APPENDICES

A IMPLEMENTATION OF THE LLM-BASED APPROACHES

We designed prompts and user interfaces to implement the DIRECT and STRUCTURED approaches. For our study, it was crucial that we test our hypotheses with the highest-quality problem frames and intermediate content that LLMs can generate. To this end, we iteratively designed prompts informed by guidelines for effective prompting [14, 17, 74, 83]. The main practice involved was to give the LLMs clear, concise task instructions while making sure to break complex tasks into smaller sequential subtasks that can reap the most from LLMs' reasoning competence. Accordingly, we set up a system role that describes problem reframing and upcoming tasks (see Table 3), then tested our prompts by using the model that performed best at the time of the research (GPT-4o).

Another important factor we considered is how to guide users to interact with the LLMs as intended under each approach. While users converse freely with the agent in the FREE-FORM setting, the DIRECT and STRUCTURED ones should guide users to employ particular methods with the models. Therefore, we designed a user interface wherein users click buttons to prompt the LLMs in each step of problem reframing. This button-based interaction has the advantage of preventing users from putting the LLMs to different purposes; guaranteeing a similar experience across participants who differed in their prompting skills let us accurately assess the potential of each approach relative to others. Below, we introduce the prompts and system design of the DIRECT and the STRUCTURED approach.

Table 3: The system prompt shared by the DIRECT and the STRUCTURED approach

Role	Content
System	You are an expert designer performing "problem reframing". Problem reframing is an essential process of solving design problems, where designers explore alternative ways of defining initial problems that can lead to creative and actionable solutions. For this, you will be given descriptions related to problems and a task for reframing the problem.

A.1 The Direct Approach

The core idea behind the DIRECT approach is that designers interact with LLM-generated problem frames only (Figure 11 presents this principle). To enable that, we designed an interface in which users click a button to prompt LLMs to generate alternative problem frames. In this interface, presented in Figure 12, users read a problem description, press the button to (re)generate alternative problem frames, and build on the LLM-generated frames. To make the process iterative, we designed two prompts each comprising input, task, and output components (shown in Table 4). The first prompt is used to generate the initial set of frames on the basis of the problem description alone. This prompt includes a problem description, a reframing task, and a format for describing problem frames (following Dorst's frame template [19]). The second prompt is used iteratively. Here, the previous frames function as input to instructing LLMs to generate different frames.

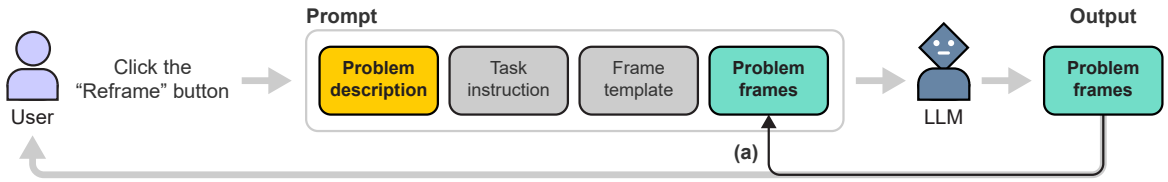

Figure 11: The DIRECT approach employs LLM-generated problem frames to yield a broader array of problem frames in the next run (a).

