PlanTogether: Facilitating AI Application Planning Using Information Graphs and Large Language Models

Dae Hyun Kim[*]
dhkim16@yonsei.ac.kr
Yonsei University/KAIST
Seoul/Daejeon, Korea

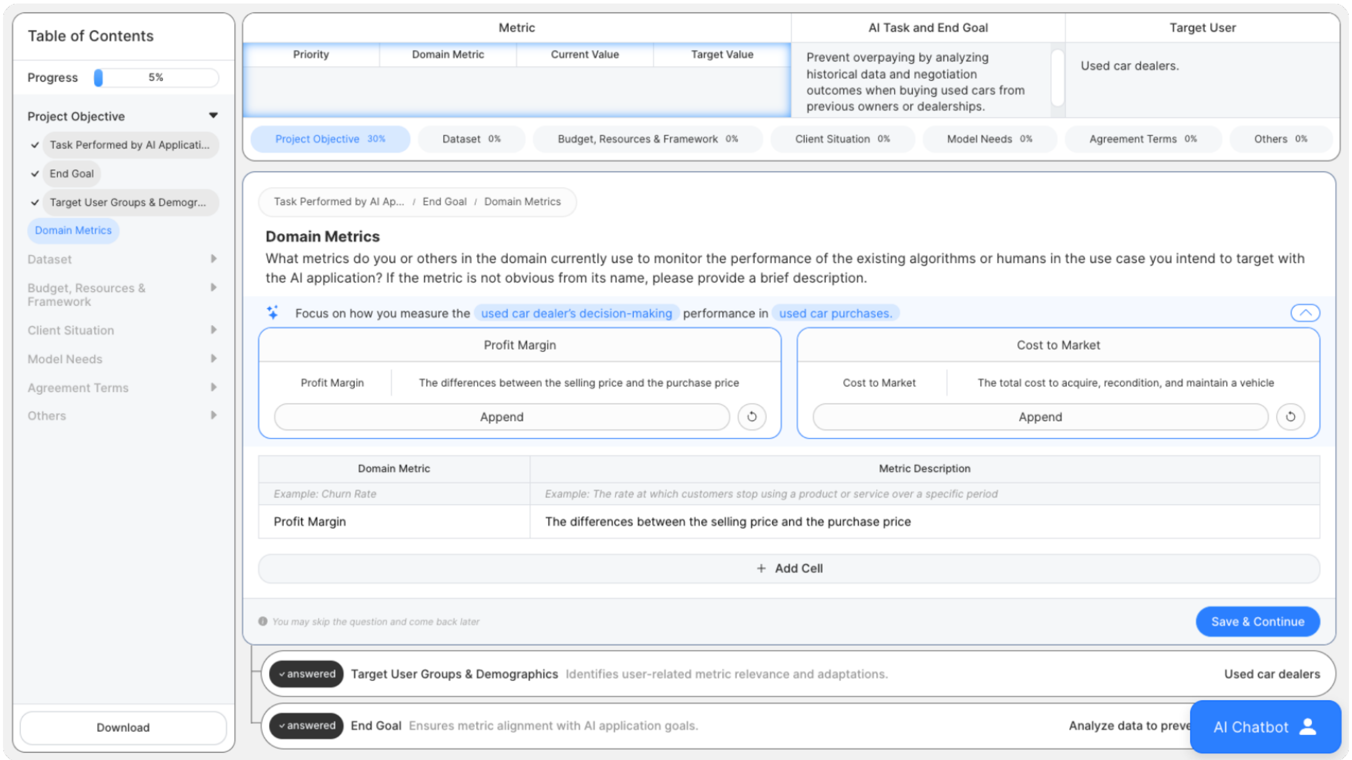
Daeheon Jeong[*]
daeheon.jeong@kaist.ac.kr
KAIST
Daejeon, Korea

Shakhnozakhon Yadgarova
Hyungyu Shin
{yadgarova,hyungyu.sh}@kaist.ac.kr
KAIST
Daejeon, Korea

Jinho Son
sjhfam@algorithmlabs.co.kr
Algorithm Labs
Seoul, Korea

Hari Subramonyam
harihars@stanford.edu
Stanford University
Stanford, CA, USA

Juho Kim
juhokim@kaist.ac.kr
KAIST
Daejeon, Korea



Metric				AI Task and End Goal	Target User
Priority	Domain Metric	Current Value	Target Value	Prevent overpaying by analyzing historical data and negotiation outcomes when buying used cars from previous owners or dealerships.	Used car dealers.

Domain Metrics

What metrics do you or others in the domain currently use to monitor the performance of the existing algorithms or humans in the use case you intend to target with the AI application? If the metric is not obvious from its name, please provide a brief description.

Focus on how you measure the used car dealer's decision-making performance in used car purchases.

Profit Margin

Profit Margin | The differences between the selling price and the purchase price

Cost to Market

Cost to Market | The total cost to acquire, recondition, and maintain a vehicle

Domain Metric	Metric Description
Example: Churn Rate	*Example: The rate at which customers stop using a product or service over a specific period*
Profit Margin	The differences between the selling price and the purchase price

You may skip the question and come back later

Target User Groups & Demographics Identifies user-related metric relevance and adaptations. Used car dealers

End Goal Ensures metric alignment with AI application goals. Analyze data to preve

Figure 1: PlanTogether is a system that includes support features (tips, suggestions, and overview) to help clients with pre-collaboration plans for an AI application. This figure shows a view of PlanTogether while the client answers a question on 'Domain Metric' regarding an idea about an AI application for used car dealers (Section 3). In addition to a question-answering interface, the system provides tips and suggestions below the question. It also displays a plan overview at the top of the screen. The system helps clients navigate information dependencies and write actionable plans reflecting their domain expertise.

Abstract

In client-AI expert collaborations, the planning stage of AI application development begins from the client; a client outlines their needs and expectations while assessing available resources (*pre-collaboration planning*). Despite the importance of pre-collaboration plans for discussions with AI experts for iteration and development, the client often fails to reflect their needs and expectations into a

[*] Both authors contributed equally to this research.

CHI '25, Yokohama, Japan
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3714044

concrete actionable plan. To facilitate pre-collaboration planning, we introduce PlanTogether, a system that generates tailored client support using large language models and a *Planning Information Graph*, whose nodes and edges represent information in the plan and the information dependencies. Using the graph, the system links and presents information that guides client's reasoning; it provides tips and suggestions based on relevant information and displays an overview to help understand the progression through the plan. A user study validates the effectiveness of PlanTogether in helping clients navigate information dependencies and write actionable plans reflecting their domain expertise.

CCS Concepts

• **Software and its engineering** → **Designing software**; • **Computing methodologies** → **Artificial intelligence**; • **Human-centered computing** → *Interactive systems and tools.*

Keywords

AI application planning, planning support system, information graph, personalized guidance

ACM Reference Format:
Dae Hyun Kim, Daeheon Jeong, Shakhnozakhon Yadgarova, Hyungyu Shin, Jinho Son, Hari Subramonyam, and Juho Kim. 2025. PlanTogether: Facilitating AI Application Planning Using Information Graphs and Large Language Models. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26-May 1, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/3706598.3714044

1 Introduction

Collaborations between clients and external AI experts (e.g., AI solutions companies, freelance AI engineers) have become widespread when building AI applications; such collaboration enables domain experts, who often possess limited AI expertise, to realize their AI application ideas [112]. The AI applications enabled by these collaborations include product quality control [81], cancer detection and treatment [52], hotel operation and sales management [45], and customer support chatbots [45, 124]. Within the client-AI expert collaboration, the *planning stage*, which precedes the AI expert's application development by executing the plan, is a key stage of information flow, which usually follows three phases: pre-collaboration phase, main discussion phase, and execution preparation phase [65, 95]. Planning starts from the client as they define their needs and expectations from the AI application and assesses the available resources prior to commencing collaboration with the AI expert (*pre-collaboration phase*). Next, the client communicates with AI expert, often via discussions and documentation exchanges (e.g., project specifications), where they exchange the required insights and calibrate needs and expectations based on technical feasibility (*main discussion phase*). Finally, the planning moves to the AI expert, who would come up with the technical approaches to meet the client's calibrated needs and expectations and the resources available (*execution preparation phase*).

Due to the dependency of the later planning phases on the outcomes of the pre-collaboration phase, the quality of the *pre-collaboration plan* produced by the client can heavily influence the experience during the main discussion phase. However, the client often faces challenges in coming up with a concrete and actionable pre-collaboration plan due to knowledge barriers (i.e., lack of knowledge about terminology and AI capability, understanding of how to tie their specific plan with the knowledge) and a lack of understanding of the AI expert's information needs [11, 26, 65, 116]; without a concrete and actionable pre-collaboration plan, the client and AI expert must undergo multiple rounds of iteration for the AI experts to outline development plans, which can be both time-consuming and frustrating [65]. Existing workbooks outline the key considerations in AI application development (e.g., AINeedsPlanner [65], People+AI Workbook [100]). However, static workbooks cannot flexibly provide support (e.g., response examples) that covers the long tail of possible AI applications, resulting in mental burden and inaccurate interpretations when the client tries to transfer the information over into their own plan [135]; tailored support for each plan can lead to a better-situated understanding of how to answer each consideration in the context of the plan [15, 29, 56, 70, 127].

To facilitate pre-collaboration planning in AI application development, we introduce PlanTogether (Figure 1), a system that provides guidance tailored to each client as they outline an initial pre-collaboration plan. Beneath an intuitive user interface that resembles a typical web survey form with an overview of planning progression, the system models the contents of the pre-collaboration plan as the *Planning Information Graph*, a naturally arising data structure arising from the abundance of dependencies between the various pieces of information required for AI application planning [64]. The graph, with nodes representing pieces of planning information and edges representing information dependencies, is grounded on the contents of the taxonomy of AI expert's information needs and the AINeedsPlanner workbook, a state-of-the-art workbook for pre-collaboration planning [65]. This graph structure allows the system to traverse the graph and form an ordering of the questions that prioritizes asking relevant questions first to guide the client through the planning process while simultaneously compiling the information necessary for generating helpful tips and suggestions. The Guidance Generator module utilizes the information redundancies in the graph to combine user answers with related questions, and presents personalized tips and suggestions for answering specific questions.

Through a user study, we find that the graph-based features and the overview included in PlanTogether make crucial contributions in helping the clients navigate and take advantage of the dependencies present in pre-collaboration planning, leading to more actionable plans that better captures the client's domain expertise.

