

Framing Creative Work: Helping Novices Frame Better Problems through Interactive Scaffolding

STEPHEN MACNEIL, Department of Cognitive Science, University of California, San Diego, USA

ZIJIAN DING, Department of Electrical and Computer Engineering, University of California, San Diego, USA

KEXIN QUAN, Department of Cognitive Science, University of California, San Diego, USA

THOMAS PARASHOS, Department of Computer Science, California State University, Northridge, California, USA

YAJIE SUN, Department of Computer Science and Engineering, University of California, San Diego, USA

STEVEN P. DOW, Department of Cognitive Science, University of California, San Diego, USA

Problem framing—the process of defining a problem—has been described by many researchers and designers as the crux of the design process. However, novice designers struggle with problem framing. To better understand this process and the potential for scaffolding, we conducted two studies. In the first study, we analyzed 41 problem statements from an introductory design course and found that novices often omit key information, like the primary stakeholder or the obstacles they face. To get novices to reflect on and include necessary design information, we created a tool called ProbLib that cues novices to explicitly reflect on aspects of the problem such as the stakeholders. To evaluate this approach, we conducted a between-subjects study (N=73) to compare ProbLib with an unstructured open text form. We found that participants using ProbLib wrote higher quality statements, included more information, and were more confident about specifying design needs. We observed creative behaviors such as brainstorming and analogical reasoning.

CCS Concepts: • **Human-centered computing** → **Empirical studies in HCI**.

Additional Key Words and Phrases: problem framing, novice design, scaffolding, problem statements, templates, decomposition

ACM Reference Format:

Stephen MacNeil, Zijian Ding, Kexin Quan, Thomas Parashos, Yajie Sun, and Steven P. Dow. 2021. Framing Creative Work: Helping Novices Frame Better Problems through Interactive Scaffolding. In *Proceedings of CC '21: ACM Conference on Creativity and Cognition (CC '21)*. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Discovering and framing important problems is one of the most important aspects of the design process [15, 37, 48]. A problem frame provides the designer with a way of thinking about the world. Frames provide an initial structure [26] and inform what kinds of solutions are possible [15]. The ability to frame challenging problems is the hallmark of an expert designer [12]. Experts also tend to spend significantly more time scoping and defining the problem than novices [1]. Despite its importance [15, 55], little is known about why problem framing is so hard or how designers learn to do it. This has led design researchers, like Jon Kolko, to refer to problem framing as the ‘magic’ part of the design process [37]. In this paper, we conduct two studies investigate the challenges novices face when framing problems and explore a technique for scaffolding this critical part of the design process in an attempt to demystify problem framing.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

Manuscript submitted to ACM

First, in a needfinding study, 88 introductory design students were provided instructions and a static template for writing a problem statement as part of a class project. We discovered that less than half of students (41.5%) used the provided template, few students (19.5%) described the central problem (obstacle), and that not all students (80.5%) mentioned the person affected by the problem (stakeholder). Students who did use the template included more information and also wrote 44.7% longer statements. However, even students who used the template still omitted key information such as the central problem (52.9%) and problem context (82.4%).

This first study showed that design students often omitted key design information; however, students who used the template included more information. The second study (n=73) introduced a web-based scaffold, ProbLib (inspired by the phrasal template word game, Mad Libs®, which focuses students' attention on key informational components and provides flexible templates to help students write problem statements. We found that the ProbLib system helped students articulate higher quality problem statements that featured more clearly defined stakeholders. ProbLib also led participants to feel more confident about identifying design needs. Participants also exhibited interesting behaviors around brainstorming, parallel prototyping [5, 16], and creatively combining [61] problem components to draw interesting connections [21] across problem components. Despite these benefits, we also identified challenges. Participants described instances of design fixation [31] and a desire for more flexible scaffolding.

2 RELATED WORK

Our work focuses on conceptual and strategic scaffolds to help novices understand the important parts of the problem and guide them toward generating problem statements that include relevant information. In domains like storytelling [32], citizen science [47], and crafting help request emails [29], templates can scaffold novices to perform more like experts. Building on this tradition of work, we explore templates and problem components to guide novice problem framing. By highlighting informational components that are key to design, our approach attempts to “make the tacit explicit” [49].

2.1 Problem framing in design

While it is important to acknowledge that design is seldom a linear process; problems are often considered the starting point for defining the design space and providing an initial structure to *ill-structured problems* [26]. This initial structuring and subsequent restructuring of the problem is known in design as *problem framing* [14, 26, 56]. In most design frameworks, problem framing happens early in the design process [59], though it is also well understood that problems and solutions co-evolve [11, 45] as problems are framed, solutions are developed, and problems are reframed [4]. Regardless of when problem framing occurs, researchers and practitioners have repeatedly emphasized the importance of problem framing in design [14, 15, 55].

Previous work has compared how novices and experts frame problems in naturalistic settings. Novices tend to spend almost half as much time scoping and defining the problem than experts [1, 30]. Novices also have a tendency to focus more on solutions than on problems [43, 44]. Cross et al. studied expert designers and concluded that the hallmark of an expert designer is their ability to frame challenging problems [12]. Despite research on the importance of problem framing [1, 12, 14, 30, 44], less is known about the specific challenges novices face. This work explores those challenges by analyzing how novices externalize their thinking through the use of provided scaffolding.

2.2 Supporting novices with scaffolding

Compared to novices, experts have domain specific knowledge and skills that help them more effectively address problems. Previous work has explored methods for modeling expertise in ways that are helpful for scaffolding novice

behavior [29, 32, 47]. Scaffolding—introduced by Wood, Bruner, and Ross [62]—is a method for providing support to novices in a way that can be removed when it is no longer needed [8, 52]. However, it can be challenging to effectively support *fading*, the process of gradually reducing scaffolding [27]. The idea of scaffolding is very related to Vygotsky’s conceptualization of Zones of Proximal Development [60], where an expert guides novices to develop new skills by building on the novice’s existing knowledge.