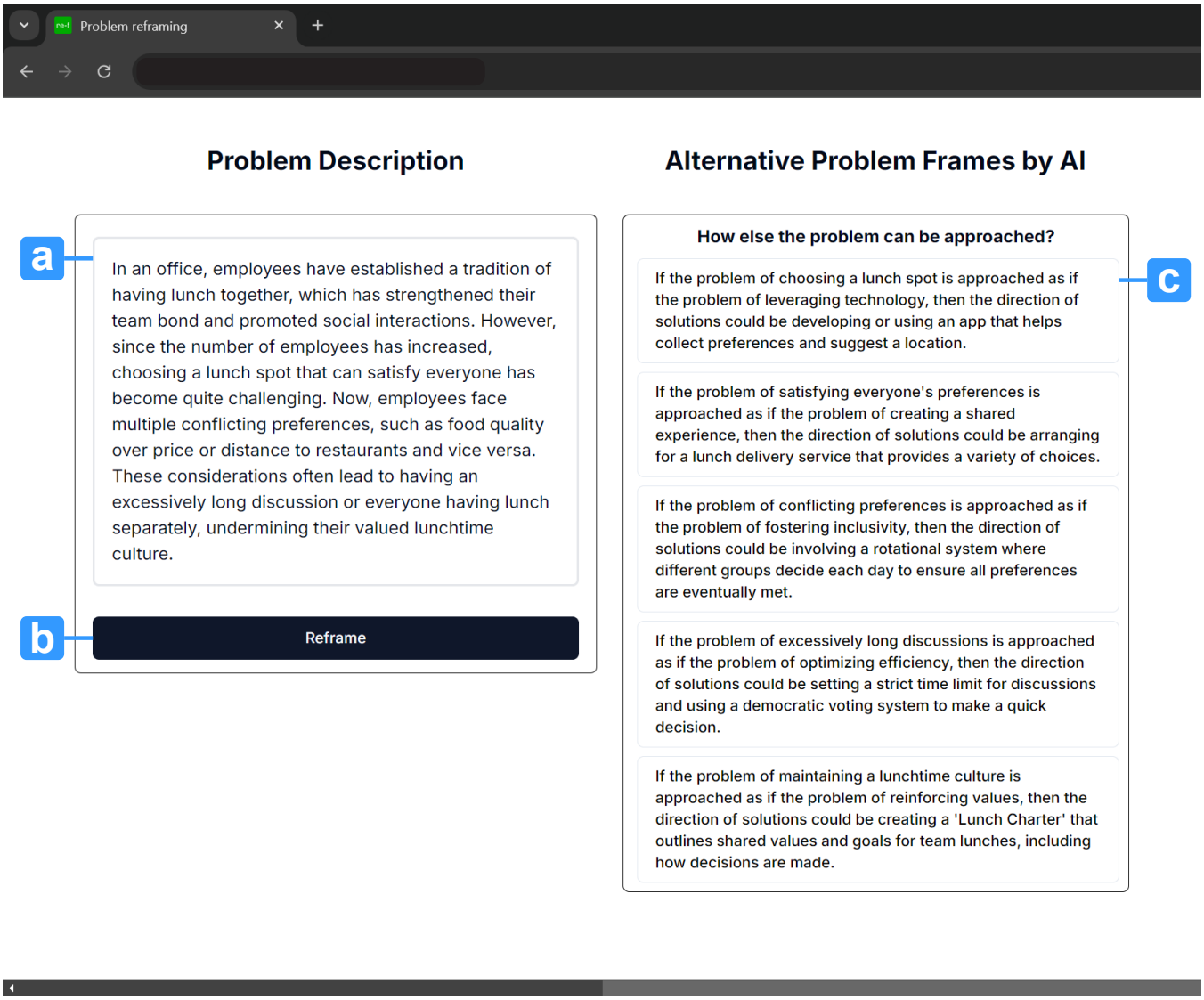

Figure 12: The interface for the DIRECT approach. On the left is the description of the problem that users need to reframe (a). Pressing the "Reframe" button (b) causes the LLMs to generate alternative problem frames, presented in one frame per row (c).

