the logic of the circuit is formalized via schematic capture into a *schematic diagram*, which visualizes the circuit's components as symbols and the circuit's interconnections (*nets*) as topological lines between the components' pins. This logic is then transferred to a *layout diagram*, where components and the connections between them are placed in a physical coordinate space. Finally, the design is then physically fabricated and assembled into a functional PCB.

A PCB is a sheet of fiberglass with buried conductive paths to connect components in the same way that wires connect components on a breadboard. A *component* is a circuit element that performs a certain electrical function, such as a resistor, inductor, capacitor, diode, transistor, or integrated circuit (IC). These components have exposed metal *pins* which are terminals soldered to exposed conductive pads on the PCB surface. A typical PCB can have tens, hundreds, or thousands of components, most smaller than a breadcrumb. These pads (and thus the pins of components soldered to them) are interconnected through a dense labyrinth of conductive paths, called *traces*, or *nets*, buried within the PCB surface. PCB contains multiple layers (usually between 2 and 16) each with a maze of traces. Tiny conductive tunnels called *vias* connect traces on different layers of a PCB, including connecting the two sides of the PCB. The exposed pins and pads are typically where an engineer would access an electrical signal via a sharp *probe*. The probe would connect to a test instrument, such as a multimeter, to record a measurement.

PCBs are designed using Electrical Computer-Aided Design (ECAD) tools. Currently popular ECAD toolchains include OrCAD [9], Allegro [1], Altium Designer [2], Zuken CADSTAR [8], EAGLE [3], KiCAD [6], and EasyEDA [4]. As these toolchains are built to support PCB design (rather than debugging) they follow a generally linear process in support of two major functions: schematic capture and layout. Navigation between representations in all tools is supported by text-based search of component *reference designators* and *net names* within the design files. Reference designators are sometimes printed directly on the PCB itself via a *silkscreen*

[1]Sparkfun. PCB Basics. https://learn.sparkfun.com/tutorials/pcb-basics/all
[2]Bryan Siepart, Adafruit. Make your own PCB with Eagle, OSH Park, and Adafruit! https://learn.adafruit.com/making-pcbs-with-oshpark-and-eagle

layer. A handful of ECAD tools, such as Altium [2] and KiCAD [6], support the notion of *cross-probing* a component where, if both the schematic and layout tools are open, a component selection in one will highlight the corresponding component in the other view. However, schematics are often exported to PDFs for portability, and layouts are sometimes viewed in Gerber or ODB++ viewers. These alternative formats and tools inhibit cross-probing and are sometimes not searchable, increasing cross-navigation friction. In this work, we seek to extend the notion of cross-probing to and from the physical PCB.

3 RELATED WORK

In this section, we provide background on augmented reality systems and place our work among the literature related to extending the capabilities of breadboards and PCBs.

3.1 Augmented Reality

Augmented reality (AR) has long been seen as a paradigm that can decrease the barrier between virtual and physical information transfer. This transfer process can consist of two components: the presentation of the information to the user, generally in the form of visualizations; and the ability for the user to interact with the visualization, perhaps enabling the user to query for additional information. Prior work has shown that AR systems presenting spatially-tracked information, even with no interaction component, can be effective not only in reducing error rate and mental effort across industrial tasks such as order picking [23] and object assembly [12, 26], but also as a medium for understanding abstract concepts such as how electrons flow through a circuit [13, 14]. Extending AR experiences to enable interaction makes them even more powerful. Such interactions might enable actions such as the selection of elements in the physical world to be used as part of a virtual tool operation. Digitaldesk [28] demonstrated such interactions with examples such as allowing the user to move a number from a physical price list into a virtual calculator. Our work explores the design space of both of these aspects of AR – visualization and interaction – and how they might enhance how electrical engineers work with PCBs.

3.2 Augmented Reality for PCBs

While PCBs are often considered the staple of industry level electronics, breadboards are often used by students and hobbyists for their solderless reconfigurability that enables rapid iteration, and are rarely used as part of the hardware development process in industry. While recent work in the HCI community aimed at the student population has demonstrated a number of breadboard augmentation techniques [15, 19, 22, 29], PCBs are substantially more intricate, requiring much more careful augmentation. We motivate our work as a means toward this end. In this section, we discuss more directly comparable related work in the realm of specifically augmenting PCBs.

3.2.1 Visualization Tools. A few tools support visualizing certain component metadata, such as location, directly on the PCB. InspectAR [5] is a recently released tool that uses mobile AR to overlay elements of the layout and associated metadata onto a camera view of the PCB displayed on a mobile tablet or PC. It is targeted toward supporting industry professionals, with couplings to industry standard ECAD tools. The tool does not seem to support direct interaction with the PCB itself, measurement interactions, or a topological schematic view. The sales webpage offers strong testimonials speaking to the increased assembly and debugging efficiency from decreased context-switching, claiming "an average 30% reduction in lab-time." While these indications speak strongly to the hypothesis that mixed reality visualization of layout metadata on PCB can increase efficiency, a systematic study is yet to be published. The Mascot [7], a robotic workbench from Robotas, helps to support operators performing hand assembly of through hole components by steering a projected laser spot to the installation location on an anchored PCB. Similarly, Hahn et al. [18] generated an AR tool with textual and graphical cues delivered through a smartglass for assisting workers performing PCB assembly, indicating that the tool allowed for errorless part picking and assembly. Hahn et al.'s tool, InspectAR and Mascot all provide board-locked augmented instruction for PCB workflow, driving information from the virtual design files to the user's view of the PCB. Our work broadens the design space seeking to also incorporate augmented interaction and measurement to pass data in the opposite direction, that is, interactive capture in the PCB view can be passed to the virtual design files.

3.2.2 Adding Interactivity and Measurement. Pinpoint [25] is a tool designed to assist in PCB debugging by allowing users to modify and measure the circuit *in situ* after the PCB is fabricated. The tool modifies the layout of a PCB by inserting breakable connections on some traces. While not using augmented reality per se, the tool connects the virtual and the physical by using GUI-controlled relays to make and break these connections. For form factor designs and mass-produced PCBs, modifying the layout for test is typically restrained to adding test points on critical nets for bed-of-nails, on-line testing or manual access for workbench debugging. Our work seeks to support existing debugging workflows that do not modify the PCB design, and instead ease access to measurement points by guiding users with augmentations. More relevant to our work, BoardLab presents a magnetically tracked stylus that enables interactions from board to schematic, such as selecting and identifying components on the schematic by touching the components on the board as well as taking voltage measurements and having the measurement annotated on the schematic [17]. Although the system looks promising, no formal evaluation was reported. Our work studies whether the interactions afforded by such a stylus would be helpful to electrical engineers, as well as exploring interactions that are synergistically enabled as augmented interaction and measurement is paired with simultaneous augmented visualization.

4 STUDY 1: FORMATIVE NEEDS FINDING

To gain an understanding of the needs of electrical engineers during their debugging workflows and characterize their existing workflows, tools and methods, we conducted a formative needs finding survey and semi-structured interviews.

4.1 Participants and Procedure

We recruited 8 participants who hold electrical engineering roles in academic labs and industry (high technology, consumer electronics firms). While all of our participants regularly design and debug their own PCB designs, their experiences spanned one-off or low-volume designs for research or hobby purposes, complex development boards with thousands of components, and mass-produced form factor logic boards shipping hundreds of thousands of units (see Table 1).

We conducted remote semi-structured interviews with the participants. Each of the interviews lasted for about 1 hour and consisted of 3 main parts:

(1) We asked the participants about their current debugging flow, strategies, common pain-points, and needs.
(2) We solicited feedback on initial speculative design concepts that we described using sketches and hypothetical use-cases. Participant were also invited to share any functionality or interactions they were missing in the currently existing tools.
(3) We asked them whether collaboration was important in their day-to-day work. When relevant, we specifically asked about how the information is transferred when more than one person is involved in the workflow.

We supplemented the interview data with our own professional experience debugging PCB in both industry and academic institutions. We analyzed their responses via thematic analysis [11], first transcribing the interviews, then coding recurring themes, and finally noting outliers from the norm.