Functionally, scaffolding can operate at multiple levels to support motivation (or sustained engagement), meta-cognition (or self-reflection), conceptual understanding, and strategic planning related to process of the scaffolded activity [3, 28]. Multiple kinds of scaffolding can work in parallel to support a holistic learning experience; however, learning outcomes and task performance can be improved by providing even just one of these scaffolds [3]. Scaffolding can help novices perform more like experts through strategies like ‘marking critical features’, ‘reducing the degrees of freedom’, and ‘demonstrating’ [62]. Each of these strategies focus novices attention on the most important parts of the process and reduce opportunities for mistakes. Examples of scaffolding include procedural training and explanations [47, 54], templates [2, 13, 29, 63], expert patterns [32], checklists [20], examples [51], and process control [27]. Often, scaffolding is developed through observing and capturing the best practices of experts [32, 47, 51].

2.3 Decomposition in the design process

Decomposition is a technique for reducing complexity during problem solving. In engineering, designs can be decomposed around the product, process, or problem [41]. In product design, designs can be decomposed into the systems’ function, behaviors, and structure [23]. For crowds, tasks can be decomposed into sub-tasks [34]. In deliberation, arguments can be decomposed through issue-based information systems [9], such as deliberatorium [35]. Most of this work considers functional decomposition, where large problems are sub-divided into sub-problems [64], sub-arguments [35], and sub-designs [38] that can be addressed individually and then synthesized additively. These approaches make an assumption about the nature of problems. They assume that the challenge lies in the scale of the problem and not in the complexity. They also assume a rational approach to design that doesn’t fully account for decomposibility and complexity [6]. In this work, we consider decomposition from a structural perspective as a way to break down a thinking process to reflect on the structural parts of the problem that matter most. Here, the goal is not about breaking down problems to be solved, but about considering the problem holistically, brainstorming on the problem components and then reflecting on them to compose a problem statement.

3 NEEDFINDING STUDY: CONTENT ANALYSIS OF PROBLEM STATEMENTS IN A DESIGN COURSE

Despite the importance of problem framing [15, 37, 48], very little is known about how novices frame problems and develop problem statements. In this study, we leverage previous literature about scaffolding and templates to guide novice designers to write problem statements. Through this exploratory study, we intend to better understand: **how do novices use “How might we” templates to frame and articulate problem statements?** To investigate this question, we explore problem statements submitted by 88 students in a design course to understand what aspects students think about and articulate in their problem statements.

3.1 Study Context and Data Collection

To better understand how design novices frame and articulate problems we collected problem statements generated in a civic design course at the University of California-San Diego. Students (n=88) focused on public transportation and micro-mobility including sub-topics like *the last mile*, *safer roadways*, *equitable access to public transportation*,

and *autonomous vehicles*. Students were taught about problem framing and they follow an adapted design thinking model. The instructor provided students with the template: *How might we [make some change] so that [clearly defined stakeholder] can [goal state that they would like to achieve] in the face of [central problem that stands in the way]?* The data for this study was collected from an assignment where students were required to frame the problem they wanted to address in the course. We focus our analysis on their submitted problem statements. Students worked in pairs, except for two groups with three students and one group with four students. In total, 41 problem statements were submitted and included in our analysis. The course consisted primarily of fourth-year (n=68) and third-year (n=15) students.

3.2 Qualitative Coding Scheme

To make sense of the problem statements collected from the design course, we developed a coding scheme by thematically analyzing the problem statements and then we applied the coding scheme to every problem statement. We also evaluated whether participants used the template by coding whether or not the problems statements adhered to the template.

3.2.1 Coding for the use of the provided template. To understand whether students used the template, two coders coded the problem statements to see whether or not they used the template. The data was coded as a ‘yes’ if it adopted the template. To establish agreement, 20 problem statements were coded and the agreement level was 0.794 as measured by Cohen’s Kappa; indicating substantial agreement [42].

3.2.2 Developing and applying a coding scheme for problem statements. We developed a coding scheme inductively through a hybrid of open coding and in vivo coding [24, 53]. This resulted in four problem components: *obstacle*, *goal*, *stakeholder*, and *context*. *Obstacle* was the central problem faced by stakeholders or an undesired situation. *Goal* was a desired state or preferred outcome. *Stakeholder* were the people or group of people who are affected by or able to affect the problem outcome. *Context* described the setting (location or time) when a problem might be encountered or where a solution might be relevant. Although some of these components were present in the provided template and may have biased participants toward including them, these components also align with Studer’s four *problem elements* [58].

To apply the coding scheme, we segmented each problem statement into single units of analysis. Segmenting (i.e.: unitization) is the process of breaking data into smaller units to which a qualitative codes can be applied. We applied a single code to each segmented unit. Figure 1 shows three problem statements that have been segmented and coded. To evaluate our coding scheme, we introduced a new coder who was not involved in the initial thematic analysis. The new coder was shown only the coding scheme and three coded examples. The new coder and a previous coder coded all 41 problem statements and obtained inter-rater reliability of 0.63 as measured by Krippendorff’s α_u [40], which indicates moderate to substantial agreement [42]. Krippendorff’s α_u was used to account for agreement on both how the data is coded and on the bounds of the segment (i.e.: the start and stop characters) along a continuum.