Table 4: Prompts for the DIRECT approach

Generating initial frames	Here is a description of a problem situation: `{problem description}` Generate alternative ways of approaching the problem. Show only the result in the following format: `{problem frame template}`
Generating alternative frames	Here is a description of a problem situation: `{problem description}` Here are alternative ways of approaching the problem: `{previous problem frames}` Compared to the approaches above, generate different ways of approaching the problem. Show only the result in the following format: `{problem frame template}`

A.2 The Structured Approach

The STRUCTURED approach enables using LLMs in a step-by-step reframing process (here, we follow Dorst's nine-step reframing process [19]). Following this approach, designers can use LLMs to generate not only problem frames but also content in each reframing step. When compared to merely seeing alternative frames, this technique might show value in helping designers deepen their understanding of the problem and refine their thought process. Accordingly, we designed a system wherein LLMs take the content generated in the previous step as input for the next one (this is presented in Figure 13). We implemented an interface wherein LLM-generated contents are shown in each window, grouped by similarity of purpose in problem reframing (see Figure 14). In a similarity to the DIRECT approach, users can click the "Reframe" button to prompt the LLMs to go through the entire set of steps or use a refresh button offered in each content window to prompt the LLMs to generate local content. We designed step-specific prompts for this as shown in tables 5, 6, 7, and 8. Overall, we designed these to enable a prompt-chaining technique [84], whereby the outputs can be taken as input to the next prompt. Our design decisions on the prompts are shown in the corresponding tables.

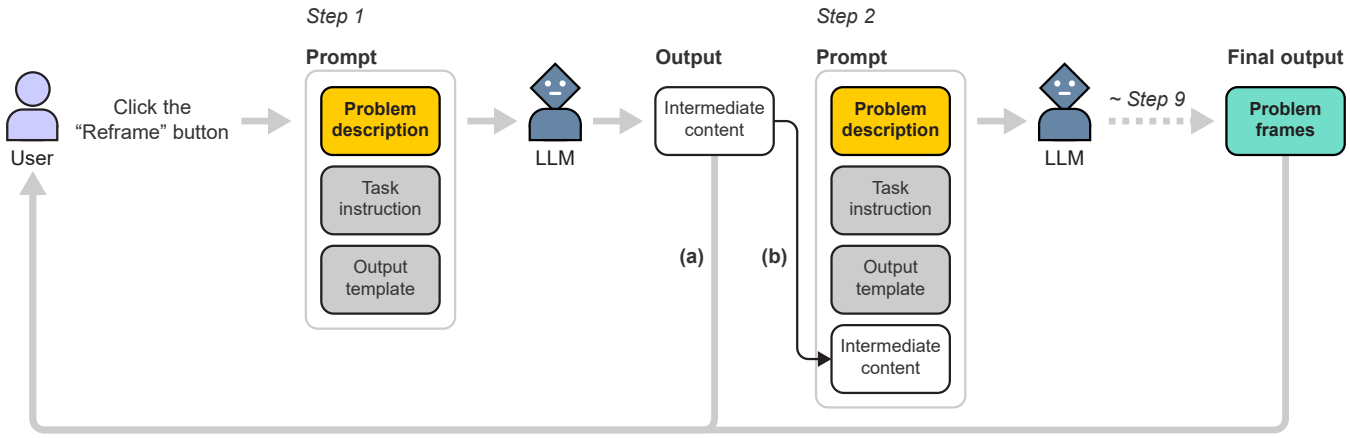

Figure 13: In the STRUCTURED approach, LLMs generate alternative frames by following Dorst's nine-step reframing process. Throughout the process, users can review and generate new specific content (a), which is used to create the content in the following steps (b).