4.2 Findings

From our participants' responses in the interviews, we extracted four sub-tasks that constitute a framework for localizing errors during debugging:

(1) Perform a visual inspection, measure output one sub-section at a time, and compare to expected values
(2) Identify an anomalous measurement and hypothesize fault causes, such as defects in design or processing
(3) Examine potentially contributing elements and make localized measurements to test hypotheses
(4) Compare real measurements to expected values derived from schematics, layouts and datasheets

Most of our participants alluded to the challenges of context switching and information logging while debugging a PCB. They raised concerns about referencing a large set of information sources during debugging (on their computer monitor or sometimes printed paper: schematics, layouts, datasheets, bills of materials, emails; on their workbench: instrument measurements, PCBs) and the frequency with which they moved between these items: *"Very often, maybe multiple times per minute"* (P1, Quote Q1). *"I would say almost constantly until I get to fab C or D"* (P2, Q2). *"I would say at least multiple times a minute I'd switch back and forth."* (P7, Q3). *"Gotta go back and forth and each time you go back and forth you add more information to your schematic, and eventually find a value that doesn't line up... Probably five or six times a minute"* (P8, Q4).

Participants stated the information they cross-referenced most often in this process included component reference designators (toward the task of component localization), component values, pin or net locations (for the purpose of determining where to probe), net connectivity, measured values from instruments, IC pin assignments, and IC orientations.

The challenge of component localization was not shared by one participant who pointed out that, in doing the end-to-end PCB design process, she was able to memorize all of the components of the design. In addition, due to the high voltage nature of her work, her circuits generally included fewer but larger components than the other participants typically worked with, allowing her to include reference information on the silkscreen of her fabricated PCB. *"I made the PCB. I verified that my design works in a PCM simulation; I mean I never had a situation where I couldn't find my component [from memory]"* (P4, Q5). Another participant also expressed a similar sentiment *"I have it memorized"* (P5, Q6). Both noted that confirming orientation and pin assignments, as well as localizing small components on complex boards still pose a challenge for localization. Participants expressed interest in having component, pin, or net metadata within their view of their board, but looked to avoid interference: *"Yeah, that would be really helpful as long as it didn't interfere with my ability to see the PCB."* (P8, Q7). *"I usually like having two scenes. Like sometimes I don't want the information... just want to know the reference designator... but then sometimes if I'm debugging, like show me that info that I need: [lists various metadata categories]."* (P5, Q8).

Finally, participants cited other explicit processes where they felt their software and design tools fell short:

Assembly: While not all users assembled their own boards, those that did expressed frustration in matching the ordered components to the correct location and orientation on the PCB.

> "So you have to have a separate BOM that you make yourself, like an Excel document or a Google document. I'd have that, the schematic, and I have the layout up on my laptop so I'd be switching back and forth between all of them, trying to figure out where each component went. it was not fun." (P7, Q9)

Measurement: The need to localize probe points, trace net connections, and log measurements were shared as common pain points.

> "It's typically just like you look down at the board you make a measurement, you know, you might have a Google Doc with your testing records in it or something that you're documenting as you go." (P8, Q10)

Bring-up: Before green-lighting an entire production run, the bring-up process is followed after completing the first board assembly: 1. visual inspection, 2. confirm correct component stuffing, 3. confirm correct pin one locations (an indicator for verifying orientation), 4. perform open/short test on all voltage rails, 5. apply power with current-limited supply, 6. check each voltage rails for correct level, 7. perform functional sub-system checks. Some users resorted to general purpose software to prepare customized views ahead of time.

> "What I'll do is [that] I'll have a premade PowerPoint deck, and I have everything I need with all the steps

Table 1: Recruited participant backgrounds.

	Field	Experience	Primary Tool	Designs	Study 1	Study 2
P1	Academia	Design, Release, Assembly, Functional Check, Rework	EAGLE	Mixed Signal, Embedded Systems, Wearables, Typically small, two-layer boards	X	X
P2	Industry	Design, Release, Engineering validation, Mass production, Field failure analysis	Cadence OrCAD/Allegro/PCB, Zuken CADSTAR	Mixed signal and high speed development boards (large format, thousands of components, 14 layer), form factor for wearable devices (12 layer)	X	X
P3	Academia	Design, Release, Functional check	EasyEDA	Antenna Patterning	X	
P4	Academia	Design, Release, Assembly, Functional check	Altium	High wattage power circuits	X	
P5	Industry, Hobby	Design, Release, Engineering validation, Mass production	Cadence Allegro/PCB	LED display, GPS radio module, Charging and battery protection circuits, FPC	X	X
P6	Industry	Design, Release, Engineering validation, Mass production	Cadence Allegro/PCB, Altium	Mixed signal, Actuator drivers, High-voltage designs, FPGA boards, form factor for wearables (4 layer), FPC	X	X
P7	Industry	Design, Release, Assembly, Engineering validation, Mass production, Field failure analysis	Cadence Allegro/PCB, KiCAD	Small form factor for wearables (12 layer), large form factor boards for gaming console (12 layer), FPC, RFPC	X	X
P8	Academia, Hobby	Design, Release, Engineering validation	Eagle	High-voltage designs, Aerospace, Power electronics	X	X

> and all the images I need, tables that I can fill in. So I'll, you know, identify all the pin one locations and what's stuffed and not stuffed with the picture that I make ahead of time." (P7, Q11)

Collaboration: A more frequent occurrence as a result of the recent COVID-19 pandemic, a handful of participants felt that working with collaborators who were less familiar with their own design, tackling a new developer kit, or approaching someone else's design posed new challenges.

> "I'm actually going to pass this design to this other engineer who's going to get some of those boards, who's not familiar with the design and I think for someone that's, you know, not familiar with the design [who] is trying to do like bring up, that would be extremely helpful to go or like look at a part or touch part with a stylus and have it pull its datasheet and point it to like, where it is on the layout and schematic... Yeah I think it's kind of brutal with what we do at [redacted], which is now, there'll be like a rework for [the technicians], and we'll have to like manually label. We have to take a picture of the layout, and then bring it into like some type of editing tool like PowerPoint and then like add arrows to the points that we want to probe... it's like we're passing back like a billion, like little like pictures, and you have to like talk on the phone a lot about it." (P5, Q12)

Two industry engineers shared that communicating with technical staff from the fabrication house was sometimes challenging as often these technicians who actually fabricated, reworked and tested the boards were not familiar with the design itself. In these situations, engineers usually turned to printed materials, annotated images, or emailed presentations to communicate their design intent.

4.3 Design Considerations

From this first study, a few primary design considerations were motivated by the feedback shared amongst participants:

- DC1 **Reduce context switching and facilitate pattern matching:** Participants expressed the need to move between different representations of the schematic, layout, and board often (Q1—Q4, Q9, Q10). Component, pin, and net localization in particular was cited as tedious since going between information in the schematic and board involved pattern matching with the layout as an intermediary, particularly in dense, complex boards without silkscreen.
- DC2 **Show relevant information without cluttering the view:** Participants expressed interest in having context-relevant information accessible through both the design files and also when directly interacting with the board, but were wary of excess visual clutter (Q7, Q8). Some suggested making the display of information optional, such that the user could decide to turn it off.
- DC3 **Support habitual and intuitive interactions with the PCB:** While participants each followed slightly different methods of localizing issues and taking measurements, they generally followed a common approach (Q1—Q4, Q7, Q8). An AR system should simplify methods of taking measurements or seeking information, but should not depart greatly from current habits or workflows.
- DC4 **Facilitate collaboration:** A few participants noted that finding design elements and following measurement procedures were exacerbated when working with collaborators who were less familiar with a particular design (Q12). A solution that guides the user with relevant information can be extended to help support collaboration (in real-time or asynchronously).

5 INTERACTION TECHNIQUES

We derive four core interactions motivated by the needs finding feedback and design considerations discussed above. We then assemble these core interactions as building blocks to support the debugging workflows followed by engineers. We use the term *element* here to refer the circuit elements of a component, pin, or net within the design. We use the term *element identifier* to refer to the component reference designator, pin number, or net name for their respective elements. In this paper, we will refer to a hypothetical system that implements these proposed interactions as *Augmented Silkscreen*. We will refer to the interview asset we produced and used for evaluation as the *simulator*.