The contributions of this work include:

- PlanTogether, a graph-based system that provides situated personalized guidance for clients as they blueprint a pre-collaboration plan during AI application development;
- The Planning Information Graph, a graph representation for the information of the pre-collaboration plan;
- A preliminary study confirming the accuracy of the system and domain experts' ability to discern system errors; and
- A user study supporting the helpfulness of personalized tips and suggestions as well as overviews in pre-collaboration planning in arriving at more actionable plans that better reflect the client's domain expertise.

2 Related Work

Our work is related to three areas of prior work: (1) planning in AI application development, (2) graph representation for reasoning support, and (3) LLM-powered intent elicitation.

2.1 Planning in AI Application Development

Because general software development is a complex process that typically involves multiple components and multiple stakeholders [12, 44, 61, 131], careful planning and requirement collection are central to successful outcomes [5, 12, 53]. The importance of planning has led to various requirement collection tools (e.g., Asana [51], Jira [7], DOORS [50]). As a special case of software development, AI application development requires careful planning to navigate the intricacies of component dependencies and collaborations between multiple stakeholders [64, 65, 80, 93, 116–118, 122].

However, recent work has identified unique characteristics of AI application development that calls for specialized support tools for AI application planning. A key characteristic is the data-centric nature of AI applications [2, 79, 98]. The need for large amounts of high-quality data necessitates careful planning around collection and labeling methods as well as measures against bias and anomalies [65, 110, 122, 128]. Moreover, AI technology is complex and introduces performance uncertainties prior to model development [2, 64, 80]. Hence, AI application planning requires calibrating expectations about if and how well AI can perform given tasks with the available resources [22, 79, 96] and careful risk assessment [23, 85, 86]. Other unique characteristics include the need of dedicated infrastructure for training and prediction (e.g., GPUs, data storage) [80, 103, 113] and adaptations to shifting data distributions over time [48, 113]. Our work builds on these unique characteristics to introduce a tool specialized in AI application planning for clients.

The unique characteristics of AI application development have motivated researchers and practitioners to develop various tools and systems to aid AI application planning. A thread of work has focused more on specific parts of AI application planning (e.g., data [9, 39, 48, 93], user experience and interface [28, 85, 117]). Existing tools also assist navigation of AI risks [23, 86], lower-barrier communication between stakeholders [28, 59], model design iterations [3], and clarification of stakeholder interests [20, 27, 137]. Another thread of work attempts to taxonomize the various considerations required in AI application planning and organize the considerations into guidelines [3, 43, 78, 83, 100, 116, 129] or workbooks [65, 100]. AINeedsPlanner [65] specifically targets client-AI expert collaborations with various forms of questions designed to guide the stakeholders. The workbook includes fill-in-table, fill-in-the-blank, and free-form questions structured into 8 chapters and 19 subchapters; the questions include static examples constructed by the authors. As a part of the ongoing efforts to facilitate AI application planning, our work expands the horizons of existing support tools as the first to introduce dynamic tailored assistance to elicit concrete and actionable AI application ideas from clients during the pre-collaboration phase. In particular, while we ground the contents of PlanTogether on AINeedsPlanner, PlanTogether significantly extends the workbook by introducing the Planning Information Graph that captures the information dependencies between the questions to guide client's reasoning and to dynamically generate personalized tips and suggestions.

2.2 Information Graph for Reasoning Support

Information graphs are structured representations of information that uses *nodes* to represent each unit of information and *edges* to represent their relationships [55]. Their focus on relationships between information has allowed the adoption of information graphs in various types of information. Information graphs are capable of capturing structures present within document contents [101] (e.g., co-occurrence of entities or concepts [57, 142], relationship between concepts [139], semantic relations between contents [14, 114]); they can capture relationships between documents (e.g., citation networks [82, 99, 141]). Information graphs are also often adapted to the needs of specific domains. In the education domain, knowledge graphs represent concepts and the dependencies between them [75, 119, 123]; in the productivity domain, workflow graphs represent the flow of information and prerequisites between the comprising tasks [24, 36, 40, 126]. Previous research has found the benefits of using information graphs in managing, navigating, and reasoning about complex interconnected data, for both humans [16–18, 25, 33, 88, 106] and intelligent systems [32, 76, 121, 138]. Since information involved in AI application planning is complex and interconnected [64], we leverage the Planning Information Graph to capture the considerations and the information dependencies between them.

Building on the benefits of information graphs, researchers have developed visualizations and techniques to harness the information graph's ability to aid human reasoning. Many techniques assist users by using visualization and visual interaction methods (e.g., digital whiteboard interactions [91, 120], natural language interactions [111], structured exploration that surfaces substructures [73, 120, 133]), although some tools based on citation networks (e.g., Google Scholar [104], Semantic Scholar [105]) keep the graph implicit while relying on graph traversals. In the learning domain, other than visualizing the relationships between concepts [119, 123] and supporting navigation [75], researchers have shown the possibility of tailoring learning materials using by modeling the knowledge state of the learners [1, 19, 31, 71, 87, 134]. Furthermore, in the productivity domain, researchers demonstrated the possibility of supporting workflow optimization using the workflow graph [15, 54, 66]. While our system also focuses on providing graph-based reasoning support for users, it focuses on the unexplored problem of supporting clients in AI application planning.

2.3 LLM-Powered Intent Elicitation

With the advent of large language models (LLMs), researchers have quickly adapted the technology into intent elicitation tools and techniques. The LLM-powered tools help users discover and express their intents through various user-agent interactions (e.g., simulated dialogues [30, 77], question-answering [37, 74, 108], user feedback [94], multi-agent conversations [92]). The LLM agents collaborate with the users by seeding initial ideas [35, 125, 136], expanding ideas [107], and iteratively refining the ideas [41]. Prior work shows that LLM enables more efficient and accurate intent elicitation due to its nuanced understanding of language and reasoning capabilities [6, 8, 69]. As a system for capturing the users' AI application development intents into a plan, PlanTogether benefits from using an LLM.

Furthermore, recent work has shown the importance of proper contextualization in LLM-based intent elicitation. Hong et al. [47] points to the importance of using contextual clues (e.g., location, action) beyond information present within the dialogue. Conversely, situating the interactions with LLMs in well-designed contexts (e.g., scenes, scenarios) and reflecting information learned from past interactions can boost the efficacy of intent elicitation [4, 37, 74, 136]. In particular, Louie et al. [77] and Gero et al. [35] find that contexts familiar to the user can lead to natural elicitation of intents that capture the user's domain knowledge [35, 77]. Based on the promises of contextualization in improving elicitation efficacy and capturing domain knowledge, we carefully designed our system to utilize the contents in the plan to provide LLM-powered support that reflects the current status of the plan and the user's situation.

3 Usage Scenario

In this section, we describe a usage scenario of PlanTogether with an example of Sarah, a used car dealership manager. She realizes that her dealership frequently overpays when purchasing used cars from their previous owners. Based on success stories of applying AI in various domains, she decides that AI could also help with her situation. Prior to the consulting an AI expert, she organizes her situation and needs using PlanTogether, which offers her personalized support.

Specifically, we first explore how the system interface and features guide her user flow with the first few questions in the 'Project Objectives' section (Section 3.1). We then further highlight the value of the system's situated guidance (tips and suggestions) through a comparison with Rick, who uses a baseline web survey without the situated guidance (e.g., AINeedsPlanner [65]) to perform pre-collaboration planning (Section 3.2).

The scenario is based on real usage patterns from the preliminary evaluation (Section 5) and the user study (Section 6).

3.1 Sarah's User Flow

We first follow Sarah's user flow as she uses the system interface and features to navigate through the first few questions in the 'Project Objectives' section.

Step 1: Starting with 'Task Performed by AI Application'. (Figure 2a) Sarah starts by working on the 'Project Objectives' section, which starts with the question on 'Task Performed by AI'. She sees the question *"Please explain the task the AI application needs to perform."* and starts thinking about how she can answer it. While thinking about this question, she sees that the Overview Panel at the top of the screen is empty as she has not yet provided any information to the system. In this panel, she also sees that the 'AI Task and End Goal' part of the panel is highlighted, indicating that her answer to the current question would influence this part. However, she realizes that she is unable to answer this question without any tips and suggestions. She decides to move on to the unanswered relevant question about the 'End Goal', something that she thinks she can answer.

Step 2: Answering 'End Goal' (Figure 2b) Sarah now sees the new question on 'End Goal': *"Describe the end goal of your AI application."* For this question, she decides that she can answer this question based on her motivation and types in: *"Make better decisions on used car purchases."* As she types in the answer, she notices that her answer would again influence the highlighted 'AI Task and End Goal' part of the Overview Panel, which helps her see that there is a potential correlation between the current question and the previous question.

Step 3: Confirming Answer Quality. (Figure 2b) After inputting the answer, Sarah wants to confirm whether her answer meets the intentions of the question. She clicks on the AI chatbot at the bottom right of the screen and asks, *"Would my answer for End Goal be helpful for AI experts in charge of AI application development?"* The chatbot responds: *"Instead of simply stating 'Make a better decision on the used car purchases,' clarify what the better decision indicates to suggest how the AI application can improve the decision making."* Based on the response, Sarah modifies her answer to: *"Avoid overpaying when purchasing used cars from their previous owners."*

Satisfied with her answer, she clicks on the "Save & Continue" button to move on to the next question. She realizes that the 'AI Task and End Goal' part of the Overview Panel is now filled in as *"Prevent overpaying when buying used cars from previous owners."* based on her answer.