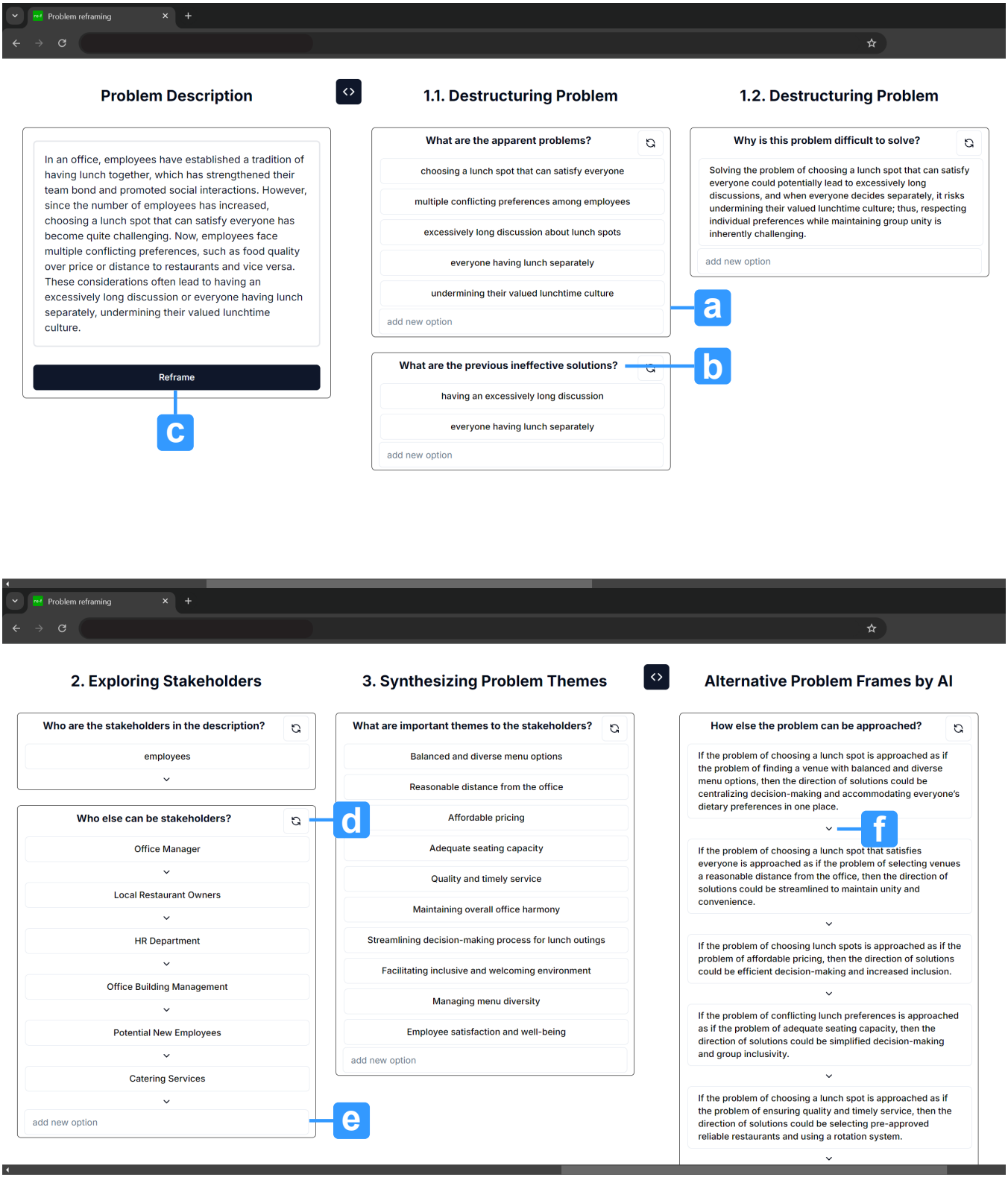

Figure 14: The interface for the STRUCTURED approach. The leftmost of its six columns displays a problem description while the others show LLM-generated content in every successive window (a) to represent each step in Dorst's reframing process, in turn. So that the nature of the content is clear at all times, the windows display the key points in each step (b). Users can click "Reframe" to prompt the LLMs to run through the entire set of steps (c). They can also click the re-generation button, to explore alternative content within each step (d), or add their own (e). To prevent information overload, the interface keeps some content (f), described in Figure 15, hidden by default.