5.1 Core Interaction Techniques

The core interactions are categorized by direction of information flow: either from the design files (schematic and layout) toward the PCB, or from the PCB toward the design files (Fig. 2).

5.1.1 From Design Files to PCB. The two core interactions relating design files to the PCB are *element localization on PCB* and *metadata annotation on PCB*.

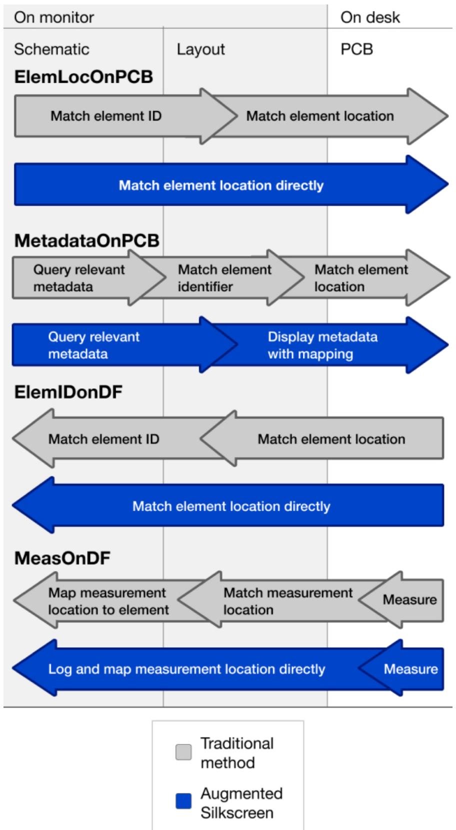

Figure 2: Information Flow of Core Interactions

Element localization on PCB (ElemLocOnPCB): As per DC1, we found that engineers traditionally follow a two-step process to localize elements on the PCB given a target element on the schematic. First, they textually pattern-match an element identifier to the corresponding one in the layout file using the find command. Then, they spatially pattern-match the layout to the PCB to identify the corresponding PCB element. A few ECAD tools [2, 6] support cross-probing between schematic and layout. Using an AR system allows us to extend this interaction to the physical PCB, such that a selection in the layout or schematic view results in an augmented highlight of the matching design element directly on the PCB.

Metadata annotation on PCB (MetadataOnPCB): During the process of debugging, engineers often query the schematic for element attributes that determine the function of the circuit, such as resistor values, IC packaging, diode reverse voltages, or inductor max currents. They keep this knowledge in short-term memory as they subsequently formulate hypotheses for a root cause or look to take their next diagnostic measurement. As per DC1 and DC2, we seek to minimize the cognitive load of keeping information in short-term memory by bringing this information to the PCB through annotating the PCB element with this element metadata in the view of the user. Additionally, user-inputtable text field annotations can enable users to annotate elements with freeform notes.

5.1.2 From PCB to Design Files. The two core interactions relating PCB to the design files are *element identification within design files* and *measurement annotation within design files*.

Element identification within design files (ElemIDOnDF): We learned that engineers follow the same two-step process as described in ElemLocOnPCB in reverse to identify or localize elements on the schematic given a target element on the PCB. Pertinent to DC1, we propose enabling directly making selections on the PCB instead via an interactive probe to select, identify, and localize the same element within the schematic and layout. Probes are commonly used in PCB measurement tasks and are therefore a familiar method of direct PCB interaction.

Measurement annotation within design files (MeasOnDF): Finally, taking diagnostic measurements is a key part of debugging workflows. Augmented Silkscreen would support this interaction by capturing measurement data from benchtop test equipment probed on the PCB and relaying it back to the design files addressing DC1 and DC3. As a practical implementation note, nearly all benchtop test equipment break out their `get` and `set` functions over SCPI/VISA, a standardized measurement instrument API.

5.2 Interaction Technique-Supported Workflows

We synthesize these core interaction technique building blocks to support entire debugging workflows.

5.2.1 Diagnostic Measurement. Participants described the process of capturing and logging measurements as an important method to assist in deductive root cause analysis. Combining the ElemIDOnDF

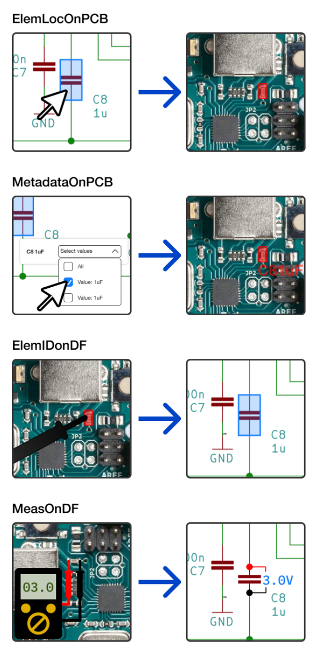

Figure 3: Depictions of Augmented Silkscreen core interactions

and MeasOnDF interactions enables users to take a measurement with probes (MeasOnDF), capture the location the measurement was taken (ElemIDOnDF pin), identify the involved nets (ElemIDOnDF net), and include the information on the design file view along with the captured measurement. For example, consider a user that wishes to record measurements, and so starts a new debugging session in Augmented Silkscreen's design file view. The user may take a measurement of a voltage rail with a digital multimeter. From the position of the probe locations on two pins, the corresponding nets for the positive and negative probe terminals are determined via ElemIDOnDF. The measurement value is captured along with its location in the design file view via MeasOnDF.

5.2.2 Bring-up. When engineers first apply power to their boards, they typically follow an exacting protocol to ensure all components were assembled properly. By automating ElemLocOnPCB interaction, all uninstalled component locations and all pin one locations (indicative of correct component orientation) can be highlighted directly on the PCB permitting rapid visual checks. Similarly, by entering a set of desired nets to test into the design file view, and optionally providing functional limits, Augmented Silkscreen can sequentially display probe points on the PCB, again via the ElemLocOnPCB interaction. A user may then follow the *diagnostic measurement* technique described above to sequentially capture the measurements back to the design files for comparison to set limits.

5.2.3 Visual Inspection. Participants described the need to sometimes query an element's metadata directly within the board view, for example, after noticing a given component's rise in temperature or in determining to which net a certain pin was connected. By combining ElemIDOnDF and MetadataOnPCB the user can select an element directly via probe on PCB and have the metadata annotated directly in the PCB view without having to refer to the design files.

5.2.4 Remote Collaboration. To facilitate remote collaboration, many participants pointed out the need to call out to specific elements on the board with a set of instructions. In support of DC4, this can be achieved by splitting Augmented Silkscreen's design file view and augmented PCB view across two locations. Synchronous collaboration can be enabled if one user (for example, the board designer) has the design file view and the other user (the remote debugger) has the PCB. The designer may select elements such as component or pins (probe locations) to display on the remote debugger's view of the PCB via ElemLocOnPCB. The *diagnostic measurement* interaction may then be used to capture the remote debugger's measurement values back to the designer's design file view. In an asynchronous collaboration, the designer could leverage the MetadataOnPCB interaction to tag elements in the PCB view with freeform instruction call outs. This could be helpful for communicating rework instructions or step-by-step debug procedures.

6 STUDY 2: USER EVALUATION

To solicit feedback on the interactions we designed, we remotely conducted another round of structured interviews using an interactive simulation. Each interview lasted approximately one and a quarter hours.