P5	How might we [create better transit] for [lower income and homeless population] so that [they have equitable access to transportation in the areas they live] in [without being displaced]?	Legend: [obstacle]
P39	How can we [change our current road infrastructure] [in urban cities] so that [electric scooter users] can [ride safely] in the face of [dangerous riding environments]?	[goal] [stakeholder]
P23	How might we [redesign the common intersection] to [provide increased safety] for [all micro-mobility users]?	[context]

Fig. 1. The coding scheme applied to three problems statements collected from the civic design course that focused on public transportation and micro-mobility. We segmented and coded based on four key elements: obstacles, goals, stakeholders, and contexts.

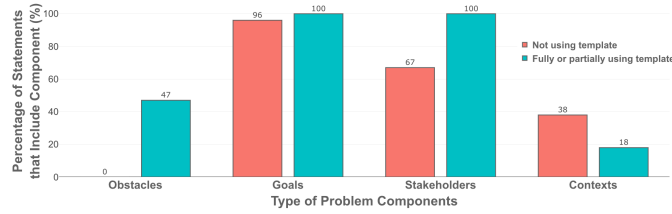


Fig. 2. The percentage of ‘design information’ included when using or not using the template.

3.3 Results

3.3.1 Less than half of the student teams appeared to use the template, those who did included more information. Based on the data coded in Section 3.2.1, we analyzed the extent to which participants used the template and whether using the template led participants to include more information. We found that only 41.5% of the problem statements adhered to the template. Of these problem statements that used the template, 10 (58.8%) modified it in some way. We find that across most of the problem components, using the template led participants to include more information as shown in Figure 2. For teams that used the template, 47.1% articulated the central problem; however, teams that didn’t use the template always failed to articulate the central problem. Similarly, every team that used the template included a stakeholder, whereas only 66.7% of teams that did not use the template included a stakeholder. Teams seemed to exclude key details about the context, regardless of whether they use the template or not. It generally appears that using the template increased the information that teams contributed; however, given that more than half of the teams that used the template still omitted the central obstacle and the problem context, more scaffolding may be required. Finally we observe that teams which used the template wrote 44.7% more words on average ($28.88 > 19.96$).

3.3.2 Statements emphasized some components more than others. Based on the coding scheme described in Section 3.2.2, we calculated the descriptive statistics and distributions for the codes (i.e.: obstacle, goal, stakeholder, and context) that appeared across the problem statements independent of whether the template was used. In our analysis, we found that almost every problem statement (40/41) explicitly included a goal to be achieved. However, few (19.5%) of the problem statements included a clearly articulated core obstacle that the problem intended to address. So while the goal was often clearly stated, it was often unclear what obstacles prevented the goal from being achieved directly. Other information was often missing, for example, only 29.3% of statements described the context of the problem and 80.5% explicitly specified a stakeholder. As an example, the following problem statement implies a stakeholder but don’t state them explicitly: “How might we raise awareness and update each other of poor road conditions in San Diego?”(P37). These results suggest that teams struggled to include the right design information and to specify that information clearly.

3.4 Discussion

This study analyzed the problem statements submitted in a design course to better understand students’ problem framing process. We found that students who used the provided template appeared to write longer statements and include more of the relevant design information identified in our thematic analysis and in previous typologies of problem components [58]. However, only 41.5% of the problem statements followed the provided template. Despite multiple cues, students tended to de-emphasize obstacles and focus primarily on goals. This aligns with previous studies that suggest design novices focus disproportionately on goals and solutions rather than on obstacles and problems [43, 44].

Additionally, despite the the course’s emphasis on human-centered design, where a clearly defined stakeholder is required to develop effective solutions [10, 39, 46], students who did not use the template were less likely to include a stakeholder. Not focusing on a clear stakeholder often resulted in vague problem statements that were aspirational but would be hard to achieve in practice. For example, “How might we shift the perspective around civic design to something more sexy like AI [Artificial Intelligence] or AVs [Autonomous Vehicles] and motivate the community to take action in improving their environment?” (P35).

In summary, more explicit guidance appears to be needed to help novice designers include the right information in their problem statements. Though templates and instruction cued students to think about these aspects, many students still omitted this essential information. Given that our analysis is limited to only what students externalize, the problem statement, it is unclear whether the challenge with problem framing is associated with a gap in a student’s thinking process or with an inability to articulate their thoughts as a problem statement. More work is needed to more explicitly decompose novices thinking processes and to externalize their thoughts in a way that can be observed.

4 PROBLIB: DECOMPOSING THE PROCESS OF FRAMING PROBLEMS

In our needfinding study, templates appeared to help students include more information, however less than half of the class used them. It was difficult to distinguish between challenges related to thinking about the components of a problem versus the thinking associated with composing the components together to form a problem statement. In this study, we further decompose the problem framing process by giving participants an area to think about each component type (i.e.: obstacles, goals, stakeholders, and contexts). By further decomposing the problem framing process, we encourage participants to externalize their thinking, which may provide individual benefits [36] and help us understand how participants think about the components and how they composing them as a problem statement. To deliver this scaffolding, we developed a web app which allowed us to carry out the study online and log user interactions.

ProbLib provides two scaffolds: problem components and templates. As shown in Figure 3, the problem components—*obstacle, goal, stakeholder, context*—are presented on the left and the problem templates can be filled in on the right. Placing components on the left was intended as a subtle encouragement to start by brainstorming the components before composing them into a problem statement. However, no specific ordering was enforced by the system. The problem components decompose users’ thinking process to consider each design aspect which is inspired by the attribute listing design method [50]. Thinking about each component individually also promotes selective focusing [57]. Whereas being able to see all of the components at once might foster creative combination [61]—where designer draw creative connections between elements. To write problem statements, users can select from drop down list for each component or skip the template by clicking on the pencil icon. The process of composing the problem components into a statement is a second opportunity for creative combination and reinterpretation. We explore two research questions:

RQ1 How does scaffolding for decomposition affect novices’ problem statement generation?