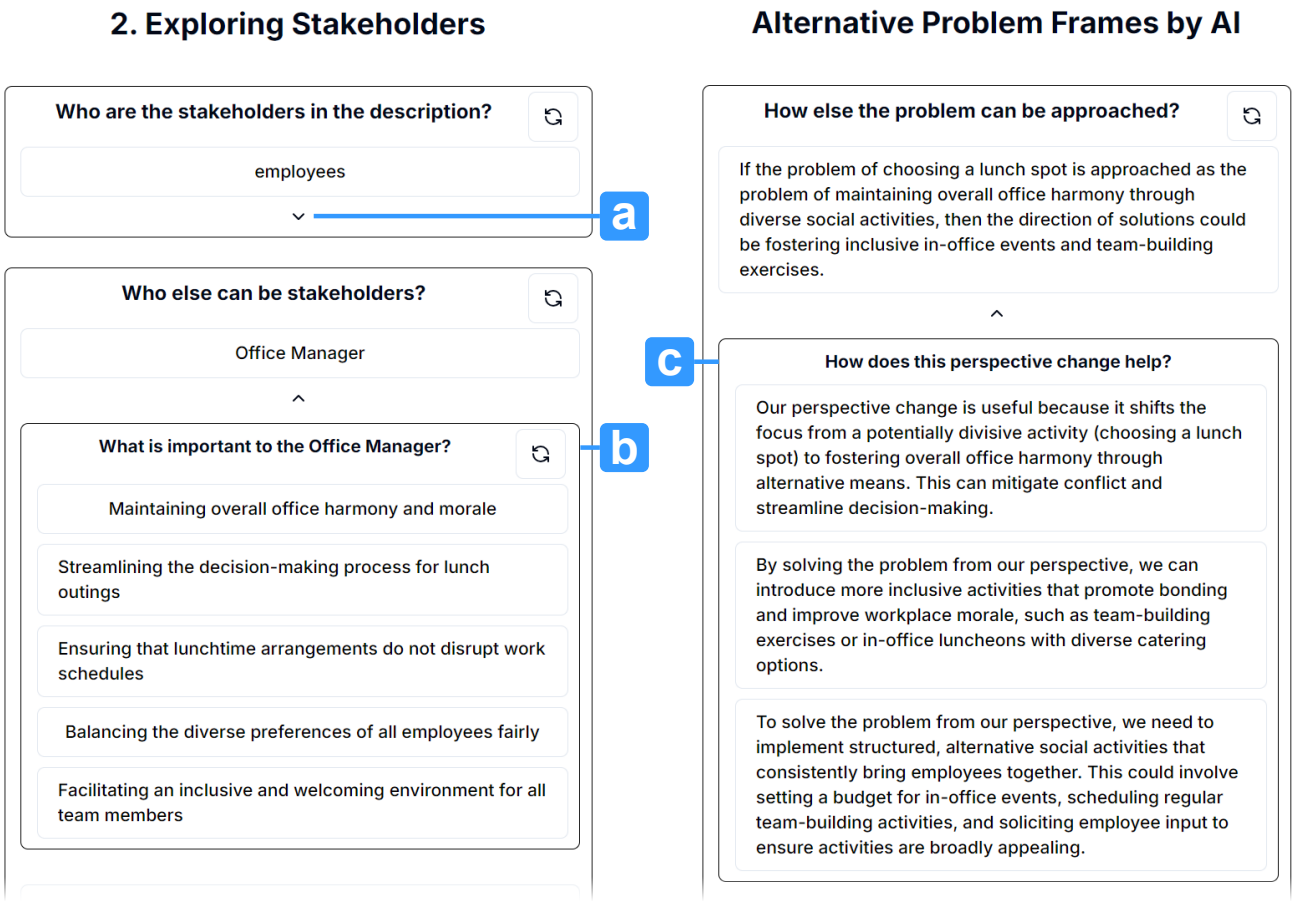

Figure 15: The steps related to the stakeholders and problem frames have hidden windows that fill out the Dorst process. Users can reveal them by clicking the down-arrow button (a). For the former, the window shows what each stakeholder finds important in the problem situation (b). For each problem frame, the window describes how a shift in perspective to the new frame might help address the problem (c); this covers the frame's utility, additional benefits that the frame could bring, and concomitant changes that are feasible.