6.1 Participants and Procedure

For continuity, study 1 participants were re-invited to participate. Six out of the original eight participants were able to join (see Table 1). No additional participants were recruited. The study was divided into three main sections:

(1) Feedback on core interactions and interaction-supported workflows
(2) Feedback on variations on the attributes of core interactions
(3) Timed element localization tasks

6.1.1 Interview Assets. To help participants envisage and solicit feedback on our interaction concepts during the remote interviews, we produced two artifacts: a web-based, interactive PCB simulator and a set of envisioning video sketches [27]. The simulator mirrored the workbench setup described by participants in Sec. 4.2. It comprised of two components: (1) an in-browser, interactive schematic and layout viewer on the participant's monitor (Fig. 4(A)) simulating the schematic and layout viewer an engineer would have open on their computer during debugging, and (2) an in-browser, interactive top-view PCB image on the participant's own touchscreen device (Fig. 4(C)) simulating the PCB the engineer would have on their lab bench during debugging. In order to deliver Augmented Silkscreen-interactions that span design files and board augmentation, the two were linked via a web socket enabling real-time interaction in the schematic and layout viewer to affect augmentations on the mobile PCB view, and vice versa. Participants could select a component, pin, or net in either the schematic, layout, or mobile PCB view (via clicking or tapping via probe) and have the corresponding elements highlighted in the other two views. Additionally users could right click on a component revealing metadata and a button to show that metadata augmented on the PCB view (Fig. 4(A). Presenting interactions via this simulation prototype allowed for the users to use their own devices at home and allowed us to easily modify specific attributes of the way the core interactions were presented to users (see Sec. 6.1.6). This web socket could also be disabled to test situations without Augmented Silkscreencross-linking between design files and board (see Sec. 6.1.4). For the evaluation, we used the schematic, layout, and PCB image of an Arduino Uno R3[3].

Additionally, to further assist users in visualizing the interactions, we recorded a set of POV video walkthroughs for each interaction technique and each interaction-supported workflow. Each sketch illustrated the schematic and layout interactions in screen capture and view of a desk top PCB in a time-synced inset (ex. Fig. 4(D)). The PCB augmentations in the shown videos were projected via overhead projector (AAXA P7[4]). A narration also helped to describe the interaction. During the interview, if the participant was unclear on the video content, the interviewer provided additional description until it was clear.

6.1.2 Part 1 Procedure. The participant was shown each video sketch and the interactive simulator, starting with core interaction techniques and ending with interaction-technique supported workflows. Between each video sketch, the participant was then verbally asked the following questions:

[3]https://store.arduino.cc/usa/arduino-uno-rev3
[4]https://www.aaxatech.com/products/p7-pico-projector.html

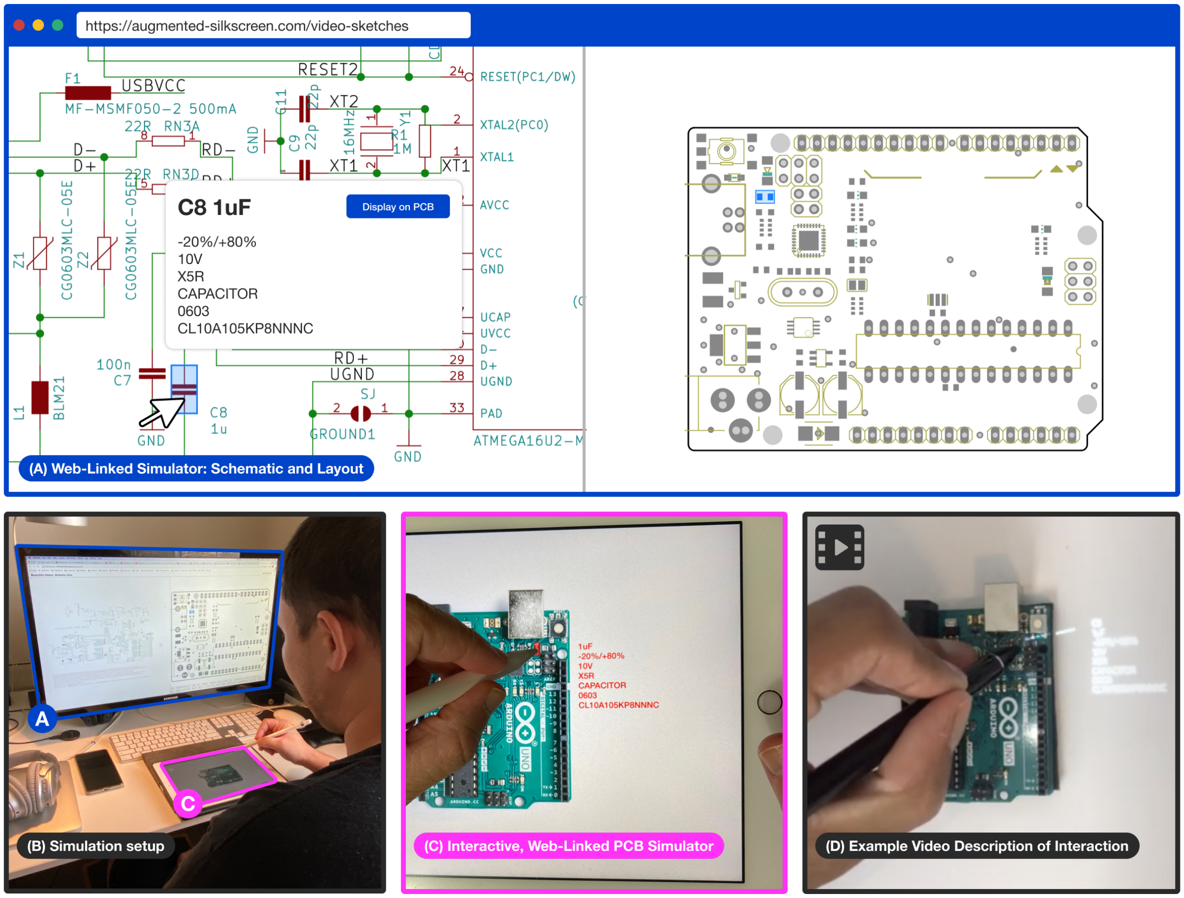

Figure 4: The web-linked simulator consists of two components: an on-monitor design files viewer and an on-device PCB view. (A) In-browser canvases contain interactive views of the schematic (left) and layout (right) of the design, just as engineers would have on their monitor during PCB debugging. Here, the user has selected capacitor C8 in the schematic (A, left), which would cause the corresponding element to highlight in both the layout (A, right) and PCB view (C). (B) A remote participant has the simulator open on their monitor and touch screen device. (C) The PCB simulator imitates a PCB a user would have on their desk. Here, a component is augmented with a box highlight and a metadata annotation. A user can use a probe to interact with the PCB simulation, which would affect the state of schematic and layout (A). (D) Freeze frame of inset from an example video description (Visual Inspection video). The video actually shown to the user contains a time-synced screen recording of design file view (A) with (D) inset picture-in-picture.

(1) *"Would you find this interaction to be helpful, not helpful, or have no impact on your debugging workflow?"*
(2) *"How might this interaction affect your workflow?"*
(3) *"How likely would you be to adopt this interaction on a scale from 1 (would not use) to 7 (very likely to adopt)?"*

We analyzed responses via thematic analysis [11], by transcribing the interviews, coding recurring themes, and noting outliers.

6.1.3 Part 2 Procedure. To better understand how certain design decisions align with the stated design guidelines, we asked participants to assess variations on attributes of the core interactions in five categories (Fig. 5):

(1) *PCB Target Visualization:* Per DC2, how does the visual design of augmentation influence element localization in ElemLocOnPCB? Options: (a) Box—filled in rectangle, (b) Circle—unfilled circle, (c) Crosshair—perpendicular, intersecting lines, (d) Layout inset—highlight is shown briefly, an inset segment of the layout local to the target element is projected next to the board
(2) *PCB Annotation Method:* Per DC2, what is the preferred length and where is the preferred area for on-board annotations in MetadataOnPCB? Options: (a) None—no annotation, (b) On-board minimal—annotation depicting only element identifier and value projected adjacent to the element (potentially overlapping the PCB), (c) Off-board minimal—only element identifier and value; projected on tabletop adjacent to the board, (d) On-board full—all element metadata fields projected adjacent to element, (e) Off-board full—all element metadata fields projected on tabletop