RQ2 How do novices engage with scaffolding designed to decompose the problem framing process?

4.1 Method and Experimental Context

To evaluate our research questions, we conducted a between-subjects experiment. Participants were randomly assigned into two conditions. In the experimental condition, participants used the ProbLib interface. In the control condition, participants used the same interface but with the problem components and templates removed. We applied a mixed-methods approach to analyze the problem components, problem statements, and responses to survey questions. Our

The screenshot displays the ProbLib interface, which is divided into two main sections: a left sidebar for adding components and a main area for writing problem statements.

Left Sidebar (a): This section contains four categories of components, each with a list of items and a '+ add new' button:

- Obstacles:** limited transportation, disease transmission
- Goals:** provide healthcare, limit physical contact
- Stakeholders:** low income communities, bus passengers
- Contexts:** when buses limit ridership, when working in food service, during rush hour

Main Area (b-d): Titled 'Your Problem Statements', this section shows how components are used to form statements.

- (b):** A template for a problem statement: 'How might we limit physical contact so that bus passengers can overcome disease transmission during rush hour?'. The components are selected from drop-down menus.
- (c):** The same statement as in (b) but with the text fields expanded into a single paragraph: 'How might we limit physical contact so that low income communities can overcome disease transmission when working in food service?'. A pencil icon indicates editing mode.
- (d):** The statement after editing, appearing as a paragraph: 'How might we provide healthcare so that low income communities can overcome limited transportation when buses limit ridership?'. A pencil icon indicates editing mode.

Fig. 3. ProbLib includes scaffolding for adding problem components (a) and for writing problem statements (b-d). a) Problem components highlight four important design considerations. Clicking *add new* creates a new component which can be edited directly as a text field. By default, (b) problem statements leverage a template with a drop-down menu to add components. (c) Clicking on the pencil icon allows users to edit the text directly. (d) After editing text, statements appear as a paragraph to improve readability.

quantitative analysis focused on the information that participants entered into the system (problem components and problem statements). Our qualitative analysis focused on analyzing the survey responses to get a better sense for participants' experiences, preferences, and challenges faced.

Participants spent 10-minutes framing problems related to COVID-19 using ProbLib or open text forms (Control). In both conditions, participants received identical training about problem components (e.g.: obstacles goals, stakeholders, and context) including a definition and examples. This training also included tips and examples for writing a good problem statement. Additionally, participants in the ProbLib condition reviewed a one-minute video about how to use the ProbLib interface. Participants in both conditions were provided with this task prompt:

The COVID-19 pandemic is causing many unprecedented problems. Many countries have locked down—closing stores and shops in an effort to slow the spread. Today, you can help by identifying key problems that you or others are facing in light of COVID-19. These problems can be directly related to COVID-19, such as lack of masks at hospitals, or related to the effects of COVID-19, such as unemployment rates skyrocketing. Go for quantity and quality!

4.1.1 Participants. We recruited 74 students from our university's participant recruitment platform. Students received course credit to participate. We asked participants whether they had taken a design class, whether they had a design portfolio, whether they had been paid to do design work, and whether they did design for a full-time job. 71.2% of participants disagreed with all of these statements, indicating a general lack of design experience in our sample. Participants were randomly assigned to the ProbLib or Control condition and there were no statistically significant differences between ratings on these questions between the two conditions in the pre-survey responses. One participant was removed from the Control condition because they did not complete the study, resulting in 73 participants.

4.1.2 Selecting a favorite problem statement. As part of our experiment procedures, we asked participant to select their favorite problem statement. Having participants select their one favorite problem statement made it easier to compare across participants. Using these favorite problem statements, we compared the word count, problem statement quality, and number of problem components across the two conditions. We additionally compared how many problem statements participants crafted in each study which included their favorite and non-favorite problem statements.

4.1.3 Measuring Quality of Problem Statements. To analyze the quality of the problem statements. We recruited two designers that were not affiliated with this project to rate each problem statement along a five-point scale (1=Low Quality, 5=High Quality). One of the designers was a TA in a design course. The other was a professional designer at medium-sized company. They coded the problem statement independently and had 0.56 agreement, which indicates moderate agreement [42]. This analysis was only applied to the favorite problem statements.

5 STUDY RESULTS

To better understand the problem statements, we investigated the lowest and highest rated statements. Examples of these problem statements are presented in Table 1. Through a thematic analysis we identified a number of pitfalls that were present in lower quality problem statements. These pitfalls were not including an obstacle (P28), being either too vague or specific (P9), having multiple focal points (P21), including irrelevant information (P54), conveying an obvious problem (P9), and containing grammatical errors or confusing language. As examples, P28 includes a solution or goal without specifying a problem that needs to be addressed. P9 includes the goal of ‘curing covid’ which is too vague and ambitious to be solved directly—the scope is too large. Similarly, many problem statements described vague stakeholder groups like ‘people’ or ‘everyone.’ P21 focuses on multiple focal points including mask mandates and the price of healthcare and medication. Without a singular focus, the problem would be very hard to address. P54 includes irrelevant or disconnected information. In their statement, it is not clear how school reopening and people’s trust in vaccines are related or which to prioritize. While these problem statements only illustrate a few examples of the pitfalls that we discovered in the statements, the analysis suggests that including or failing to include the problem components is only one part of a complex design activity.