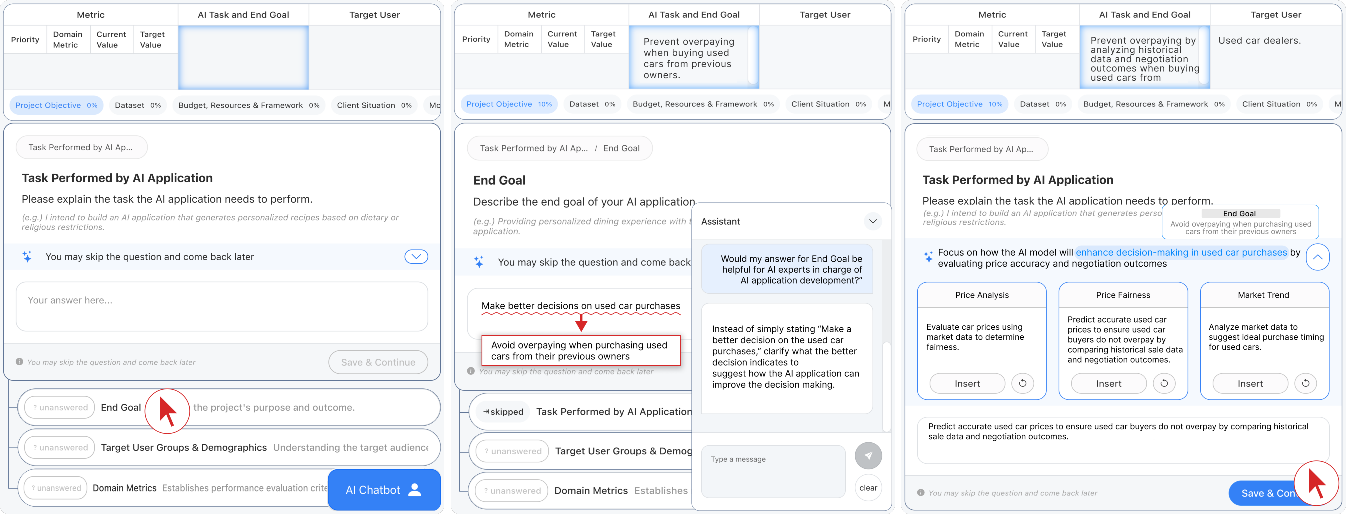
Step 4: Re-Attempt 'Task Performed by AI Application' (Figure 2c) After answering a series of questions, Sarah eventually arrives back at the question she skipped earlier about 'Task performed by AI Application'. This time, because she has provided more information to PlanTogether, it is able to generate more useful tips and suggestions for her. Specifically, she now sees the tip *"Focus on how the AI model will enhance decision-making in used car purchases by evaluating price accuracy and negotiation outcomes."* with a blue highlight on the phrase: *"enhance decision-making in used car purchases."* Hovering over this phrase, she sees that this tip is drawing information from the previously answered 'End Goal' and that she needs to carefully think about the end goal as she answers the current question. She clicks on the downward arrow button (⌄) next to the tip to obtain personalized suggestions tailored to the current state of the client-side plan that she can refer to while answering this question.

3.2 How System's Tips and Suggestions Improves Client Experience

Next, we explore how the system's personalized tips and suggestions improve client experience by discussing how Sarah would approach a question including an unfamiliar concept. We highlight the improvement of client experience by drawing a comparison with Rick in the same situation, who instead uses a baseline web survey with the same contents, but without the system features.

3.2.1 *Rick's Experience with a Baseline Web Survey.*

As Rick fills the baseline web survey, he makes it to the question about 'Domain Metrics': *"What metrics do you currently use to monitor the performance of the algorithms or humans?"* He is unfamiliar with



Task Performed by AI Application

Please explain the task the AI application needs to perform.

(e.g.) I intend to build an AI application that generates personalized recipes based on dietary or religious restrictions.

You may skip the question and come back later

Your answer here...

You may skip the question and come back later

End Goal the project's purpose and outcome.

Target User Groups & Demographics Understanding the target audience

Domain Metrics Establishes performance evaluation crite

(a) Step 1: Skipping the question

Prevent overpaying when buying used cars from previous owners.

End Goal

Describe the end goal of your AI application

(e.g.) Providing personalized dining experience with t application.

Make better decisions on used car purchases

Avoid overpaying when purchasing used cars from their previous owners

Would my answer for End Goal be helpful for AI experts in charge of AI application development?"

Instead of simply stating "Make a better decision on the used car purchases," clarify what the better decision indicates to suggest how the AI application can improve the decision making.

(b) Steps 2, 3: Answering using AI chatbot

Prevent overpaying by analyzing historical data and negotiation outcomes when buying used cars from

Used car dealers.

End Goal: Avoid overpaying when purchasing used cars from their previous owners

Focus on how the AI model will enhance decision-making in used car purchases by evaluating price accuracy and negotiation outcomes

Price Analysis: Evaluate car prices using market data to determine fairness.

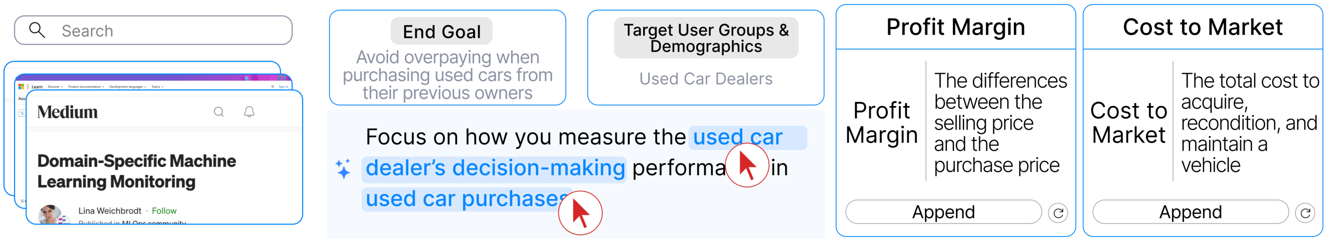
Price Fairness: Predict accurate used car prices to ensure used car buyers do not overpay by comparing historical sale data and negotiation outcomes.

Market Trend: Analyze market data to suggest ideal purchase timing for used cars.

Predict accurate used car prices to ensure used car buyers do not overpay by comparing historical sale data and negotiation outcomes.

(c) Step 4: Provided tip & suggestion

Figure 2: Views of the PlanTogether interface from the usage scenario in Section 3.1. We omit irrelevant parts of the interface (e.g., Table of Contents) for focused illustration.



Domain-Specific Machine Learning Monitoring

Avoid overpaying when purchasing used cars from their previous owners

Used Car Dealers

Focus on how you measure the used car dealer's decision-making performa in used car purchases

Profit Margin: The differences between the selling price and the purchase price

Cost to Market: The total cost to acquire, recondition, and maintain a vehicle

(a) Rick & Sarah: Web search

(b) Sarah: PlanTogether tip

(c) Sarah: PlanTogether suggestions

Figure 3: The tools Rick and Sarah use to answer the question on 'Domain Metrics' in the usage scenario (Section 3.2).

the notion of 'domain metrics' and lacks ideas about how he can approach this question. He opens up a search engine and searches for *"domain metric"*. After digging through relevant webpages, he finds the definition *"custom metrics that are specific to your product"* and the importance of capturing user expectations and experience [130], which gives him a rough grasp of the terminology. However, Rick has difficulty linking the information and his situation into a coherent answer; he forms the need for some examples would get him started – he searches for *"domain metric examples"* and finds examples in domains such as marketing and education. Because the problem of used car purchases is specific and relatively uncommon, the examples require extra effort for Rick to link to his problem and formulate his answers; he ends up browsing multiple webpages with various modifications to the query before arriving at a level of understanding to produce an answer.

3.2.2 *Sarah's Experience with PlanTogether.*

Sarah, working with PlanTogether, also arrives at the question about 'Domain Metrics'. Unlike Rick, Sarah sees a tip for answering the question: *"Focus on how you measure the used car dealer's decision-making performance in used car purchases."* Hovering over the highlighted phrases *"used car dealer's"* and *"decision-making in used car purchases"*, Sarah sees references to her previous answers in the questions about 'Target User Groups & Demographics' and 'End Goal', respectively. She re-reads the question in the context of her previous answers.

After some thought, she decides to get concrete examples by clicking on the expand button (⌄), which displays the suggestions: *"Profit Margin"* (the difference between the selling price and the purchase price) and *"Cost to Market"* (the total cost to acquire, recondition, and maintain a vehicle). Through the examples, Sarah forms a rough idea about how she should approach this problem, but she decides to search for *"domain metric"* to ensure that her mental model is correct; the same webpage Rick saw about domain metric [130] confirms her understanding. She combines her understanding with the question context and her domain knowledge (maximizing profit margin is the goal of used car dealers, price gap aligns with the problem of overpayment) and decides to accept the suggestion: *"Profit Margin."*

4 The PlanTogether System

PlanTogether is a system for supporting pre-collaboration planning during AI application development that utilizes information graphs and LLMs. Based on prior literature on interactive user interfaces, human-AI collaboration, and AI application planning, we designed the system interface and features along the following design principles (Figure 4):

DP1 [*Provide Situated Support Based on Available Information*] The information interdependency in pre-collaboration planning [64] means that each piece of planning information can contribute to generating personalized support; the system should use relevant information given by the users when providing in-situ support, adapting to additional information that becomes available [10, 49, 84] and being transparent about what led to the provided support [10, 49].

DP2 [*Engage Users Throughout the Planning Process*] Pre-collaboration planning is a process of consolidating and expressing intent as well as decision-making for the clients [65, 96]; the system should not automate away users' thought process, but keep users involved in decision-making by invoking user thoughts and actions [10, 13, 49, 60, 102].

DP3 [*Keep Users Aware of the Big Picture*] Clients can lose sense of the big picture when dealing with the immense and highly-interconnected information in pre-collaboration planning [64, 65]. To ensure a sense of progression and an understanding of how each piece fits into the overall plan, the system should ensure that the big picture is visible to the users [97, 132, 140] and structured and presented in a way that is easy to parse [58, 115].

DP4 [*Assume Minimal Technical Background*] Because the target users of the system are clients who may not possess expertise in AI and other computer-science-related domains [64, 65, 116], the system should use design conventions familiar to the those with minimal technical background [10, 115].

Based on these design principles, the interface (Figure 6 encapsulates the complex data structures and algorithms underneath an intuitive interface modeled after web survey forms (DP4) with proper overviews for tracking progression (DP3). Underneath the interface, the system comprises the *Planning Information Graph* and the *Guidance Generator* (Figure 5). The system captures the interdependencies of the pre-collaboration planning information by using the *Planning Information Graph*, whose nodes represent each unit of planning information and edges represent the dependencies between them. The *Guidance Generator* module generates in-situ tips and suggestions tailored to the current status of the plan by using the edges in the graph (DP1) and induces user agency by requiring comparisons of multiple suggestions (DP2). The module also generates an overview that surfaces the progression of the plan (DP3).

We used LLMs when generating various guidance and summaries for their reasoning capabilities and abilities to utilize knowledge about the real world. We used Open AI's `gpt-4o-2024-08-06` and `gpt-4o-mini-2024-07-18`, the state-of-the-art models at the time of development. We include the LLM prompts in the supplemental material.

4.1 System Interface

Based on DP4, PlanTogether interface (Figure 6) is modeled after typical web survey forms to leverage familiarity and minimize the entry barrier for users; it includes the *Main Panel* (Figure 6C) that presents questions and receives answers from the users while providing tips and suggestions, as well as the *Table of Contents* on the left (Figure 6A) showing the plan structure and completion (denoted by ✓ on the left) that the user can use to navigate between questions. The *Overview Panel* on top (Figure 6B) further complements the functionality of Table of Contents in providing the big picture to the users according to DP3 by providing a structured summary of the planning information provided by the user and highlighting where the answer to the current question would fit into the overview.