Table 5: Characterization of the prompts in the STRUCTURED approach for destructuring of problems. Step 1 and 2's prompts focus on retrieving factual information from the problem description. While GPT-4o was able to list apparent problems without additional guidance, it frequently generated ineffective solutions when the problem description lacked specific solutions, so we instructed GPT-4o to respond with "No previous solutions" in these cases. In step 3, GPT-4o struggled to grasp the concept of paradoxical problems at first and often produced a verbose summary of the initial problem. For correction, we introduced a concise explanation of paradoxical problems based on Dorst's work and set a limit to the output length to assure of clarity.

Step 1: Apparent problems	Here is a description of a problem situation: `{problem description}` List apparent problems that are described in the description. Show only the result in the following format: `{output template}`
Step 2: Ineffective solutions	Here is a description of a problem situation: `{problem description}` From the description, list previous solutions that did not work. If there are no previous solutions, reply ["No previous solutions."]. Show only the result in the following format: `{output template}`
Step 3: Paradoxical problems	Here is a description of a situation: `{problem description}` Here are apparent problems: `{apparent problems}` Here are previous ineffective solutions: `{ineffective solutions}` What other problems arise by solving or attempting to solve one problem (i.e., paradox)? If your explanation is based on your speculation, explicitly express your uncertainty. Explain it without being verbose. Explain it in maximum two sentences. Show only the result in the following format: `{output template}`

Table 6: The STRUCTURED approach's prompts related to exploring stakeholders. We intentionally avoided using the term "problem" when referring to the problem description. This helped guide the LLMs to identify stakeholders within a broader context rather than remain confined to a problematic viewpoint. Step 4 focuses on retrieving factual information about stakeholders explicitly mentioned in the problem description, while steps 5 and 6 exploit the LLMs' generative capacities to suggest potential stakeholders and what each stakeholder values in the given situation. Particularly for step 6, we updated GPT-4o's system prompt to produce responses from a specific stakeholder's angle, taking advantage of LLMs' strength in role-playing [34, 68]. We also limited each stakeholder's perspective to five points, since GPT-4o tends to generate excessively verbose and repetitive lists. To further facilitate generation of unique perspectives specific to the stakeholder in question, we provided a list of other stakeholders, enabling ready differentiation.

Step 4: Current stakeholders	Here is a description of a situation: `{problem description}` List stakeholders, people who are involved in the situation. List the ones stated in the description. Show only the result in the following format: `{output template}`
Step 5:Potential stakeholders	Here is a description of a situation: `{problem description}` List potential stakeholders who are not stated in the description. Show only the result in the following format: `{output template}`
Step 6: Stakeholders' perspectives	You are a "`{stakeholder}`" in the following situation: `{problem description}` Answer my question as a "`{stakeholder}`". In contrast to other stakeholders such as `{list of other stakeholders}`, what are the things that only you care about? Tell me five fundamentally different ones. Show only the result in the following format: `{output template}`

Table 7: How we implemented the STRUCTURED approach particularly for "Synthesizing Problem Themes." We experimented with two prompting strategies: a) clustering stakeholders' perspectives, then generating a theme for each cluster and b) identifying N unique needs from stakeholders' perspectives. In testing with GPT-4o, we found that the clustering often failed to capture stakeholders' views accurately. It tended to exclude several perspectives (seemingly at random), group unrelated topics together, or generate overly simplistic themes (e.g., producing vague themes such as "Inclusiveness" instead of more specific ones such as "Facilitating an inclusive and welcoming environment"). In contrast, prompting GPT-4o with a specific number of unique needs to identify resulted in more distinct perspectives. We set the number of unique needs to 10 and allowed users to (re-)generate alternative themes if they deemed this necessary.