Condition	PID	Problem Statement	Rating
Control	P28	Mandatory mask-wearing should be implemented.	1
Control	P21	Those who have underlying illnesses are at high risk and there needs to be and mask mandates, also, many cannot afford their own healthcare or medication as those prices have been skyrocketing throughout the pandemic.	1
Control	P57	Before the pandemic, many children relied on the provided school lunches to get a meal everyday. now that schooling is done online, how might we be able to provide food to children who come from a food insecure household?	4
ProbLib	P9	HMW cure covid so that small businesses can overcome financial issues covid-19?	1.5
ProbLib	P54	HMW lower the spread of covid-19 so that students can overcome people don’t trust vaccines when we need to open schools?	2
ProbLib	P73	HMW help people in low-income communities with no methods of transportation find a way to appropriate treatment centers?	4.5

Table 1. Sample problem statements from each condition. The quality rating (5=High Quality) represents the average rating given by two raters. ‘How might we’ was written as HMW to save space.

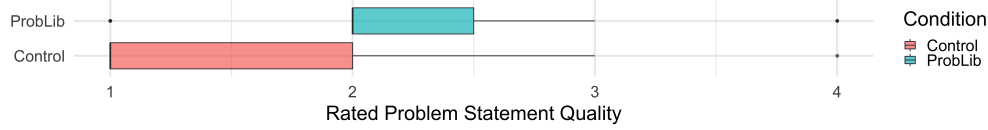


Fig. 4. A boxplot shows the distribution of problem statement ratings by condition, based on an expert designer’s ratings. Those who used ProbLib wrote higher quality statements compared to the Control participants who did not receive scaffolding.

5.1 RQ1: How does scaffolding for decomposition affect novices’ problem statement generation?

5.1.1 Decomposition improves problem statement quality over baseline. We analyzed the expert-rated problems statements that were selected by each participant as their favorite. Based on a Kruskal-Wallis test, we observed a statistically significant difference between the two conditions ($\chi^2 = 8.06$, $df = 1$, $p < 0.01$). Participants using ProbLib had higher quality problem statements ($M = 2.16$, $SD = 0.86$) than those who used an open-text form ($M = 1.6$, $SD = 0.85$) as shown in Figure 4. A Kruskal-Wallis test was used because quality scores were ordinal and not normally distributed.

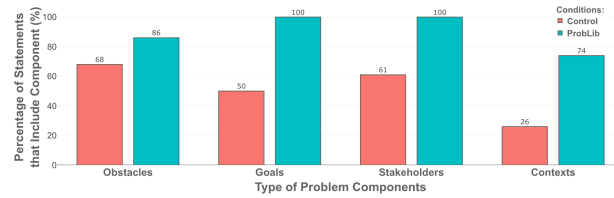


Fig. 5. Participants using ProbLib included more components than those using an open text form.

5.1.2 Decomposition cued participants to include more components and more specific stakeholders. To understand whether ProbLib affected which problem components participants included, we qualitatively coded the problem statements as previously described in Section 3.2.2. We observed that participants using ProbLib tended to include more of the problem components as summarized in Figure 5. Due to the importance of stakeholders in human-centered design, we further analyzed whether and how participants specified a stakeholder. Participants using ProbLib were more likely to include a stakeholder and provided more specific stakeholders as measured by a Chi-Square test ($\chi^2 = 17.57$, $df = 2$, $p < 0.01$). Pairwise post-hoc comparisons with Bonferroni correction showed two significant pairwise differences, summarized in Figure 6.



Fig. 6. A barchart shows that ProbLib not only helped participants remember to include a stakeholder, but also helped them articulate more specific stakeholders. * indicates statistically significant post-hoc comparisons.

5.1.3 *There was no difference in word count or the number of problem statements between conditions.* There was no difference in the number of words used in each problem statement as measured by a Kruskal Wallance test ($\chi^2 = 1.33$, $df = 1$, $p = 0.25$). There was also no difference between the number of problem statements created in either condition as measured by an ANOVA ($F = 0.79$, $df = 1$, $p = 0.37$).

5.1.4 *Decomposition increased confidence about designing solutions and identifying needs.* To analyze participants' changes in confidence from the pre-survey to the post-survey, we used a Mann-Whitney test for paired-samples. We observed two statistically significant increases for participants using ProbLib for the statements: "I feel comfortable identifying design needs" $U = 447.0$, $p < 0.05$; "I feel comfortable developing design solutions" ($U = 458.0$, $p < 0.05$). However, confidence related to the second statement also increased in the control condition ($U = 568.0$, $p < 0.05$). When comparing conditions, confidence in identifying design needs only increased for participants using ProbLib.

5.2 RQ2: How do novices engage with scaffolding designed to decompose the problem framing process?

To understand how participants used the scaffolding introduced by ProbLib, we analyzed participants' survey responses.

5.2.1 *Scaffolding around decomposition led novices to explore alternatives.* Many participants described how the components spurred creativity and broader divergent thinking. Participants described the process as "... a really good brainstorm session" (P15) and that "it was efficient and allowed room for creativity" (P28). Participants described how the experience helped them think broadly. P6 describes this broader thinking, saying "it also allowed me to more clearly look at the wide array of individuals and groups who can be affected by a particular action." P16 also described this broader thought process, "...it allowed me to construct my thoughts and consider all the things affected." These participants appear to be brainstorming components of the problem. This broader perspective included thinking not only about problems, but about problems and solutions together. P32 shared that the scaffolding "allows you to think about each part of the problem and solution." While most participants were able to come up with alternatives, some participants may benefit from additional inspiration. Over half of the participants agreed (15) or strongly agreed (8) with the statement, "I ran out of ideas for problems by the end of the 10 minutes." Nine participants disagreed with the statement and three strongly disagreed. A few participants described running out of ideas. As an example, P35 said "I think the challenge is just that I could not come up with more ideas." Others, like P22, described only having trouble brainstorming specific component types, "I found it very difficult since I couldn't come up with obstacles or contexts." As a solution, P4 suggested that "some examples would be a inspiration."