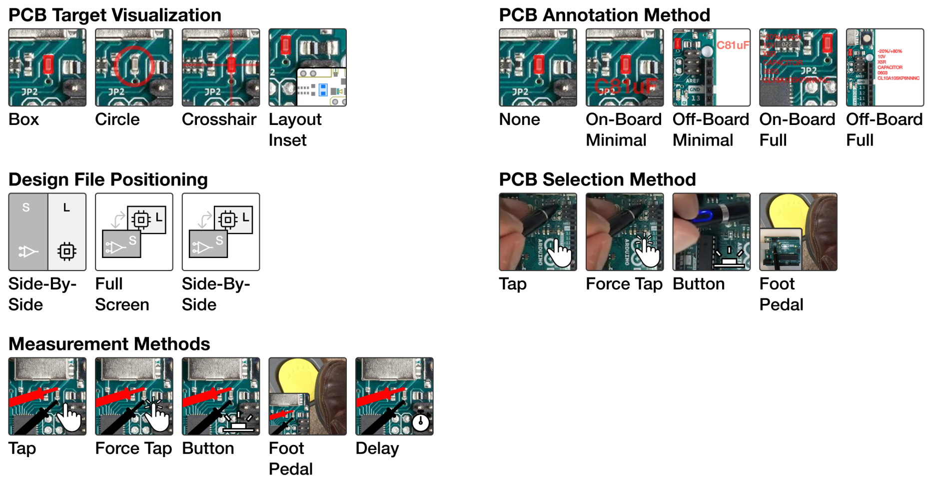

Figure 5: The matrix of variations we presented to participants to elicit design feedback on attributes of the core interactions.

(3) *Design File Positioning:* Per DC1, how the positioning of design files on monitor influence element identification within design files in ElemIDOnDF? Options: (a) Side-by-side, (b) Full screen—schematic and layout each took entire screen, flipped between files, (c) Layout inset—layout local to the target element is inset on schematic view
(4) *PCB Selection Method:* Per DC3, what is the preferred method to trigger selection of PCB elements with a probe for ElemIDOnDF? Options: (a) Tap—tap top of element briefly to select, (b) Force tap—similar to BoardLab, applying force to probe tip triggers selection, (c) Button—button on barrel of probe triggers selection, (d) Foot pedal.
(5) *PCB Measurement Capture Method:* Per DC3, what is the preferred method to trigger a measurement capture on PCB with a probe for MeasOnDF? Options: (a) Tap—tap pins to capture selections, (b) Force tap—applying force to probe tips triggers capture, (c) Button—button on barrel of probes triggers capture, (d) Foot pedal, (e) Delay—stationary probes for 3 seconds triggers capture.

Items (1), (2), and (3) were delivered via the interactive web simulator. Items (4) and (5) were described with the video sketches. After the demonstration, we asked participants about their general impressions, how they would rank the presented variations, during which workflow they might use it, and why.

6.1.4 Part 3 Procedure. Users performed two timed component selection tasks: finding components on the board given a target in the design files, and finding a component in the schematic given a target on the board. An interactive web simulator delivered the schematic and layout on their monitor, and a PCB image stand-in on their touchscreen device in an imitation workbench set up (Fig. 4(B)). A standard capacitive stylus shipped to the participants was used to select components on the PCB. For the *find on PCB task*, a target component was highlighted on the schematic and layout. The user's task was to select the corresponding component on the board. When Augmented Silkscreen (cross-linking between design files and board) was enabled, the target component on the board had an augmented highlight as well. For the *find on schematic* task, a target component was highlighted on the board. The user's task was to select the corresponding component on the schematic. Selection cross-linking between the layout and schematic as in KiCAD [6] was enabled as baseline. When Augmented Silkscreen was enabled, the target component on the schematic and layout were highlighted. For each task, all six users performed twenty different component selections: ten selections with Augmented Silkscreen and ten without, with order randomized (for a total of 60 samples per condition minus omissions). Users were permitted a short practice round to familiarize themselves with the selection task. Audio feedback indicated if a user selection was correct or not. Schematic and layout visualizations were kept the same between conditions. Timing started when the component to be found was presented to user (on the schematic and layout in the *find on PCB task* and on the board in the *find on schematic task*). Timing stopped when the user selected the correct corresponding component (on the board in the *find on PCB task* and on the schematic in the *find on schematic task*). If a user selected the incorrect component, the data was logged as a mistarget; if a user indicated it was due to a fault of the capacitive stylus or touchscreen, rather than a true mistarget, the datum was omitted.

6.1.5 Findings from Part 1: Feedback on Core Interactions and Interaction Supported Workflows. Users provided illuminating feedback on their preferences and uses of the core interactions and workflows (Fig. 6).

Users unanimously rated EL-PCB favorably for localizing components, pins, and nets. *"It basically saves me an extra step... this allows me to go from schematic to board immediately"* (P1, Q13). *"I have to do it pretty much manually in different pieces of software. So, this would have saved me a lot of time"* (P6, Q14). Specific situations in which EL-PCB would be particularly helpful included working with complicated and dense boards (P1), unfamiliar boards (P2), or boards without silkscreen (P5). P7 also pointed out that *"trying to find specific patterns, especially with things are rotated and whatnot*

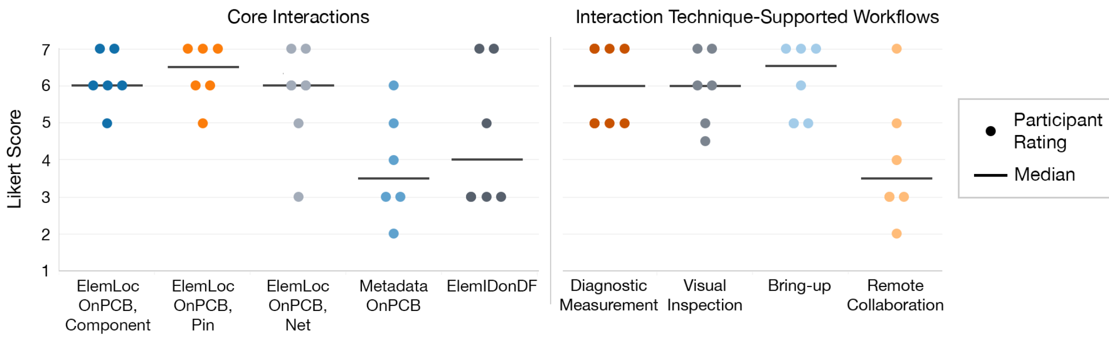

Figure 6: Results from Part 1 (core interaction and workflow feedback).

is very error prone" (Q15) for humans, and that computationally-driven augmentation can help to disambiguate.

P1 cited that there might be practical issues with the precision of an AR system highlighting very small details (the 500 µm pitch pins of a QFN24 package), and P5 raised concerns with the potential for trust issues if the projections were inaccurate: *"this would be useful but misleading, because yeah what if they got stuck with the wrong part."* (Q16) As P7 pointed out, however, *"as the pitch gets more and more fine, being able to identify the pin doesn't help quite so much, because there's not a lot you can do if it's so small that you can't probe it [anyway]."* (Q17)

Nonetheless, highlighting specific pins connected to a given net was also liked, specifically to identify suitable measurement probe points for a given net: *"[when] looking for a component that was easy to probe without shorting anything else nearby...I don't even have to worry about like looking through the package size on the schematic sheet and trying to find something that's like on the net that's going over different schematic pages. [With augmentation] I can just literally look at the board and say, 'Oh, I got this big capacitor right here that has a good probing point.'"* (P6, Q18). Another user looked to use this feature to locate accessible nodes since many are obstructed by features such as shield cans or mechanical cases "in industry boards." (P2, Q19).