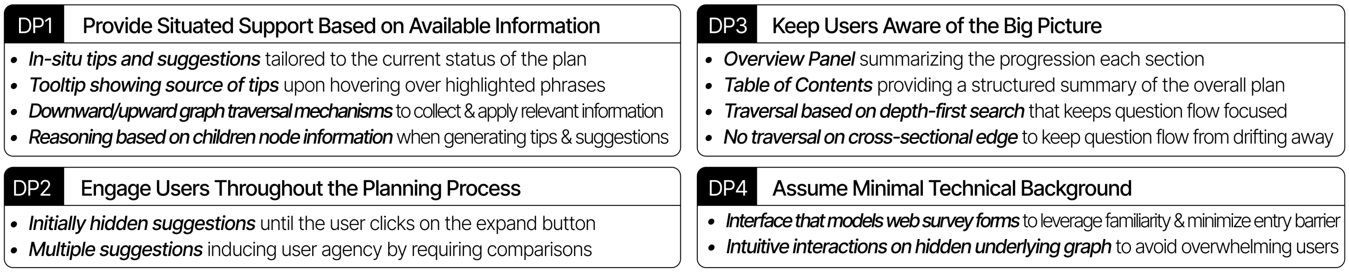
As the primary component for receiving user input, the main panel includes support features based on the underlying Planning Information Graph. It provides tips (Figure 6Cii) for answering the question and suggestions of potential answers (Figure 6Ciii) based on the answers provided by the user (DP1). The interface not only conveys where the system obtained information for generating the tips when a user hovers over highlighted phrases of the tip (Figure 6Ci) (DP1), but also generates up to three suggestions instead of one to require the users to actively read and compare the suggestions before selecting one or providing their own answer (DP2). Moreover, to promote active independent thinking, the suggestions remain hidden until the user clicks on the expand button (⌄) (DP2). At the bottom (Figure 6Civ), the Main Panel also provides *graph-based navigation* to relevant questions (i.e., neighboring nodes on the Planning Information Graph). The decision to not explicitly surface the graph is a deliberate design choice based on pilot studies involving various visualizations of Planning Information Graph revealing that clients are often unfamiliar with and overwhelmed by graph visualizations (DP4). Further detailed descriptions of the interface panels are included in Appendix A.

To reduce the cognitive effort from switching windows to use an LLM chatbot and designing prompts reflecting the current plan status, we include an LLM-based *AI chatbot* (Figure 6D) in the system that can access user responses.

4.2 Planning Information Graph

The *Planning Information Graph* (Figure 7) represents the information covered in the pre-collaboration plan; the *nodes* represent the various pieces of information involved in pre-collaboration planning and the *edges* represent the information dependencies between the nodes. Nodes are grouped into *sections* based on the information contents to help users mentally group questions and navigate through them. This graph structure, which naturally arises from the inter-dependencies of planning information [64], allows the system to directly identify the nodes whose information can help reason about a specific nodes; the system simply needs to traverse to children nodes.

A *node* of the Planning Information Graph (Figure 7A) is a question-answer pair with a title (`Title`, `Question`, `Answer`). It also includes additional properties that the system needs to traverse and generate new nodes in the graph (`Type`, `Embedding Vector`) as well as display answering interfaces (`Answer Form`; complete



DP1 Provide Situated Support Based on Available Information

- *In-situ tips and suggestions* tailored to the current status of the plan
- *Tooltip showing source of tips* upon hovering over highlighted phrases
- *Downward/upward graph traversal mechanisms* to collect & apply relevant information
- *Reasoning based on children node information* when generating tips & suggestions

DP2 Engage Users Throughout the Planning Process

- *Initially hidden suggestions* until the user clicks on the expand button
- *Multiple suggestions* inducing user agency by requiring comparisons

DP3 Keep Users Aware of the Big Picture

- *Overview Panel* summarizing the progression each section
- *Table of Contents* providing a structured summary of the overall plan
- *Traversal based on depth-first search* that keeps question flow focused
- *No traversal on cross-sectional edge* to keep question flow from drifting away

DP4 Assume Minimal Technical Background

- *Interface that models web survey forms* to leverage familiarity & minimize entry barrier
- *Intuitive interactions on hidden underlying graph* to avoid overwhelming users

Figure 4: Design decisions made according to each of the four design principles.

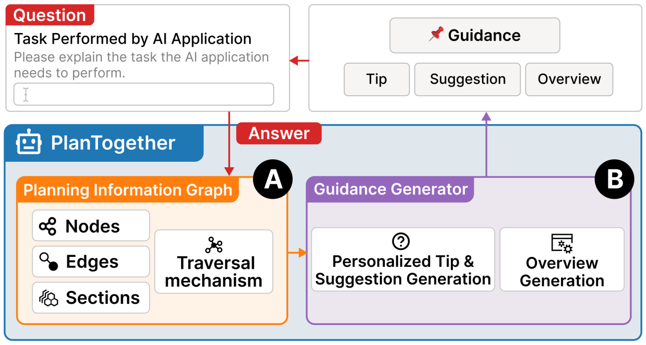

Question
Task Performed by AI Application
Please explain the task the AI application needs to perform.
Guidance
Tip
Suggestion
Overview
PlanTogether
Answer
Planning Information Graph
A
Nodes
Edges
Sections
Traversal mechanism
Guidance Generator
B
Personalized Tip & Suggestion Generation
Overview Generation

Figure 5: The constituents of PlanTogether and the information flow between the constituents.

list and examples of answer forms available in the supplemental material). The Type of a node is either *core*, indicating that it originates from AINeedsPlanner [65], or *supplemental*, indicating that it has been system-generated to provide reasoning support. The `Embedding Vector` allows the system to efficiently detect duplicate nodes when generating supplemental nodes.

An *edge* of the Planning Information Graph Figure 7B) indicates that the `Child Node` can help reasoning for the `Parent Node`. The `Type` of an edge is either *prerequisite*, indicating that the child is required to answer the parent (e.g., 'Domain Metrics' is required to answer 'Domain Metric Measurement Method' in Figure 7B) or *contextual*, indicating that the child provides important context for reasoning about the parent.

Detailed specification of the constituents of the Planning Information Graph is available in Appendix B.

4.2.1 Planning Information Graph Traversal. The traversal on the Planning Information Graph is designed to form a flow that can best support the user's reasoning process by quickly building up information that the Guidance Generator can use when assisting the client (DP1), while also keeping the flow focused (DP3).

To keep the flow focused, the system uses an adaptation of the depth-first search tree traversal algorithm; the algorithm prioritizes relevant and helpful information for the current question. Overall, the traversal happens *downward* to the child either when the system needs additional information from the user to help answer the current question; the traversal happens *upward* to the parent when the system has obtained significantly more information from the user to reattempt answering the original question. Based on these downward and upward mechanisms, the system operates on each node in five steps (Figure 8). The system (1) first fulfills any prerequisites before (2) presenting the current node. If the user is unable to answer the current node, the system (3) traverses to the core information children nodes before (4) attempting again. Until the user is able to answer the current node or decides to return to the question later, the system (5) uses the reasoning ability of an LLM to generate novel supplemental children nodes that include questions that can further help reason about the current node. During the traversal, although there are edges between sections, we limit the traversal to nodes in the same section to avoid the question flow from drifting away, although the system freely uses information from other sections in other sections to generate tips and suggestions.

Detailed Planning Information Graph traversal algorithm is available in Appendix C.

4.2.2 Planning Information Graph Construction. To identify the core nodes and the sections of the core nodes of the Planning Information Graph, we utilized the taxonomy of AI experts' information needs from the client and AINeedsPlanner, the client workbook resulting from the taxonomy, presented by Kim et al. [65]. We first obtained the core nodes from the presented taxonomy and wrote down the questions for each node based on the wordings in the workbook and iterated on the wordings through pilot studies. Next, we sectioned the core nodes roughly based on the sections of the AINeedsPlanner with minor modifications based on the taxonomy and discussions among the authors. For the edges, two authors independently labeled each pair of nodes to determine the presence and type of information dependency and merged the edge labels through discussions, involving the third author for mediation.

The resulting Planning Information Graph includes 58 nodes and 327 edges across 7 sections: 'Project Objective', 'Dataset', 'Budget, Resources & Framework', 'Client Situation', 'Model Needs', 'Agreement Terms', and 'Others'.

Since the constructed Planning Information Graph may have imperfections, we built a mechanism for the graph to evolve over continued use, forming new edges and removing extraneous edges. We include the constructed Planning Information Graph and the graph evolution mechanism in the supplemental material.