Step 7: Themes	Here are what stakeholders care about in a problem situation: `{stakeholders' perspectives}` Capture the ten most fundamental and unique needs. Write each need briefly, without being verbose. Show only the result in the following format: `{output template}`

Table 8: Our system's handling of steps 8 and 9, which reverses the order from Dorst's procedures of having designers generate problem frames and then assess them. In our system, LLMs evaluate how viewing the problem in terms of each theme could assist in reframing (step 8), then use the assessments to generate an alternative problem frame (step 9). While designers typically make assumptions about alternative perspectives before generating frames, LLMs do not naturally follow such a process. Changing the order exploits LLMs' reasoning capabilities to guide more effective reframing. For the assessment, we applied Dorst's evaluation criteria for problem frames [19] to guarantee high-quality evaluation in reframing.

Step 8: Frame assessment	Here are descriptions of a problem: `{apparent problems}` Here is why solving the problem is difficult: `{paradoxical problems}` To overcome the difficulty of solving the problem, we now look at the problem from a completely different angle, "a problem of `{theme}`". Think step by step to explain the benefits of our perspective change in detail: - Usefulness: How useful is our perspective change for solving the difficult problem? - Additional benefits: What are the other benefits of solving the problem from our perspective? - Feasibility: What changes need to be made to solve the problem from our perspective? Show only the result in the following format: `{output template}`
Step 9: Problem frames	Here is a description of a problem: `{problem description}` Here is why solving the problem is difficult: `{paradoxical problems}` In response, we now look at the problem from a completely different angle, "a problem of `{theme}`". Here is how we expect the perspective change to solve the problem: `{frame assessment}` According to our expectation, reframe the problem in a single sentence. Write it concisely without being verbose. Show only the result in the following format: `{problem frame template}`

B QUIZ FOR TESTING PARTICIPANTS' UNDERSTANDING OF PROBLEM REFRAMING

We designed a quiz to test participants' competence in problem reframing. This is reproduced as Table 9. While designers are often categorized as "experts" on the basis of their number of years of experience in the industry, experience on its own is not an accurate proxy for how well designers understand the concept of problem reframing [16, 23]. For instance, some designers with several years of practice under their belt might not have engaged in problem reframing or may have an incorrect understanding of the process, due to a lack of formal training or to exposure to poor definitions. We developed five quiz items that helped us assess participants' knowledge spanning the key concepts, processes, and outcomes of problem reframing. The questions and response options were based on the core literature about problem reframing [15, 19, 65]. So that participants could not easily get categorized as experts by choosing responses at random. we utilized items that could have multiple correct answers.

Table 9: The quiz items used in our study and the correct answers

Q1. Which are *incorrect* descriptions of problem reframing? Select all.	
It requires iterative exploration of the problem and solutions space.	✗
It is exploring alternative ways of approaching the problem.	✗
It often requires collecting more information about stakeholders.	✗
It is developing solutions to initial problems only.	✓
Q2. Which are the tasks in problem reframing? Select all.	
Understanding why the problem is difficult to solve.	✓
Assessing the feasibility of potential solutions to the reframed problems.	✓
Gathering fund for implementing solutions.	✗
Filtering out stakeholders who are causing the problem.	✗
Q3. Which cases describe reframing design problems? Select all.	
I reframed a math problem to find a better solution.	✗
I redefined a problem based on what stakeholders find important to themselves.	✓
I summarized the original problem description into a problem framing.	✗
I defined a new problem by uncovering the root cause of the problem.	✗
Q4. Which are the final outcomes of reframing a problem? Select all.	
Improved understanding of the problem.	✓
A description of the final problem framing.	✓
A consensus on the most effective solution.	✗
A finalized prototype of the solution.	✗
Q5. Which are *not* important for problem reframing? Select all.	
Overcoming fixation on the initial problem.	✗
Blaming stakeholders who are causing the problem.	✓
Exploring potential solutions to the reframed problems.	✗
Speculating additional benefits of solutions other than solving the initial problem.	✗

C THREE DESIGN PROBLEMS FOR REFRAMING

We articulated three design problems for reframing, described below.