As an interaction in and of itself, MetadataOnPCB rated poorly, with many users citing that selecting metadata on the schematic to display within the board view was redundant: *"If I'm already on the schematic and the information is there on the schematic it's not super helpful to display it again on the side of the board."* (P2, Q20). However, when combined with EI-DF to form the Visual Inspection interaction (allowing users to trigger augmented metadata annotations directly from the board as opposed to the schematic) the interaction was universally preferable. *"I'm doing this 24/7, like which pin is which... if I could do something to where I could just be looking at the board, and not have to look away, and just having it projecting on top of it. That would be – oh my god – I would use that like 100% of the time."* (P5, Q21). P2 added that *"in factory environments... this feature could limit the number of different screens or devices I need to carry around."* (Q22)

Diagnostic Measurement workflow was found to be helpful in not only alleviating an individual's own mental demands but also in capturing unambiguous logs for documentation and interacting with others. *"This would be extremely helpful, because... what I've been doing up until this point is, I may like probe something and then I'll just keep in the back of my mind like okay it's this value. And then I'll just keep like three to four values in my head"* (P5, Q23). *"I've had personally a whole lot of negative experiences of people measuring the wrong things and telling us that they got such and such results."* (P8, Q24).

All participants appreciated the interaction technique-supported Bring-Up workflow: *"very helpful useful during board bring-up, we're just trying to take a basic DC measurement on multiple nodes quickly, smoke test... a lot of times that's tens of nets that we're trying to measure,... and if we can very quickly step through that without having to go back and forth between the board and the PC recording and Excel or whatever, would definitely save a lot of time."* (P2, Q25). *"I could see this like cutting down board bring-up down in time by like hours."* (P5, Q26) Some users suggested they also found it useful to define and project probe points for unstructured debugging: *"[I'm] interested in just the fact that it can highlight the two things I want to probe at the same time though so I know exactly where to put those probes."* (P1, Q27). The explicit probe point projections inspired some users to suggest it can assist train those unfamiliar with their design. *"If I was giving this to like a, like undergrad or like an intern or something you know, probably be pretty useful for them just to quickly catch on"* (P1, Q28). *"If you... can turn this [probe point projection] into an automation program... this would be great at a factory."* (P5, Q29).

Finally, the Remote Collaboration scenario received a varied set of rankings. One user, affected by remote work during the recent COVID-19 pandemic said, *"I think just the ability to communicate very unambiguously; not only like what part, you know, because you could use the reference designators in like an email or a phone call or whatever to tell people what to look at, but, like, in terms of selecting actual points to measure. I think that would be super helpful. Personally, what I've had to do a lot of recently is taking pictures of the manufacturing preview, and then like drawing a circle on what we need and then sending that back and forth on Slack. And so just, you know, in the sense that, that will speed that up a lot."* (P8, Q30). Many users thought the interactions were useful, but were not frequently involved in collaborative debugging situations. They recognized that they may have growing utility as the pandemic continues to affect workplaces: *"I think [these interactions] can be*

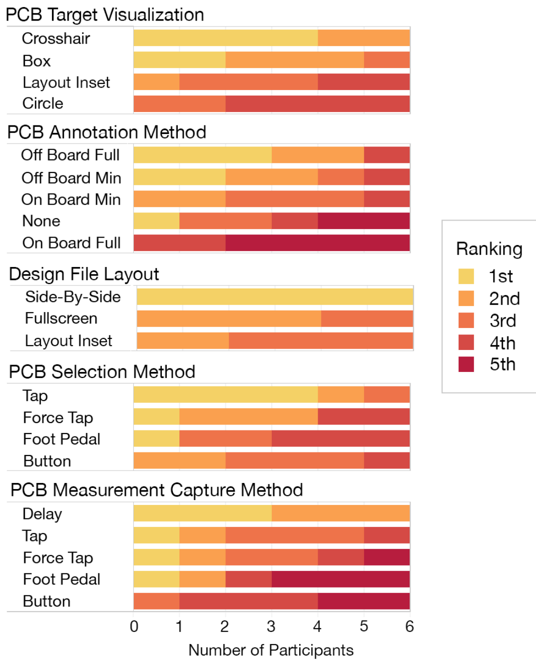

Figure 7: Results from Part 2 where users ranked their preferred variations of core interaction attributes.

useful and especially I guess in a situation like we have right now where everyone's work-from-home. And if I don't want to go to a factory, and there's lab techs at the factory, and they're there trying to figure out what's wrong with the board, and they don't necessarily have the expertise about that board, I could probably walk them through it and highlight stuff...I could see it useful in that kind of situation but it's more of a hypothetical because it's not something that I've really done much yet myself." (P6, Q31). One user also expressed that it'd be helpful to point out to their software colleagues certain buttons, switches, and plugs to interact with on their development kit, but they are not often in situations where the other user would be actively probing pads.

6.1.6 Findings from Part 2: Feedback on Core Interaction Attributes. Users ranked target visualizations for ElemLocOnPCB. Crosshair was the primary choice, but box was seen as nearly as good. *"The crosshair makes it super easy to individually pick out which one is which"* (P7, Q32). *"box one is my favorite because it superimposes the most accurately on the part of interest"* (P5, Q33). A user suggested that a crosshair transitioning to a box is good for initial targeting and reducing visual clutter. The circumscribed circle was missing the details of the component's contour, reducing visual precision and as a result being ranked lowest among most participants. *"The circle is misleading because it's like encompassing multiple parts"* (P5, Q34). Towards this end, an AR system must be precise enough to provide unambiguous augmentations on PCB elements (which can be less than millimeter square for the smallest pins). To help resolve ambiguous cases, a local inset of the layout was appreciated as visual confirmation, but users looked for it to be combined with an on-board visualization rather than as a standalone method.

Toward annotating metadata on the board (MetadataOnPCB), participants generally preferred off-board annotations (indicating that on-board annotations felt cluttered), but there was disagreement on how much information to present—between our options presented, there was an even split between all metadata or none, suggesting there is likely some ideal middle ground. Some users indicated their preference would be to control what metadata would be presented. Uniquely, P7 preferred no annotations, but if choosing one, preferred to have a minimum amount of information right on the board, citing that it yields the shortest distance between the component and annotation.

For identifying elements in the design files (ElemIDOnDF), we asked users their preferred design file view to better understand if having an augmented view of the board changed their current design file habits. All preferred to maintain a side-by-side view of the schematic and layout simultaneously, screen real estate permitting, but also looked to have a full screen options as well. Two participants saw value in the layout pop up: *"I do like the spirit of the peek when you click on it, especially if you... want to try and get your bearing with where the component is on the board, but I feel like if you have the crosshair you don't really need that so much"* (P7, Q35). On the other hand, some participants felt like the inset covered information in the schematic. One user indicated that the augmented board view was usable enough, it could eliminate their need to have the layout view on their screen. *"If I was debugging, every time I needed to find a component I would use this feature, there's no reason I would look at the board file if I had this feature"* (P7, Q36).

For the on-board element selection method in support of ElemIDOnDF, most of the participants indicated, if technically feasible, a simple light tap was preferred. *"The most intuitive one of the best is just tap to select with minimal force"* (P1, Q37). Multiple users worried that a force tap could damage small components, and that pressing a stylus button could cause probes to slip off small probe points. One participant liked the foot pedal selection the most, citing that it allows them to place probes carefully eliminating situations where components can be shorted or damaged.

Amongst the methods to trigger measurement capture (MeasOnDF), delay was the almost universal preference, as it matches the natural use of a multimeter (waiting for the measurement to settle). *"I feel like the way just a normal multimeter works is very intuitive, you just tap on things and like sometimes it takes [after] there's a delay on the screen."* (P1, Q38). One user (P6) ranked foot pedal at the top of the list, as they felt it was deliberate while allowing precise positioning of probes in both hands.