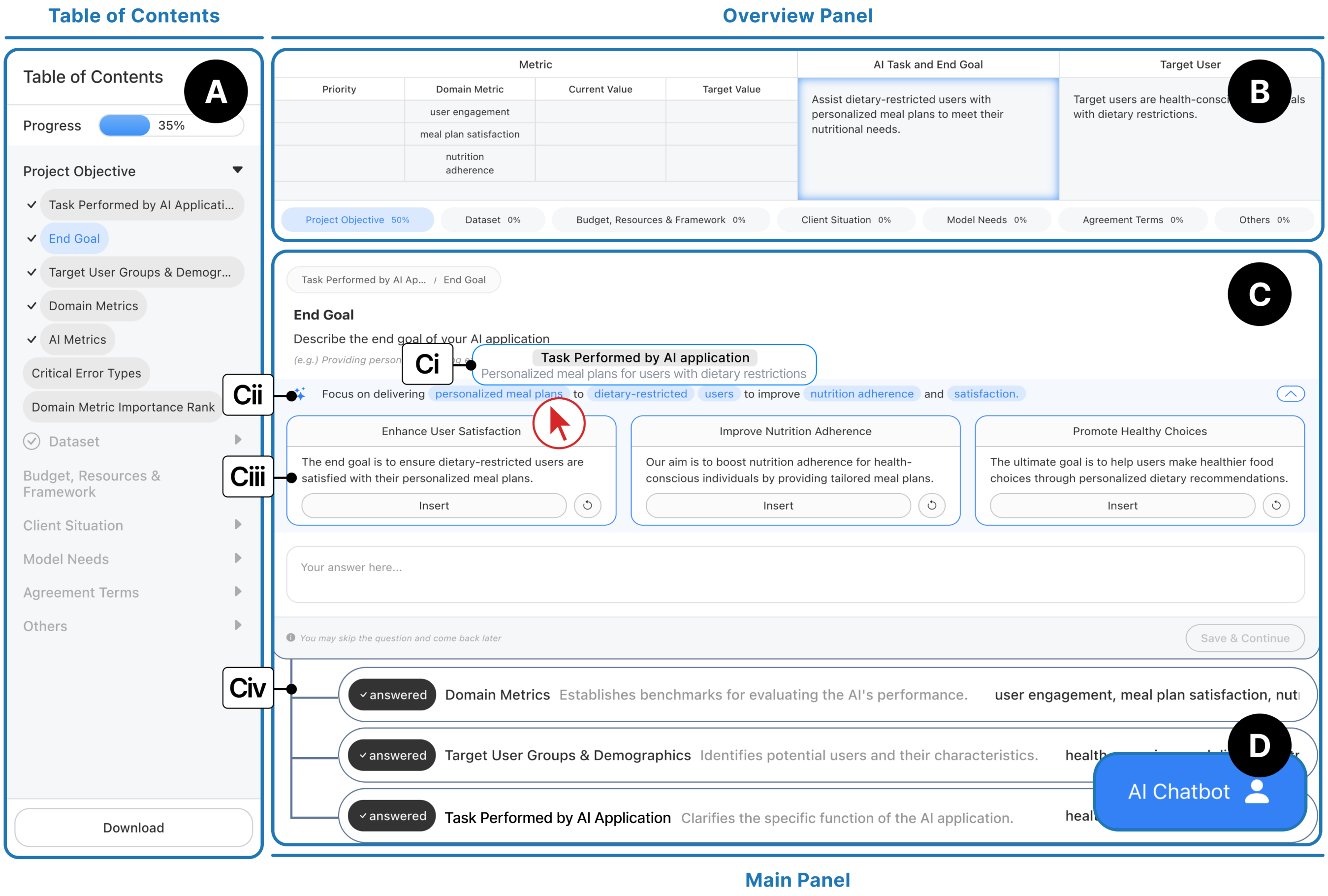


Table of Contents

Progress 35%

Project Objective
- ✓ Task Performed by AI Applicati...
- ✓ End Goal
- ✓ Target User Groups & Demogr...
- ✓ Domain Metrics
- ✓ AI Metrics
- Critical Error Types
- Domain Metric Importance Rank

Dataset

Budget, Resources & Framework

Client Situation

Model Needs

Agreement Terms

Others

Metric				AI Task and End Goal	Target User
Priority	Domain Metric	Current Value	Target Value	Assist dietary-restricted users with personalized meal plans to meet their nutritional needs.	Target users are health-consci als with dietary restrictions.
	user engagement				
	meal plan satisfaction				
	nutrition adherence				

Project Objective 50% | Dataset 0% | Budget, Resources & Framework 0% | Client Situation 0% | Model Needs 0% | Agreement Terms 0% | Others 0%

Task Performed by AI Ap... / End Goal

End Goal

Describe the end goal of your AI application

(e.g.) Providing person

Task Performed by AI application
Personalized meal plans for users with dietary restrictions

Focus on delivering personalized meal plans to dietary-restricted users to improve nutrition adherence and satisfaction.

Enhance User Satisfaction	Improve Nutrition Adherence	Promote Healthy Choices
The end goal is to ensure dietary-restricted users are satisfied with their personalized meal plans.	Our aim is to boost nutrition adherence for health-conscious individuals by providing tailored meal plans.	The ultimate goal is to help users make healthier food choices through personalized dietary recommendations.
Insert	Insert	Insert

Your answer here...

You may skip the question and come back later

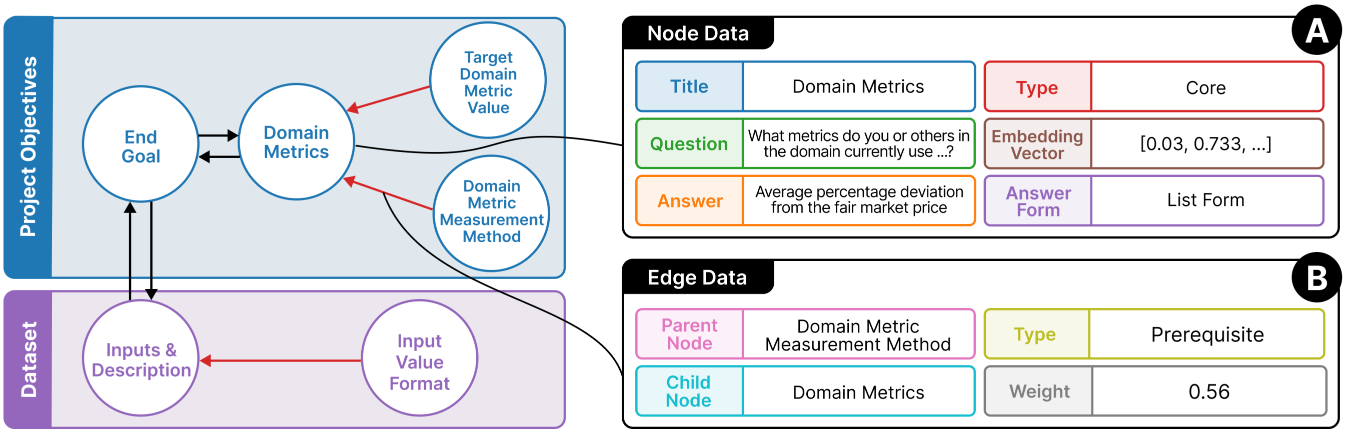
Save & Continue

✓ answered Domain Metrics Establishes benchmarks for evaluating the AI's performance. user engagement, meal plan satisfaction, nut

✓ answered Target User Groups & Demographics Identifies potential users and their characteristics. health

✓ answered Task Performed by AI Application Clarifies the specific function of the AI application. heal

Figure 6: The PlanTogether interface. The interface, modeled after typical survey form interfaces, includes the Table of Contents (A), the Overview Panel (B), and the Main Panel (C). It also includes an AI chatbot (D) that can provide further support.



Node Data

Title	Domain Metrics	Type	Core
Question	What metrics do you or others in the domain currently use ...?	Embedding Vector	[0.03, 0.733, ...]
Answer	Average percentage deviation from the fair market price	Answer Form	List Form

Edge Data

Parent Node	Domain Metric Measurement Method	Type	Prerequisite
Child Node	Domain Metrics	Weight	0.56

Figure 7: A subset of the Planning Information Graph. A directed edge from a parent node to a child node indicates a prerequisite (→) or a contextual relationship (→). (A) and (B) show the data structures of a node and an edge in the graph, respectively.

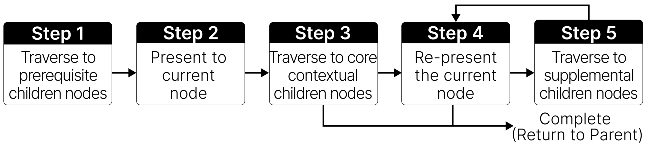

Step 1
Traverse to prerequisite children nodes
Step 2
Present to current node
Step 3
Traverse to core contextual children nodes
Step 4
Re-present the current node
Step 5
Traverse to supplemental children nodes
Complete (Return to Parent)

Figure 8: The five steps of graph traversal.

4.3 Guidance Generator

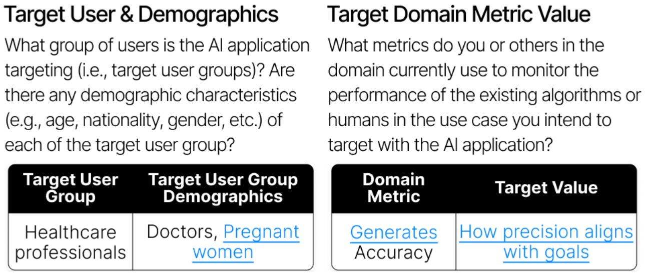
The Guidance Generator (Figure 5B) is a module that uses the Planning Information Graph (Figure 5A) to create the necessary guidance for the client as they fill out the pre-collaboration plan. The module collects and utilizes information available in children nodes to generate tips and suggestions tailored to the user's plan (DP1). information in each section to provide a holistic overview of the progression (DP3).

4.3.1 Personalized Tip & Suggestion Generation. When presenting the question to the user, the Guidance Generator prompts an LLM with the node question and (question, answer) pairs from all answered children nodes to generate a *tip* and *suggestions* for the client. By basing the tip and suggestions on the Planning Information Graph, the system can provide personalized guidance for easier comprehension and internalization of the question contents and expectations.

When generating tips, the module instructs an LLM to also explicitly extract the references to the neighboring nodes. The interface shows these references to the user to transparently show its information source (DP1) (Figure 6Ci) so that the user can contextualize how the information in the neighboring nodes contributes to answering the current question.

When generating suggestions, the module includes the generated tip in the input and asks an LLM to generate up to three suggestions that have sufficient diversity. The display of the multiple suggestions is a deliberate design choice to implicitly pressure the user takes agency in perusing and comparing the suggestions before either choosing one or writing their own answer (DP2). When the client requests a new suggestion by clicking on the refresh button (↻), the module again uses an LLM to generate a suggestion that is different from the previously presented suggestions.

4.3.2 Overview Generation. The Guidance Generator utilizes the information provided in the nodes to generate contents for the Overview Panel (Figure 6B) to help clients see their progression through pre-collaboration planning (DP3). When we designed the overview panel, we linked each piece of information displayed on the panel and the core information nodes, allowing the Guidance Generator to draw information from the corresponding nodes, while highlighting the information currently being worked on to help the user link information and locate updates to the overview [63, 68]. To fill up the Overview Panel, the module *copies* simple answers (e.g., 'Domain Metrics' in Figure 6B), *summarizes* longer answers or answers across multiple nodes with an LLM (e.g., 'AI Task and End Goal' in Figure 6B), or *simulates* concrete example data based on data descriptions with an LLM (e.g., 'Input' of the dataset).



Target User & Demographics

What group of users is the AI application targeting (i.e., target user groups)? Are there any demographic characteristics (e.g., age, nationality, gender, etc.) of each of the target user group?

Target User Group	Target User Group Demographics
Healthcare professionals	Doctors, Pregnant women

Target Domain Metric Value

What metrics do you or others in the domain currently use to monitor the performance of the existing algorithms or humans in the use case you intend to target with the AI application?

Domain Metric	Target Value
Generates Accuracy	How precision aligns with goals

(a) Consistency Error **(b) Relevance Error**

Figure 9: Examples of erroneous suggestions generated by PlanTogether. (a) The suggestion *"Pregnant women"* in 'Target User Group Demographics' is inconsistent with the 'Target User Group' *"Healthcare professionals"* (consistency error). (b) The suggestion *"Generates Accuracy"* for 'Domain Metric' and the suggestion *"How precision aligns with goals"* for 'Target Value' are irrelevant (relevance error).