5.2.2 *Brainstorming problem components enabled creative combination and recombination.* In the previous analysis, we observed that participants brainstormed multiple options for each component type, but we also observed that participants came up with more problem statements than each component type. One explanation is that participants reused and recombined problem components. As an example, one participant that submitted 18 problem statements did so by combining and reusing their components. They only used 4 distinct stakeholders to compose these 18 problem statements. Other participants also described making connections across components:

"By writing down the components, I am layering the 'dots' that can be connected together to provide a full picture of the problem to be solved." (P42)

Another participant talked about making connections and reflecting critically: "This tool was extremely easy to acclimate... I believe that this tool helped me make connections and forced me to think critically about the nature of our world in its current state." (P19). P16 shared this sentiment, citing the ability to "... see the connections between everything

involved” as a primary benefit of the scaffolding. When asked about specific strategies employed, participants described making connections as a specific strategy with P15 saying “I try to match categories together.” Finally, in addition to making connections, the scaffolding also helped identify gaps:

“It made it easier because once I was able to identify one component I could start filling in the blanks for the other components. It helped me to connect my ideas and make them into statements that worked. It facilitated the process by allowing me to brainstorm ... If something was missing, it made me easier for me to also come up with that missing component.” (P11)

To analyze how many participants reused problem components, we analyzed participants problem statements and components. If a component was contained in multiple statements, we annotated it with a tag. We then divided the components with tags by all components (with and without tags). We found that overall, 26.3% of 407 problem components submitted by participants were used more than once by the same participant. Goals were the most reused component with 32.4% of goals being used multiple times. Context was the next most reused component with 28.8% of contexts being reused. 22.8% of stakeholders and 22.5% of obstacles were also repeated. While reuse and combination appears to be associated with making connections, this reuse may also be evidence that some participants ran out of ideas and reused components to reduce the cognitive load associated with generating new ideas. If so, this reiterates the need for inspiration expressed by participants in Section 5.2.1.

6 DISCUSSION AND DESIGN IMPLICATIONS

In this paper, novice designers used a variety of scaffolding to help them structurally decompose problems during the problem framing process. The goal was both to understand the potential for supporting novices with interactive scaffolding and to learn more about how novices frame problems. Study 2 introduced a decomposition approach aimed at making “the tacit explicit” [49]. This scaffolding encouraged participants to externalize [33, 36] their thoughts about the problem components before composing problem statements. Decomposition helped participants selectively focus their attention [57]. It also gave us unique insights into their thinking process and the components they considered when writing problem statements. Below we revisit our research questions and consider implications for future scaffolding.

Our analysis showed that the scaffolding provided by ProbLib for decomposing the problem framing process improved the quality of problem statements and increased participants confidence in being able to identify design needs. We also observed that participants were more likely to include necessary design information. When articulating stakeholders, participants tended to be more specific compared to just using an open text form. Participants cited a range of benefits associated with scaffolding decomposition: it helped them brainstorm, make connections [61], reflect [55], and think more broadly about problem framing. Participants were able to identify ‘gaps’ in their thinking and make ‘connections’ across and within the problem components they generated. Most participants brainstormed multiple alternatives for each component type before generating problem statements, which demonstrates a kind of parallel thinking shown to be useful when prototyping [5, 16]. Participants also reused and recycled components by mixing and matching, potentially reducing cognitive load while creating opportunities creative combination [61].

Despite these successes, participants also faced some challenges. Similar to Study 1, participants had difficulty coming up with components at the right level of abstraction, often biasing toward components that were either too specific or too vague. Some participants included lists of conceptually similar stakeholders instead of focusing on a single one. This kind of grouping may be an instance of *chunking* [17, 25] where designers group conceptually related items into a prototype [22]. Additional support may be needed to coach novices toward arriving at the right level of abstraction for

each component through *heuristic feedback*. In Study 2, some participants commented that the templates were overly restrictive which reiterates findings from non-design contexts [29]. Here we see an opportunity to develop more flexible methods of interacting with templates through *adaptive templates* that can adapt to support flexibility.

6.1 Design Implications: Providing Support for Process and Synthesis

Our work explored how scaffolding can support individuals, but only scratched the surface on how it might dynamically adapt to the needs of individuals. Riffing on Buxton’s mantra [5], the scaffolding appears to help novices include the *right information*, but additional coaching could help them to get the *information right*. One option would be to coach novices with feedback based on text analyses of problem components. For instance, lists can be detected by identifying commas and conjunctions. Detecting lists could prompt the system to recommend a template with multiple spaces for stakeholders or cue novices to hone in on a single stakeholder. Currently our approach partially follows the cognitive apprenticeship model [8] by providing opportunities for modeling (problem components), reflection across components, scaffolding for decomposing and composing problem statements. By introducing coaching, our approach could deliver on all of the features of the cognitive apprenticeship model. Finally, static scaffolding in the form of templates was too restrictive for some. Prior work recommends gradually reducing scaffolding through *fading* [52], but this is challenging in practice [27]. We could imagine intelligent templates that adapt to provide benefits for more advanced novices. As an example, Google’s Smart Compose [7] mines previous emails and the current context to infer intent and make autocomplete suggestions. A similar approach could analyze the statement—as it is being written—to suggest similar templates and to autocomplete with problem components.

7 CONCLUSION AND LIMITATIONS

In this paper, we conducted two studies to better understand novices’ experiences with scaffolding designed to help with problem framing. Our first study identified four important components of a problem, aligning with prior work [58]. We observed that most students did not use the provided template and consequently omitted important information, like stakeholders and the central problem to be addressed. Using the template appeared to help, but even those who used the template often missed information. This inspired us to provide more explicit scaffolding.