6.1.7 Findings from Part 3: Timed Element Localization Task. On average, users performed the *find on board* task 31% faster with Augmented Silkscreen compared to without, with a mean difference of 1135 ms across all samples (per-sample t-test, $t(58)=4.31$, $p<.001$; per-participant Wilcoxon Signed-Rank, $Z=0$, $p=0.031$). True mistargets fell from 16.94% without Augmented Silkscreen to 8.47% with Augmented Silkscreen. Users rated ease, rose from a median of 5.0 to 7.0, and confidence rose from 5.5 to 7.0, on the 7-point scale. *"I was very confident [with Augmented Silkscreen] because generally I knew what I was looking for, and also the highlight was basically telling me it, I didn't need to double check it most of the*

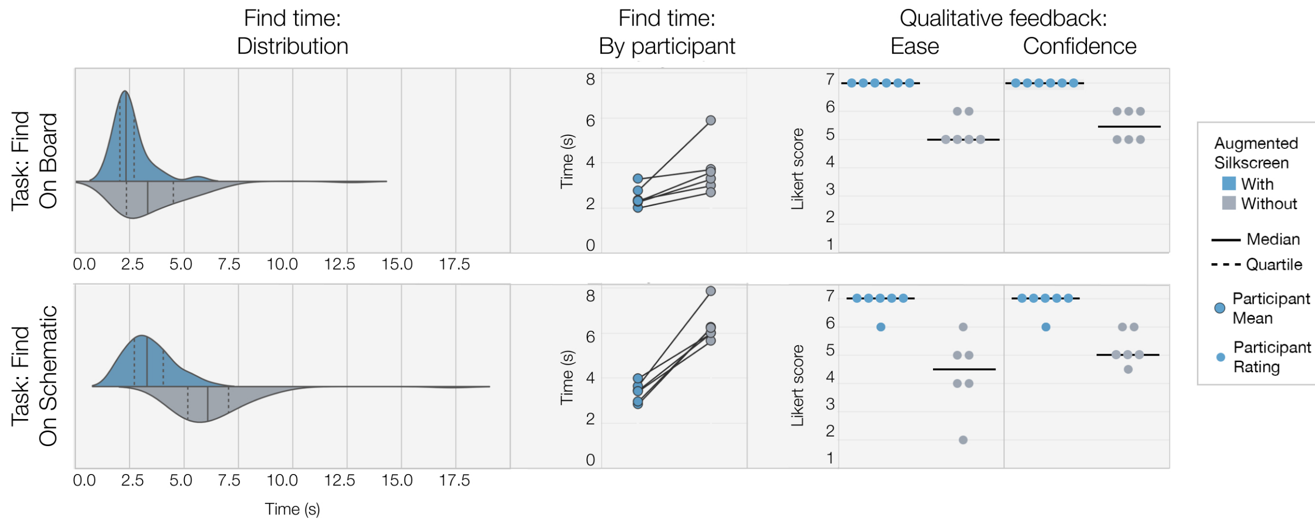

Figure 8: Results from Part 3 where users participated in a timed element localization task.

time... Without the highlight, I'd have to look at it, choose which one I'm pretty sure it was, double check, and then go back and click once I was confirmed what I thought was." (P6, Q39).

In the *find on schematic* task, users were, on average, 46% faster with Augmented Silkscreen, with a mean difference of 2923 ms across all samples (per-sample t-test, $t(59)$=10.1, p<.001; per-participant Wilcoxon Signed-Rank, Z=0, p=0.031). True mistargets were 3.33% for both conditions. Users' ease and confidence score increased to a median of 7.0 for both metrics, from 4.5 and 5.0 respectively, out of 7 points. *"If I have to click it on the layout and have it show up on the schematic, yeah, that's helpful compared to what I have... it already, you know, just saves me the step of switching windows essentially, in searching."* (P7, Q40).

We note that the selection task given may have been too easy (as evidenced the by high ease score for the baseline) with a single page schematic and small, low component count board relative to what is typically found in a commercial product. A more complex design (with greater number of schematic pages and higher component count) may have yielded a starker difference between control and condition with Augmented Silkscreen, with the control more likely to take tens of seconds to minutes to localize a given component as per the qualitative feedback during needs finding. *"Definitely would be significantly easier with the AR link just because we're navigating like 55 page schematics as opposed to this simple one pager here with a really simple layout, so it's much more difficult to keep your schematic and board view aligned... It's a lot more like zooming on the board side, page changing on the schematic side, and then cross referencing to a real PCB."* (P2, Q41).

7 DISCUSSION

Through a three-part study, we have explored the design space of using AR visualization and interaction as tools for assisting electrical engineers with their PCB debugging workflows and preliminarily evaluated a proposed set of interaction techniques. In particular, we found that four specific tasks benefited the most from our proposed interaction techniques:

(1) finding components (ElemLocOnPCB components) and probe points (ElemLocOnPCB pins and nets) on the PCB,
(2) immediately providing element metadata at the board without referencing design files (Visual Inspection)
(3) logging of unstructured measurement queries with associated probe points (Diagnostic Measurement)
(4) unambiguous, spatially co-localized, and potentially automated probe point visualizations for directed measurement workflows (Bring-up)

In support of DC1, participants' most cited reason for their expectation of increased efficiency was confirmed to be the reduction of context switching between files. For example, ElemLocOnPCBremoved out the need to reference layout when moving from schematic to board (Q13, Q14, Q15), Visual Inspection cut out the need to flip from board to schematic to pull metadata information (Q21), and Diagnostic Measurement and Bring-Up workflows took out the need to flip to instrument panels and logging documents (Q25).

Careful design is needed towards maintaining a fine balance between providing relevant information and avoiding a cluttered view (DC2) and warrants further study. While users generally agreed that information should be provided out of the direct line of sight to the PCB, there was disagreement on how much is helpful (see MetadataOnPCB Findings). Breaking out control to users for them to adjust based on their context may be best.

The choices amongst users affirmed that supporting habitual interactions with the PCB (DC3) is an important design consideration, as evidenced by responses to the preferred PCB element selection method (simple tap, Q37) and measurement capture method (delay, in line with current behavior, Q38). Users were enthusiastic (Q23, Q25, Q26) about interaction-supported workflows that directly mirrored and supported their existing practices (e.g. Diagnostic Measurement, Bring-Up).

Finally, users generally were interested in how the techniques could help support collaboration (DC4), but for only one did the use case arise frequently enough to say they would adopt it (Q31). However, these situations may be increasingly common with a progressively more globalized electronics manufacturing pipeline, a stay-at-home pandemic, and decreasing knowledge barriers for participation in electronics design.

Feedback from participants elucidated a few practical challenges towards the construction of a future system regardless of how augmentations are delivered (head-mounted device, handheld mobile video pass-thru AR, projective AR, or other). First is the need for extremely precise and stable, board-locked visualizations. Users expected the system to be able to augment the smallest pin that can be reasonably probed, or a methodology to disambiguate imprecise visualizations. Second is the need for accuracy in board-locked visualizations. Users expressed that they would mistrust the system if it could not provide accurate overlays (Q16). Furthermore, PCB modifications during debugging such as reworked components or breakout wires may also cause the element on the board to no longer align with the design files. A function to support deviations from the imported design files could address this. Finally, on the wish list for one participant was a system that could be easily portable to a number of environments, such as the factory (Q22).

7.1 Limitations and Future Work

Due to the COVID-19 pandemic, studies were conducted remotely prohibiting the ability to collect observational data. Following the pandemic, we would look forward to observing user workflows directly in a simulated debugging task, beyond the video call with video prompts and a PCB simulator we used in this evaluation. We could perform the timed evaluation on a real, potentially more complex, PCB than the web simulator board which is limited to the real estate of the user's touchscreen. One participant did note, however, of our simulator, *"I felt like basically your emulation setup was pretty representative of like how the tool would be in in real life ... it carried over pretty well."* (P6, Q42). Future work could extend the timed element selection tasks in this paper toward a more general and open-ended timed debugging procedure allowing for multiple proposed interactions to be leveraged. The small number of participants may limit the generalizability of the findings. While we received interesting feedback from our relatively small sample size, a larger study could yield greater variety and nuance in the discussion, especially on topics where the responses were less uniform (e.g. MetadataOnPCBand Remote Collaboration). It would also provide larger effect sizes in the analysis of the quantitative data. Users cited that referencing datasheets is a common source of debugging information as well. Work towards parsing and linking component datasheets to be able to provide context-relevant data would further help to decrease context-switching. Only click- and touch-based methods of element selection and file navigation were considered for this study, but some users expressed interest in multi-modal methods combining voice or gaze.