5 Preliminary Evaluation

Providing suggestions is a core feature of our system that not only fully leverages the information dependencies present in Planning Information Graph and the reasoning capabilities of LLMs, but also greatly affects the planning process. Therefore, before a full-scale user study to obtain a holistic evaluation, we decided to get a detailed understanding of the suggestions; we conducted an IRB-approved preliminary evaluation of the system to understand (1) whether PlanTogether generates accurate suggestions and (2) whether the domain experts would be able to discern the erroneous suggestions generated by the system and mitigate the negative effects on the final plan.

5.1 Method

We designed a minimal version of the system that implements only the core features of PlanTogether (i.e., tips, suggestions, overview); the minimal system excluded the AI chatbot. To understand whether domain experts can discern errors included in the suggestions, we included only one suggestion instead of three and had the participants first decide on whether to accept or reject the given suggestion before making edits.

We recruited 12 domain experts fluent in English through online communities within KAIST and personal networks. For each participant, we required at least a bachelor's degree in their domain and an AI application idea in their domain. The years of domain experience ranged between 1-15 years (median = 3.5 years) and the domains included fields such as telecommunication, medicine, and fashion (full list in the supplemental material). After a system tutorial, participants filled either the 'Project Objectives' [46, 65, 109, 116] or the 'Dataset' section [34, 65, 116, 118, 122], known in prior work as essential yet difficult sections for clients. We performed the study in English.

5.2 Results

Accuracy of Suggestions. Throughout the 12 sessions, we identified a total of 128 instances of suggestions generated for the participants (Table 1). Based on an analysis by two authors with third author mediation, 103 instances (80%) addressed the question with answers consistent with the contents of the rest of the pre-collaboration plan. However, 14 instances (11%) contradicted other parts of the pre-collaboration plan (*consistency error*; example in Figure 9a), and 2 instances (2%) provided answers irrelevant to the given question (*relevance error*; example in Figure 9b). An additional 9 instances (7%) included both types of errors (*consistency + relevance error*).

Client's Ability to Discern Errors. In general, the participants were keen on the errors, either revising or rejecting 23 (92%) of the 25 erroneous suggestions. There was an imbalance in how well the clients discerned the errors; participants did not accept any of the suggestions that contradicted their plans (i.e., consistency error, consistency + relevance error; Table 1 right column), while they accepted suggestions irrelevant to the question that were consistent with the plans (relevance error; Table 1 bottom left). This suggests that clients are able to discern the most frequent type of errors generated by the system, hence keeping the harms from erroneous suggestions under check.

The results of the preliminary evaluation indicate that PlanTogether shows desired behaviors; it is able to generate mainly accurate suggestions and the users can distinguish errors well, limiting the harmful effects of errors. Whether these desired behaviors actually leads to better pre-collaboration plan is answered through the user study (Section 6).

6 User Study

To understand whether the PlanTogether's graph-based features and the overview help clients during pre-collaboration planning and result in more concrete and actionable plans, we conducted an IRB-approved between-subjects study comparing the complete PlanTogether (PlanTogether condition) and a baseline system that excludes all the graph-based features (suggestions, tips, and graph-based traversals) and the overview (baseline condition). To control for the placebo effects that can arise from including AI features in the system [67], we kept the AI chatbot in the baseline condition, which is not a graph-based feature nor an overview feature. We consider the following two hypotheses:

H1 [*Better Quality of Resulting pre-collaboration Plans*] The graph-based features and the overview included in PlanTogether lead to higher-quality pre-collaboration plans that are more concrete and actionable.

H2 [*Helpfulness of PlanTogether Features*] The graph-based features and the overview included in PlanTogether provide helpful guidance to clients performing pre-collaboration planning.

We note that the system included a minor implementation issue involving traversals to prerequisite nodes, which did not significantly impact user experience, that we later revised. We include study materials in the supplemental material.

Table 1: Frequencies of errors included in the LLM-generated suggestions generated during the preliminary evaluation and frequencies of user decisions on the suggestions. The table shows the counts in the format of 'total (accepted/revised/rejected)'

		Suggestion Consistency	
		Consistent	Inconsistent
Question Relevance	Relevant	103 (47/12/40)	14 (0/2/12)
	Irrelevant	2 (2/0/0)	9 (0/0/9)

6.1 Participants

We recruited 18 participants fluent in English through posts on online communities within KAIST as well as multiple other institutions (Table 2). To model the real-world use cases of these pre-collaboration planning systems, we recruited domain experts with ideas about applying AI to tasks in their domain. As a way of ensuring domain expertise, we required that the participants have at least a bachelor's degree in their domain, not related to computer science or artificial intelligence. None of the participants had prior experience in participating in prior AI application development projects; along with the between-subjects design, this removes the confounding effects of prior experience. We additionally enforced that they have some initial thoughts on applying AI within their domain that they have not consulted or collaborated with AI experts about (i.e., have not proceeded past the pre-collaboration phase). To capture the space of the diverse domains, we prioritized diversification of AI application domains.

The study lasted approximately 5 hours, and each participant received a compensation of 150,000 KRW (≈ 110 USD) via direct deposit. Due to the long duration of the study and to streamline the process, we conducted the study in four group sessions, each including 4 or 5 randomly assigned participants. We assigned the PlanTogether condition to two of the group sessions (P1-P9) and the baseline condition to the other two group sessions (P10-18).

6.2 Procedure

We conducted the study in English through Zoom [143] to facilitate participation from individuals beyond our geographical area. After an introduction and obtaining consent, participants completed a *pre-study survey* including questions on their domain background, prior experience in participating in AI application development, and the AI application idea.

Next, we gave a *tutorial* on the version of the system the participants were assigned to. We explained the system interfaces and their respective functionalities. Participants followed the tutorial along and then explored the interface on their own. We answered any questions about the system during this step.

Then, we proceeded with the *main study*, in which the participants tool the role of clients performing pre-collaboration planning ahead of beginning collaborations with AI experts. Participants individually completed the entire pre-collaboration plan over three hours using the system corresponding to their assigned condition. During the pre-collaboration planning, we asked the participants to 'think-in-writing' on Google Docs [21] to ensure that we capture their independent thoughts around the experience during the process. We explicitly mentioned that they could use tools outside of our system, including search engines and LLM-based chatbots. During this process, we refrained from answering questions related

Table 2: An overview of user study participants and their AI application ideas, with the application domains in brackets.

Part.	Dom. Exp.	Condition	AI Application Idea
P1	3 yrs	PlanTogether	[Culinary Arts / Economics] Creating new and creative recipes or plating designs tailored to the unique themes of different restaurant.
P2	3 yrs	PlanTogether	[Automotive] Predicting vehicle weight to improve monitoring, enhancing driving efficiency and user safety.
P3	10 yrs	PlanTogether	[Education] Converting three case studies on negotiation scenarios into clear, illustrative cartoons.
P4	10 yrs	PlanTogether	[Business] Guiding the launch of an interactive internet platform for strategic insights and user retention.
P5	5 yrs	PlanTogether	[Marketing] Analyzing the performance of a brand's social media content to improve audience targeting and provide insights for developing future content.
P6	3 yrs	PlanTogether	[Retail] Generating topics for sales associates and building presentation outlines for retails events to enhance customer-oriented e-commerce experiences.
P7	6 yrs	PlanTogether	[Chemical Engineering] Providing comprehensive explanations of polymer synthesis techniques for research purposes.
P8	1 mth	PlanTogether	[Sports] Predicting the best battling order for a baseball team to increase chances of the team's winning.
P9	4 yrs	PlanTogether	[Medicine] Assisting in medical imaging and diagnostic procedures, supporting clinical decision-making, enhancing precision in robotic surgeries.
P10	3 yrs	Baseline	[Dietetics] Creating a personalized dietary plans for overweight women, tailored to their weight, to help them achieve a healthy weight.
P11	2 yrs	Baseline	[Delivery] Building an AI chatbot for delivery applications that understands the context of questions and answers various topics beyond present ones.
P12	5 yrs	Baseline	[Economics] Evaluating the credit risk of corporate borrowers aiding loan officers in making lending decisions.
P13	5 yrs	Baseline	[Environmental Engineering] Predicting future climate and risks, providing information to guide climate policy and help companies make informed decisions.
P14	4 yrs	Baseline	[Dentistry] Analyzing brushing habits through videos to recommend better brushing techniques and hygiene products.
P15	4 yrs	Baseline	[Medicine / Engineering] Generate death certificated based on hospital circumstances upon a patient's death.
P16	5 yrs	Baseline	[Economics] Building a conversational app to relieve stress from personal relationships and enhance mental health.
P17	2 yrs	Baseline	[Finance] Predicting the optimal times to buy or sell stocks, aiming to maximize profits.
P18	4 yrs	Baseline	[Education] Generating math problems aligned with the local high school curriculum and tailored to students' levels.

to the content of the pre-collaboration plan. After completing the pre-collaboration form, participants filled out a *post-study survey*, reflecting on their experience.

Finally, we conducted a semi-structured one-hour group interview with participants in the same group, allowing them to build on each other's answers to help recall the three-hour long main study. During the interview, we asked about the participants' overall experience with the system, the various challenges they ran into, and how they used the system or external tools to resolve the challenges, as well as the strengths and weaknesses of the system.

After the experiment, to assess the quality of the resulting pre-collaboration plans, we shared the resulting pre-collaboration plans with 9 AI experts recruited through personal connections and through contacting various AI solutions companies. To ensure that the AI experts can assess the quality of the plans, we required at least three years of AI experience as either ML/AI engineers or AI researchers, during which they collaborated with domain experts in building AI applications. We asked each AI expert to rate 6 pre-collaboration plans (3 per condition) that had previously been anonymized and stripped of any information indicative of which system was used while writing the plan. The AI experts rated (on a 5-point Likert scale) and provided opinions based on the overall quality, executability, consistency, clarity, and incorporation of the client's domain expertise. We include detailed AI expert information and the evaluation rubric and the individual AI expert ratings in the supplemental material.

6.3 Results

Assessing H1: Better Quality of Resulting Pre-Collaboration Plans. In general, AI experts rated the overall quality of the pre-collaboration plans generated in the PlanTogether condition (median = 4) significantly higher than that in the baseline condition (median = 3) (Mann-Whitney $U = 484.0$, $p = 0.032$; each plan's median overall quality in Figure 10a). AI experts often viewed pre-collaboration plans generated in the PlanTogether condition as clearly defined and complete (P2, 6-9), although occasionally including issues around adequacy and relevancy of the idea (P1, 3, 4); pre-collaboration plans generated in the baseline condition were often missing key information (P10, 11, 16-18).