We introduced ProbLib, a system for explicitly breaking down considerations for framing problems. In an evaluation study of ProbLib, we found that this scaffolding improved problem statement quality and participants’ confidence related to identifying design needs. Participants also included more information and at a more specific level of abstraction. Qualitatively, we observed interesting behaviors around brainstorming, parallel prototyping [5, 16], and creatively combining [61] problem components to draw interesting connections [21] within the problem space. These behaviors have typically been observed and studied as they relate to solutions, but it was interesting to see participants engaging in these behaviors around problems too. Creativity support tools tend to focus on ideation and overlook problem finding and problem framing [19]. ProbLib fills this gap by encouraging creative thinking during the problem framing process.

The goal of this study was to shed light on novices’ thought processes by providing affordances that encouraged externalization [33, 36]. However, these scaffolds also affected how participants thought, as cognitive tools tend to do [18]. Similarly, our experiment does not fully replicate naturalistic settings where problem framing can take hours, weeks, months, or even years [15]. In our setting, novices were asked to frame problems on-demand as part of a 10-minute task. Despite these limitations, this study captures important challenges that novices appear to face, replicates prior work related to identifying the components of a problem [58] using templates [29], and identifies interesting opportunities for creative combination and brainstorming [61].

REFERENCES

- [1] Cynthia J Atman, Robin S Adams, Monica E Cardella, Jennifer Turns, Susan Mosborg, and Jason Saleem. 2007. Engineering design processes: A comparison of students and expert practitioners. *Journal of engineering education* 96, 4 (2007), 359–379.
- [2] Jared S Bauer and Julie A Kientz. 2013. Designlibs: A scenario-based design method for ideation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1955–1958.
- [3] Brian R Belland. 2017. Computer-based scaffolding strategy. In *Instructional Scaffolding in STEM Education*. Springer, 107–126.
- [4] Richard Blyth, Nicole Schadewitz, Helen Sharp, Mark Woodroffe, Dino Rajah, and Ranganai Turugare. 2012. A frame signature matrix for analysing and comparing interaction design behaviour. (2012).
- [5] Bill Buxton. 2010. *Sketching user experiences: getting the design right and the right design*. Morgan kaufmann.
- [6] Li Chen and Simon Li. 2005. Analysis of decomposability and complexity for design problems in the context of decomposition. (2005).
- [7] Mia Xu Chen, Benjamin N Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M Dai, Zhifeng Chen, et al. 2019. Gmail smart compose: Real-time assisted writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2287–2295.
- [8] Allan Collins, John Seely Brown, and Susan E Newman. 1988. Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics. *Thinking: The Journal of Philosophy for Children* 8, 1 (1988), 2–10.
- [9] Jeff Conklin and Michael L Begeman. 1987. gIBIS: A hypertext tool for team design deliberation. In *Proceedings of the ACM conference on Hypertext*. 247–251.
- [10] Mike Cooley. 2000. Human-centered design. *Information design* (2000), 59–81.
- [11] Nigel Cross. 2001. Designly ways of knowing: Design discipline versus design science. *Design issues* 17, 3 (2001), 49–55.
- [12] Nigel Cross and Anita Clayburn Cross. 1998. Expertise in engineering design. *Research in engineering design* 10, 3 (1998), 141–149.
- [13] Ryan Deschamps, Nick Ruest, Jimmy Lin, Samantha Fritz, and Ian Milligan. 2019. The archives unleashed notebook: Madlibs for jumpstarting scholarly exploration of web archives. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, 337–338.
- [14] Kees Dorst. 2011. The core of ‘design thinking’ and its application. *Design studies* 32, 6 (2011), 521–532.
- [15] Kees Dorst. 2015. *Frame innovation: Create new thinking by design*. MIT press.
- [16] Steven P Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L Schwartz, and Scott R Klemmer. 2010. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Transactions on Computer-Human Interaction (TOCHI)* 17, 4 (2010), 1–24.
- [17] Dennis E Egan and Barry J Schwartz. 1979. Chunking in recall of symbolic drawings. *Memory & Cognition* 7, 2 (1979), 149–158.
- [18] Kieran Egan. 1997. *The educated mind: How cognitive tools shape our understanding*. University of Chicago Press.
- [19] Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. 2019. Mapping the landscape of creativity support tools in HCI. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–18.
- [20] Atul Gawande. 2010. *Checklist manifesto, the (HB)*. Penguin Books India.
- [21] Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science* 7, 2 (1983), 155–170.
- [22] John Gero, Mary Lou Maher, and Weiguang Zhang. 1989. Chunking structural design knowledge as prototypes. In *Design Theory’88*. Springer, 155–155.
- [23] John S Gero. 1990. Design prototypes: a knowledge representation schema for design. *AI magazine* 11, 4 (1990), 26–26.
- [24] Barney G Glaser. 1992. Basics of grounded theory analysis. Mill Valley.
- [25] Fernand Gobet, Peter CR Lane, Steve Croker, Peter CH Cheng, Gary Jones, Iain Oliver, and Julian M Pine. 2001. Chunking mechanisms in human learning. *Trends in cognitive sciences* 5, 6 (2001), 236–243.
- [26] Vinod Goel and Peter Pirolli. 1992. The structure of design problem spaces. *Cognitive science* 16, 3 (1992), 395–429.
- [27] Mark Guzdial. 1994. Software-realized scaffolding to facilitate programming for science learning. *Interactive learning environments* 4, 1 (1994), 001–044.
- [28] Michael Hannafin, Susan Land, and Kevin Oliver. 1999. Open learning environments: Foundations, methods, and models. *Instructional-design theories and models: A new paradigm of instructional theory* 2 (1999), 115–140.
- [29] Julie S Hui, Darren Gergle, and Elizabeth M Gerber. 2018. IntroAssist: A Tool to Support Writing Introductory Help Requests. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [30] Vikas K Jain and Durward K Sobek. 2006. Linking design process to customer satisfaction through virtual design of experiments. *Research in Engineering Design* 17, 2 (2006), 59–71.
- [31] David G Jansson and Steven M Smith. 1991. Design fixation. *Design studies* 12, 1 (1991), 3–11.
- [32] Joy Kim, Mira Dontcheva, Wilmot Li, Michael S Bernstein, and Daniela Steinsapir. 2015. Motif: Supporting novice creativity through expert patterns. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 1211–1220.
- [33] David Kirsh. 2009. Interaction, external representation and sense making. (2009).
- [34] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. 2011. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 43–52.
- [35] Mark Klein. 2011. The MIT deliberatorium: Enabling large-scale deliberation about complex systemic problems. In *2011 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, 161–161.