While not explicitly a debugging procedure, users frequently commented that the methods proposed in this work can help speed up and decrease errors in assembly and potentially validation, warranting more thorough investigations toward these use cases. Augmented Silkscreen may also be used to streamline assembly workflows by sequentially highlighting the locations of each component installation location via ElemLocOnPCB, however a Bill-of-Materials (BOM) view is likely needed to help provide an ordering to the installation procedure. Engineering validation is a process in which board revisions are tested against a set of functional requirements. Augmented Silkscreen could be used to assist in preparing samples for test which can be a manual process, but as these tests often must occur on a large sample set of the population, automated test equipment is typically leveraged.

Finally, we are excited for future work to incorporate these interactions and feedback into a deployable augmented reality system. The system would comprise of three parts: (1) a graphical program running on the user's computer that presents the schematic and layout, (2) an augmented reality system that can deliver augmentations on the PCB, and (3) a probe to track the user's interactions with the PCB. To be practical, the program would need to be built as an extension of an existing ECAD tool or as a separate application able to ingest and parse ECAD schematic and layout documents. For delivering board augmentations, multiple methods of delivering AR augmentations are possible: via headset (as in Hahn et al. [18]), via see-through mobile AR (as in InspectAR [5], or via projective augmentation (as in Mascot [7]. The board itself can be tracked via computer vision (as in InspectAR) or fixed in known location (as in Mascot). A probe to interact with the board could be tracked via magnetic tracking (as in BoardLab [17]), computer vision, optical tracking, or mechanical linkage. To the best of our knowledge, a system tying these three components together has not yet been developed. Doing so would allow for the interactions proposed in this paper to be realized.

8 CONCLUSION

In this paper, we proposed Augmented Silkscreen, a set of augmented reality interaction techniques to assist electrical engineers in PCB debugging. We find that combining augmented visualization and augmented interaction on printed circuit boards unlocks promising avenues to alleviate the frequent context switching and spatial pattern matching exercises required by engineers' current ECAD tools. For experts, this can lead to more efficient debugging. In timed element selection tasks, this led to a 31% and 46% decrease in time to find a given component on the PCB and in the schematic respectively, with potential to decrease element localization more drastically in more complex board designs. For those unfamiliar with a PCB design or PCB design in general, the unambiguity of WYSIWYG augmentations on the board directly can help to make basic PCB workflows more accessible. While the bulk of the work done by the HCI community has focused on supporting the latter group, we hope this paper will inspire more work toward supporting hardware workflow challenges for both maker and expert populations.

REFERENCES

[1] [n.d.]. Allegro PCB Designer. https://www.cadence.com/en_US/home/tools/pcb-design-and-analysis/pcb-layout/allegro-pcb-designer.html

[2] [n.d.]. Altium Designer 20 - PCB Design Software. https://www.altium.com/altium-designer/

[3] [n.d.]. EAGLE | PCB Design And Electrical Schematic Software | Autodesk. https://www.autodesk.com/products/eagle/overview?plc=F360&term=1-YEAR&support=ADVANCED&quantity=1

[4] [n.d.]. EasyEDA - Online PCB design & circuit simulator. https://easyeda.com/

[5] [n.d.]. inspectAR Augmented Reality PCB Tools. https://www.inspectar.com/

[6] [n.d.]. KiCad EDA - Schematic Capture & PCB Design Software. https://kicad-pcb.org/

[7] [n.d.]. Mascot | Robotas. https://www.robotas.com/mascot/

[8] [n.d.]. PCB Design Software - PCB Design Tool - CADSTAR - Zuken US. https://www.zuken.com/us/product/cadstar/

[9] [n.d.]. PCB Design Software | OrCAD | Cadence. https://www.orcad.com/

[10] Ronald T. Azuma. 1997. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments* 6, 4 (8 1997), 355–385. https://doi.org/10.1162/pres.1997.6.4.355

[11] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (1 2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

[12] T.P. Caudell and D.W. Mizell. 1992. Augmented reality: an application of heads-up display technology to manual manufacturing processes. In *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*. IEEE, 659–669. https://doi.org/10.1109/HICSS.1992.183317

[13] Joshua Chan, Tarun Pondicherry, and Paulo Blikstein. 2013. LightUp. In *Proceedings of the 12th International Conference on Interaction Design and Children*. ACM, New York, NY, USA, 491–494. https://doi.org/10.1145/2485760.2485812

[14] Bettina Conradi, Verena Lerch, Martin Hommer, Robert Kowalski, Ioanna Vletsou, and Heinrich Hussmann. 2011. Flow of electrons. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '11*. ACM Press, New York, New York, USA, 182. https://doi.org/10.1145/2076354.2076389

[15] Daniel Drew, Julie L. Newcomb, William McGrath, Filip Maksimovic, David Mellis, and Björn Hartmann. 2016. The Toastboard. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 677–686. https://doi.org/10.1145/2984511.2984566

[16] Steven Feiner, Blair Macintyre, and Dorée Seligmann. 1993. Knowledge-based augmented reality. *Commun. ACM* 36, 7 (7 1993), 53–62. https://doi.org/10.1145/159544.159587

[17] Pragun Goyal, Harshit Agrawal, Joseph A. Paradiso, and Pattie Maes. 2013. BoardLab. In *Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology - UIST '13 Adjunct*. ACM Press, New York, New York, USA, 19–20. https://doi.org/10.1145/2508468.2514936

[18] Jürgen Hahn, Bernd Ludwig, and Christian Wolff. 2015. Augmented reality-based training of the PCB assembly process. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, Vol. 30-Novembe. ACM, New York, NY, USA, 395–399. https://doi.org/10.1145/2836041.2841215

[19] Yoonji Kim, Youngkyung Choi, Hyein Lee, Geehyuk Lee, and Andrea Bianchi. 2019. VirtualComponent. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300407

[20] David Mercer. 2019. *Internet of Things Now Numbers 22 Billion Devices But Where Is The Revenue?* Technical Report. Newsroom>Press Releases pages. https://news.strategyanalytics.com/press-release/iot-ecosystem/strategy-analytics-internet-things-now-numbers-22-billion-devices-where

[21] Oliver J. Muensterer, Martin Lacher, Christoph Zoeller, Matthew Bronstein, and Joachim Kübler. 2014. Google Glass in pediatric surgery: An exploratory study. *International Journal of Surgery* 12, 4 (4 2014), 281–289. https://doi.org/10.1016/j.ijsu.2014.02.003

[22] Yoichi Ochiai. 2014. Visible Breadboard: System for Dynamic, Programmable, and Tangible Circuit Prototyping with Visible Electricity. In *Virtual, Augmented and Mixed Reality. Applications of Virtual and Augmented Reality. VAMR 2014. Lecture Notes in Computer Science*. Vol. 8526 LNCS. Springer, Cham, 73–84. https://doi.org/10.1007/978-3-319-07464-1{_}7

[23] Bjorn Schwerdtfeger and Gudrun Klinker. 2008. Supporting order picking with Augmented Reality. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE, 91–94. https://doi.org/10.1109/ISMAR.2008.4637331

[24] Evan Strasnick, Maneesh Agrawala, and Sean Follmer. 2017. Scanalog: Interactive Design and Debugging of Analog Circuits with Programmable Hardware. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 321–330. https://doi.org/10.1145/3126594.3126618

[25] Evan Strasnick, Sean Follmer, and Maneesh Agrawala. 2019. Pinpoint. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–11. https://doi.org/10.1145/3290605.3300278

[26] Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. 2003. Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the conference on Human factors in computing systems - CHI '03*. ACM Press, New York, New York, USA, 73. https://doi.org/10.1145/642611.642626

[27] L. Vertelney. 1989. Using video to prototype user interfaces. *ACM SIGCHI Bulletin* 21, 2 (10 1989), 57–61. https://doi.org/10.1145/70609.70615

[28] Pierre Wellner. 1993. Interacting with paper on the DigitalDesk. *Commun. ACM* 36, 7 (7 1993), 87–96. https://doi.org/10.1145/159544.159630

[29] Te-Yen Wu, Hao-Ping Shen, Yu-Chian Wu, Yu-An Chen, Pin-Sung Ku, Ming-Wei Hsu, Jun-You Liu, Yu-Chih Lin, and Mike Y. Chen. 2017. CurrentViz. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 343–349. https://doi.org/10.1145/3126594.3126646