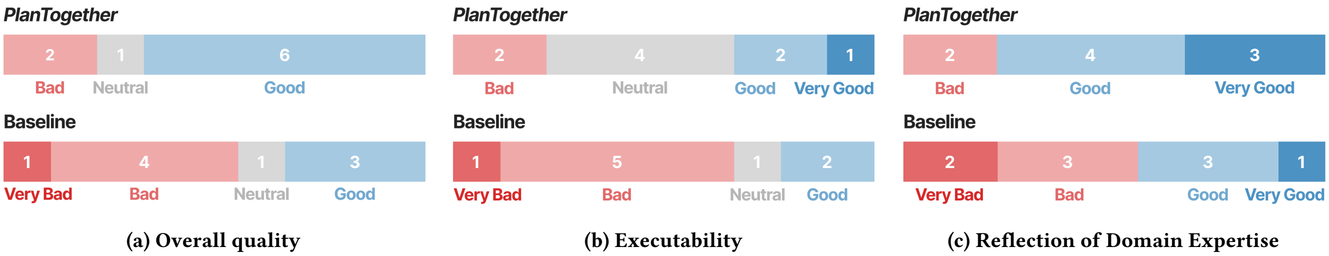
Diving deeper, AI experts reported plans generated in the PlanTogether condition (median = 3) significantly more executable than those generated in the baseline condition (median = 2) (Mann-Whitney $U = 477.5$, $p = 0.045$; each plan's median executability in Figure 10b), requiring smaller revisions to arrive at a concrete development plan for the AI expert. AI experts deemed the majority of the plans generated with PlanTogether as executable within one or two iterations with clear ideas about specific gaps to address (P2, 5, 7-9); they deemed the majority of plans generated with the baseline system as requiring significant iterations without a clear sense of a discussion direction (P10, 11, 15, 16, 18).

Furthermore, AI experts thought that plans generated in the PlanTogether condition (median = 4) significantly better-reflected the client's domain expertise than those generated in the baseline condition (median = 3) (Mann-Whitney $U = 502.0$, $p = 0.014$; each plan's median domain expertise reflection score in Figure 10c). For example, the AI experts noted that most pre-collaboration plans generated with PlanTogether captured the client's domain expert well in the detailed domain metrics (P1, 3-9), whereas only three of the plans generated with the baseline system captured the client's domain expertise well (P11, 12, 15). We hypothesize that our system helped participants comprehend the question and its expectations through personalized examples so that they could reflect their domain expertise well.

Hence, our findings are in-line with H1; PlanTogether helps produce higher-quality and actionable pre-collaboration plans that reflect the client's domain expertise.

Assessing H2: Helpfulness of PlanTogether Features. Overall, participants in the PlanTogether reported that the graph-based features and the overview PlanTogether were helpful; 8 of 9 participants explicitly mentioned the usefulness of the tips and suggestions and 6 of 9 participants specifically mentioned the usefulness of the overview.

Looking at the individual components, the tips and suggestions made major contributions to the pre-collaboration planning experience early in the client's planning flow. When the participant moved on to a question, the tip – with its references to the previously answered questions – and the suggestions position the current question in the context of the previously answered questions (P1,



(a) Overall quality (b) Executability (c) Reflection of Domain Expertise

Figure 10: The distributions of the median ratings for each of the pre-collaboration plans received for (a) the overall quality ($\alpha_{\text{Krippendorff}} = 0.59$), (b) executability ($\alpha_{\text{Krippendorff}} = 0.46$), and (c) the reflection of the client's domain expertise ($\alpha_{\text{Krippendorff}} = 0.66$). The top and bottom bars show the median rating distributions for PlanTogether and the baseline, respectively. The color of the blocks encodes the median score of each pre-collaboration plan (🟥 'very bad' (1) / 🟥 'bad' (median = 2) / ⬜ 'neutral' (median = 3) / 🟦 'good' (median = 4) / 🟦 'very good' (median = 5)). The number inside each block shows the number of pre-collaboration plans for each score category.

3). P3 stated that *"The AI suggestions based on their previous answers made them more relevant and specific to their needs."* The provided suggestions tailored to the participant's current plan formed a basis for comprehension of the question especially when the question required having a grasp of the AI concepts and guided towards concretization of the client's answers (P2, 4, 6, 9). For instance, P6 mentioned *"The personalized suggestions helped me precisely grasp the intent behind the questions."* When answering the question on 'Critical Error Types', the participant saw the tip *"Consider including how these errors might affect achieving the end goal of transforming luxury shopping experiences ⋯."* and the suggestion *"Algorithm errors could jeopardize personalized experiences for fashion-forward millennials, decreasing their likelihood of repeated purchases."* along with two others. The participant deduced that the ideal answer for the question not only describes the type of error, but also explains the potential consequences of the error. The participant found the suggestion satisfactory and accepted it.

Even in the baseline condition, suggestions and examples were deemed important in answering a given question. In particular, 6 of 9 participants utilized the provided AI chatbot to generate examples and directly copy-pasted them into their answers. However, participants expressed dissatisfaction with the generated suggestions stating that they do not understand their context and provide generic and shallow examples (P15, 18). Our system addresses these shortcomings of AI chatbots in generating suggestions by not only using carefully engineered prompts but also utilizing the graph structure and surfacing the information dependencies to the user. Moreover, despite the inclusion of the AI chatbots in the baseline, a number of participants did not use the chatbot, stating that they still rely on traditional search engines and that AI chatbots are still *'novel technology'* that they do not understand how to integrate into their workflow (P10, 16). All participants utilized the tips and suggestions in the PlanTogether condition, suggesting that our design of the system and the interactions successfully lowered the mental barrier of using novel technology.

The overview in our system guided the participants through planning, helping them maintain a sense of progress (P7, 9) as well as organize and structure their ideas throughout the process (P2, 4, 8); participants in the baseline condition complained about the difficulty in tracking their progress and a lack of understanding of the big picture (P10, 14, 15, 17).

Lastly, we observed that participants actively navigated the questions with our system (avg 103.4 total question visits) compared with the baseline (avg 59.6 total question visits). In particular, participants frequently navigated to questions outside the default flow using graph-based navigation the Table of Contents with our system (avg 18.9 times); participants did not navigate as actively in the baseline condition (avg 3.9 times). We believe that the references to other questions in the tips, graph-based navigation, and the overview features of our system successfully instilled a connected holistic picture of the plan for the user, enabling active navigation. For instance, while answering a question about 'Client-Target User Relationship', P9 saw 'Target User Groups & Demographics' as a relevant question they had already answered in the graph-based navigation feature; P9 decided to revisit and review the answer they provided for the question.

In sum, our findings are supportive of accepting H2; the graph-based features and the overview feature of PlanTogether provide helpful guidance to the clients during pre-collaboration planning.

7 Discussion

In this section, we discuss (1) the adaptation of tips and suggestions over the course of pre-collaboration planning, (2) the level of reliance of clients on tips and suggestions, (3) the role of tips & suggestions, AI chatbot, and search engines, and (4) benefits of using PlanTogether for the discussion phase.

7.1 Adaptation of Tips & Suggestions over the Course of Pre-Collaboration Planning

Over the course of pre-collaboration planning, the Planning Information Graph becomes gradually populated. Because the data available for the system to link and reason about increases over time, the quality of the tips and suggestions also improves over time. This quality improvement was apparent for the user study participants (Section 6) (P2, 5). For the AI application idea of P5, for example, when the relevant questions are mostly unanswered, the system suggests *"Maximize audience interaction by analyzing social media content performance"* for 'End goal'; once most of the relevant questions are answered, the system suggests *"Empower content creators and marketing managers with data insights to refine their content strategies and increase engagement rates,"* which better matches the client's true AI application intent by concretely linking to the answers on target users and domain metrics.

However, this improvement in quality over the course of pre-collaboration planning also implies that the user experience with the tips and suggestions toward the beginning could fall short due to the LLM making assumptions based on limited data. The system produces concrete suggestions even before the client has had a chance to fully reflect their intents into the pre-collaboration plan. If the client's unexpressed intents and the system's suggestion directions mismatch, the client may feel over-guided (P6, 7, 9). Hence, suggestion generation could take a more conservative approach, refraining from giving too concrete suggestions early on. Furthermore, as P2 suggested, the system can return to these early questions later to further leverage the quality improvements over the course of planning.

Looking at the change in the quality from a different perspective, the dependency of tips and suggestions on the previously answered questions also means that incorrect or inconsistent answers to questions can lead to erroneous suggestions later. However, it was notable that P1 and P6 were able to conversely utilize the noticeable drops in the quality of the suggestions and the links to previous questions given in the tips to backtrack the questions they answered incorrectly to go back and revise them. This observation could imply that the system may also be able to conversely utilize the changes in suggestion quality to detect potential issues in the pre-collaboration plan. In the current implementation, if the client makes changes in the plan later on, they are can use the options at the bottom of the Main Panel to traverse to questions with information dependencies and leverage the improved tips and suggestions from the additional information to modify their answers. However, they would need to manually recursively review and revise the answers; quality monitoring and detecting issues could lead to the system feature pinpointing answers that need review and revision based on the changes made in the plan.

In sum, the adaptive nature of the tips and suggestions over the course of pre-collaboration planning enables personalized support as more information becomes available, but also introduces special consideration needs when reliable information is insufficient. Yet, it opens up opportunities for inconsistency detection.

7.2 Level of Reliance of Clients on Tips & Suggestions

Our study (Section 6) suggests that clients will rely on the tips and suggestions generated by the system for guidance. This naturally gives rise to a question: would the clients *over-rely* on our system for their answers and unthinkingly accept suggestions without independent thinking? An analysis of how the user study participants interacted with the provided suggestions indicates that they were actively engaging with the provided suggestions and that the answer to the question is 'no'; they refreshed, revised, and/or rejected the initially given suggestions in 134 of the 215 suggestions that we identified, instead of simply selecting and going with one of the initially provided suggestions.

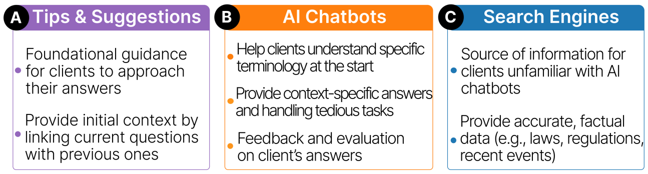

A Tips & Suggestions
Foundational guidance for clients to approach their answers
Provide initial context by linking current questions with previous ones
B AI Chatbots
Help clients understand specific terminology at the start
Provide context-specific answers and handling tedious tasks
Feedback and evaluation on client's answers
C Search Engines
Source of information for clients unfamiliar with AI chatbots
Provide accurate, factual data (e.g., laws, regulations, recent events)

Figure 11: Complementary roles of (A) tips & suggestions, (B) AI chatbots, and (C) search engines for pre-collaboration planning.