- [36] Jon Kolko. 2007. Information architecture and design strategy: The importance of synthesis during the process of design. In *Industrial Designers Society of America Conf.*
- [37] Jon Kolko. 2011. *Exposing the magic of design: A practitioner's guide to the methods and theory of synthesis*. Oxford University Press.
- [38] Philip J Koopman Jr. 1999. A taxonomy of decomposition strategies based on structures, behaviors, and goals. In *Design Theory & Methodology Conference*.
- [39] Klaus Krippendorff. 2006. The semantic turn: A new foundation for design. *ARTIFACT-ROUTLEDGE*- 1, 11 (2006), 51.
- [40] Klaus Krippendorff, Yann Mathet, Stéphane Bouvry, and Antoine Widlöcher. 2016. On the reliability of unitizing textual continua: Further developments. *Quality & Quantity* 50, 6 (2016), 2347–2364.
- [41] Andrew Kusiak and Nick Larson. 1995. Decomposition and representation methods in mechanical design. (1995).
- [42] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.
- [43] Stephen MacNeil, Sarah Abdellahi, Mary Lou Maher, Jin Goog Kim, Mohammad Mahzoon, and Kazjon Grace. 2018. Designing with and for the Crowd: A Cognitive Study of Design Processes in NatureNet. In *International Conference on Design Computing and Cognition*. Springer, 55–73.
- [44] Stephen MacNeil, Zijian Ding, Ashley Boone, Anthony Bryce Grubbs, and Steven P. Dow. 2021. Finding Place in a Design Space: Challenges for Supporting Community Design Efforts at Scale. (2021).
- [45] Mary Lou Maher, Josiah Poon, and Sylvie Boulanger. 1996. Formalising design exploration as co-evolution. In *Advances in formal design methods for CAD*. Springer, 3–30.
- [46] Don Norman. 2013. *The design of everyday things: Revised and expanded edition*. Basic books.
- [47] Vineet Pandey, Justine Debilius, Embriette R Hyde, Tomasz Kosciolk, Rob Knight, and Scott Klemmer. 2018. Docent: transforming personal intuitions to scientific hypotheses through content learning and process training. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*. 1–10.
- [48] Bec Paton and Kees Dorst. 2011. Briefing and reframing: A situated practice. *Design Studies* 32, 6 (2011), 573–587.
- [49] Chris Quintana, Brian J Reiser, Elizabeth A Davis, Joseph Krajcik, Eric Fretz, Ravit Golan Duncan, Eleni Kyza, Daniel Edelson, and Elliot Soloway. 2004. A scaffolding design framework for software to support science inquiry. *The journal of the learning sciences* 13, 3 (2004), 337–386.
- [50] Tudor Rickards. 1974. *Problem-solving through creative analysis*. Wiley.
- [51] Michael A Ringenberg and Kurt VanLehn. 2006. Scaffolding problem solving with annotated, worked-out examples to promote deep learning. In *International Conference on Intelligent Tutoring Systems*. Springer, 625–634.
- [52] Barbara Rogoff. 1990. *Apprenticeship in thinking: Cognitive development in social context*. Oxford university press.
- [53] Johnny Saldaña. 2015. *The coding manual for qualitative researchers*. Sage.
- [54] William A Sandoval and Brian J Reiser. 2004. Explanation-driven inquiry: Integrating conceptual and epistemic scaffolds for scientific inquiry. *Science education* 88, 3 (2004), 345–372.
- [55] Donald A Schön. 1984. *The reflective practitioner: How professionals think in action*. Routledge.
- [56] Donald A Schön. 1988. Designing: Rules, types and worlds. *Design studies* 9, 3 (1988), 181–190.
- [57] Fremont A Shull, Andre L Delbecq, and Larry L Cummings. 1970. Organizational decision making. (1970).
- [58] Jaryn A Studer, Shanna R Daly, Seda McKilligan, and Colleen M Seifert. 2018. Evidence of problem exploration in creative designs. *AI EDAM* 32, 4 (2018), 415–430.
- [59] Vanessa Svihla and Richard Reeve. 2016. Facilitating problem framing in project-based learning. *Interdisciplinary Journal of Problem-Based Learning* 10, 2 (2016).
- [60] Lev Vygotsky. 1978. Interaction between learning and development. *Readings on the development of children* 23, 3 (1978), 34–41.
- [61] Merryll J Wilkenfeld and Thomas B Ward. 2001. Similarity and emergence in conceptual combination. *Journal of Memory and Language* 45, 1 (2001), 21–38.
- [62] David Wood, Jerome S Bruner, and Gail Ross. 1976. The role of tutoring in problem solving. *Journal of child psychology and psychiatry* 17, 2 (1976), 89–100.
- [63] Licheng Yu, Eunbyung Park, Alexander C Berg, and Tamara L Berg. 2015. Visual madlibs: Fill in the blank description generation and question answering. In *Proceedings of the IEEE international conference on computer vision*. 2461–2469.
- [64] Haoqi Zhang, Eric Horvitz, Rob C Miller, and David C Parkes. 2011. Crowdsourcing general computation. (2011).