(1) **Aging**: As life expectancy increases, more elderly individuals are working longer, particularly in the delivery industry. However, their age-related declines in reaction time and vision have led to an increased rate of traffic accidents involving senior drivers. This led to public concern and pressure on the government to enhance safety on the roads. In response, the government has created a law to limit the maximum age for drivers in the delivery industry. On the contrary, this solution has sparked a backlash from senior drivers, who feel that the law unfairly limits their ability to work and maintain their livelihood.

(2) **Misbehavior**: A public healthcare service is struggling with the littering of cigarette butts. Despite having multiple smoking booths with ashtrays, smokers tend to throw their cigarette butts on the street. In response, the healthcare service promoted anti-smoking campaigns and increased the fines for littering. However, the littering problems have not been solved, leading to frustration among pedestrians and residents. Furthermore, the idea of prohibiting smoking has caused strong opposition from smokers, and non-smokers worry that prohibition will push smokers to find hidden places for smoking, exacerbating the littering problem.

(3) **Preference conflict**: In an office, employees have established a tradition of having lunch together, which has strengthened their team bond and promoted social interactions. However, since the number of employees has increased, choosing a lunch spot that can satisfy everyone has become quite challenging. Now, employees face multiple conflicting preferences, such as food quality over price or distance to restaurants and vice versa. These considerations often lead to having an excessively long discussion or everyone having lunch separately, undermining their valued lunchtime culture.

D PILOT STUDY

We conducted a pilot study to confirm the feasibility of reframing the design problems we had prepared (reproduced in Appendix C). Using the same procedure and platforms as for the main study, we recruited 36 people (17 self-identifying as women and 19 as men) from design-related industries and randomly assigned three participants to each design problem and approach (i.e., each problem was reframed by 12 participants).

With the pilot study, we internally reviewed the frames' quality. Overall, the participants could generate alternative frames. For each problem, we found 2–3 of the 12 problem frames novel or useful: they discussed perspectives that were new / not included in the original problem descriptions. For example, P19 reframed the problem of cigarette-butt litter as one of fostering an aesthetically pleasing environment that can autonomously deter misbehaviors rather than punish them. In the case of road safety declining because of society aging, P6 suggested focusing on designing alternative career-development programs for seniors instead of merely banning them from driving. Likewise, P17 introduced the perspective of transforming lunch-spot conflicts into social activities that can enhance group culture. There were also a few low-quality framings, aligned with the original problem descriptions, such as punishing misbehavior / using coercive techniques to prevent senior citizens from driving or resolving conflicts. The pilot study showed that our design problems can be reframed into both low- and high-quality problem frames. This supports concluding that the problems are challenging yet able to be reframed.

E PARTICIPANTS' DEMOGRAPHICS

Table 10: Participant demographics connected with each number of correct answers in the quiz

Number of correct answers	**Age** mean (and SD)	**Self-identified gender**: woman, man, other	**Design experience** mean years (SD)	*N*
0	31.97 (9.06)	19, 15, 0	5.62 (4.57)	34
1	34.88 (11.19)	38, 42, 1	7.78 (7.21)	81
2	35.66 (11.00)	50, 33, 3	8.88 (7.83)	86
3	33.20 (10.00)	24, 21, 1	7.33 (5.83)	46
4	31.14 (10.51)	12, 14, 2	7.86 (7.68)	28
5	30.40 (3.65)	2, 3, 0	3.00 (2.35)	5

Table 11: Demographics of the experts who evaluated the problem frames

Number of correct answers	**Age** mean (and SD)	**Self-identified gender**: woman, man, other	**Design experience** mean years (SD)	*N*
3	32.21 (9.67)	1, 2, 0	6.33 (3.73)	3
4	32.60 (11.12)	4, 6, 1	7.86 (7.68)	11
5	32.00 (N/A)	0, 1, 0	1.00 (N/A)	1