A major contributing factor may ironically the errors included in the system-generated suggestions. Based on our preliminary evaluation (Section 5), we hypothesize that the easy-to-discern errors in the suggestions can quickly help the clients form a critical and independently thinking mindset instead of a blind trust towards the system [62]. In particular, P1 admitted that they were initially receptive to the suggestions given by the system without giving careful thought. However, upon seeing an erroneous suggestion early in the planning, they became critical about ruthlessly accepting the provided suggestions, although the system was accurate enough to often provide meaningful suggestions.

We believe that the presentation of multiple suggestions instead of one further promotes independent thinking [72]. The client is not only given a broader sample of possible answers, but must carefully compare them to select one.

7.3 Role of Tips & Suggestions, AI Chatbot, and Search Engines

Other than the tips and suggestions generated by our system, participants actively used the provided AI chatbot or other alternative chatbots (10 of 18) as well as search engines, including Google [38] (6 of 18). Through a deeper analysis, we find that the three tools complement each other in supporting the client's planning process (Figure 11). We have already seen in Section 6 that the tips and suggestions generated by PlanTogether assist the client by positioning the current question in the context of previously answered questions and forming a basis for the current answer (Figure 11A).

For those who utilized the AI chatbot, the chatbot played a distinctive role in the planning process (Figure 11B). The AI chatbot initially plays a minor role in helping the client understand the terminology in the question (P1, 11-14). The more pronounced role of the AI chatbot is after the initial phase; clients seek help from the AI chatbot when they have questions that require specific context in their plan (P5) or when they need to perform tedious tasks (P12, 14). Moreover, clients also often seek evaluation and feedback about their answers through the AI chatbot (P4, 5, 11, 12).

Search engines not only serve as an information source for clients who are not familiar with AI chatbots, but also are crucial sources of information that the clients wish to obtain accurate information that does not include interpretation of AI chatbots (e.g., laws and regulations, information based on recent events) (P5, 9, 10, 12) (Figure 11C). The clients would reason on their own using the information they collect through search engines.

As P5 noted, while the roles of the three tools are relatively disjoint, linking the three tools and allowing information flow among the three tools would bring about a synergy between the three tools. For example, as the client converses with the AI chatbot, the tips and suggestions could be updated to reflect the conversation; the conversations with the AI chatbot can be in the context of the tips and suggestions generated by the system. This link between the tips and suggestions and the AI chatbot would not only enable smoother human-AI dialogues that are better situated in the context of previous questions, but also allow the tips and suggestions to play a role further beyond early assistance.

7.4 Benefits of Using PlanTogether for the Discussion Phase

Our user study (Section 6) shows that the graph-based features and the overview leads of PlanTogether leads to more actionable plans; it can shorten the discussion phase with the AI expert that would follow the pre-planning phase for iteration of the plan into one that the AI expert would develop. We note that our system further advances an already-improved baseline (AINeedsPlanner [65]) for improving plan executability.

In addition, similarly to AINeedsPlanner, PlanTogether could directly be used during the discussion phase as a discussion guide and boundary object. The AI expert can use the system not only to ensure coverage of all of their information needs to come up with a development plan as with AINeedsPlanner [65], but also to ensure that domain experts can fully comprehend the ongoing discussions by using the tips and suggestions generated by the system. Adding graph-based system features such as recursive discussion guides that marks nodes neighboring nodes that have been iterated on could guide revision discussions in the discussion phase.

8 Limitations and Future Work

Further Validation of the Limitations of the User Study. While we believe our findings of the user study are generalizable, limitations exist in study design. For example, for a controlled experiment that we can monitor, we had study participants perform pre-collaboration planning for 3 hours. Although this was sufficient for all participants, pre-collaboration planning in the real world usually occurs over longer periods of time over multiple sessions. In addition, we required that study participants hold at least a bachelor's degree in their domain to ensure a level of domain expertise. Yet, it is not the only way for an individual to have domain expertise and even a person without a high level of domain expertise could have needs for AI applications. Lastly, while the between-subjects study design allowed us to observe the value of our system through comparing the experiences of the participants as well as the outputs, we excluded domain experts with prior collaboration with AI experts to control the effect of prior experience. Since the exact effects of the decisions are unknown, further validation would ensure the generalizability of our findings.

Introduction of Feedback Features. While performing the pre-collaboration planning in the user study (Section 6), many participants sought feedback from the AI chatbot (P1, 5-7, 11-14). However, they often found the provided feedback unsatisfactory because of its vagueness (P5). Based on the need for feedback, future work can explore ways of automatically identifying strengths and weaknesses of a pre-collaboration plan, while suggesting revisions and triggering further thought. For instance, the system can perform both local evaluations of whether a provide response answers a given question as well as global evaluations of whether the response is consistent with the rest of the plan. The system could also include an interactive visual feedback feature based on a simulation of the AI application based on the provided information; the system would include a schematic prototype of the resulting AI application that dynamically evolves with additional information. LLMs or AutoML techniques [42] could power the prototypes.

Personalization of the System. While all other participants agreed on the helpfulness of PlanTogether in pre-collaboration planning, P7 showed reservations about the helpfulness for their AI application idea, due to the high degree of specialization needed for the domain expertise involved in the idea (See Table 2). The participant admitted that they utilize a specialized LLM that has been fine-tuned with academic papers and other documents in their domain. Based on this example case, PlanTogether will need to be extended to support the replacement of the built-in LLM with customized LLMs or support easy fine-tuning of the LLM for the system to be truly broadly applicable. Moreover, while we did not observe major differences of plan quality or user experience across underlying AI task types (e.g., classification, object recognition, generative task), detailed additional experiments would shed light on potential adaptation of the questions and support based on the underlying AI task. Furthermore, the Planning Information Graph is shared across users and evolve through their continued use. However, the ideal Planning Information graph may be different for each AI application idea (e.g., ideas dealing with sensitive data). A possible approach for graph personalization would be to keep a shared *base graph* and build *delta graphs* for each personalization dimension, which are updated with continued use; the system would combine the base graph and the suitable delta graphs for each user.

Real-World Deployment of PlanTogether. Although our user study isolates the core aspects of pre-collaboration planning in client-AI expert collaborations, there are nuances to real-world collaborations. The nuances that can affect real-world collaborations include involvement of stakeholders other than AI experts and domain experts (e.g., UI/UX designers, managerial roles, legal teams) [64, 116], the client's and AI expert's willingness to learn about AI and the domain, respectively [64], and various corporate and regional restrictions governing AI [65]. Future work can deploy PlanTogether in the real world to understand the nuances that can affect pre-collaboration planning and attempt to build the understanding into future planning support systems. Yet, due to legal information secrecy agreements between clients and AI experts in real-world collaborations, observations of real-world deployments will need to carefully navigate partnerships with multiple AI solutions companies and freelancers.

9 Conclusion

AI application planning commonly occurs as a collaboration between clients and AI experts. The clients in these collaborations outline their needs and expectations into a *pre-collaboration plan* prior to holding the initial discussions with AI experts, but it is often a difficult process that fails to yield concrete and actionable plans due to the client's lack of understanding of AI experts' information needs and knowledge barriers around AI-related information. To address these hurdles and lead to pre-collaboration plans of higher quality, we introduce PlanTogether, an information-graph based system that guides clients through pre-collaboration plans. The system comprises (1) the *Planning Information Graph Manager* that controls the order of questions generated from the Planning Information Graph, and (2) the *Guidance Generator* that generates tips and suggestions to help clients answer each question as well as overviews to help clients comprehend their progression through the pre-collaboration plan. Based on a user study, the graph-based features and the overview included in PlanTogether are able to provide helpful guidance to the clients so that clients can yield more actionable pre-collaboration plans that better capture the client's domain expertise. Our work is among the early efforts to support AI application planning during client-AI expert collaborations; we hope to see research continue.

Acknowledgments

The authors thank the members of KAIST Interaction Lab (KIXLAB) for their constructive feedback throughout the work and piloting the user study. This work is supported by the G-CORE Research Project grant at KAIST, the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No.RS-2024-00406715), and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. RS-2024-00443251, Accurate and Safe Multimodal, Multilingual Personalized AI Tutors).

References

[1] Ghodai Abdelrahman, Qing Wang, and Bernardo Nunes. 2023. Knowledge Tracing: A Survey. *ACM Computing Surveys* 55, 11, Article 224 (Feb. 2023), 37 pages. doi:10.1145/3569576

[2] Khlood Ahmad, Muneera Bano, Mohamed Abdelrazek, Chetan Arora, and John Grundy. 2021. What's up with Requirements Engineering for Artificial Intelligence Systems?. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*. Institute of Electrical and Electronics Engineer, Piscataway, New Jersey, USA, 1–12. doi:10.1109/RE51729.2021.00008

[3] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3290605.3300233

[4] Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah D Goodman. 2024. STaR-GATE: Teaching Language Models to Ask Clarifying Questions. *arXiv preprint* 1 (2024), 25 pages. doi:10.48550/arXiv.2403.19154

[5] Annie I. Antón and Donald C. Wells. 2003. Successful Software Projects Need Requirements Planning. *IEEE Software* 20 (May 2003), 44–47. doi:10.1109/MS.2003.1196319

[6] Chetan Arora, John Grundy, and Mohamed Abdelrazek. 2024. Advancing Requirements Engineering Through Generative AI: Assessing the Role of LLMs. In *Generative AI for Effective Software Development*, Anh Nguyen-Duc, Pekka Abrahamsson, and Foutse Khomh (Eds.). Springer Nature, Cham, Switzerland, 129–148. doi:10.1007/978-3-031-55642-5_6

[7] Atlassian. 2024. Using Jira for Requirements Management. https://confluence.atlassian.com/jirakb/using-jira-for-requirements-management-193300521.html. Retrieved December 8, 2024.

[8] Muneera Bano, Rashina Hoda, Didar Zowghi, and Christoph Treude. 2024. Large Language Models for Qualitative Research in Software Engineering: Exploring Opportunities and Challenges. *Automated Software Engineering* 31, 1 (Dec 2024), 8. doi:10.1007/s10515-023-00407-8

[9] Alex Bäuerle, Ángel Alexander Cabrera, Fred Hohman, Megan Maher, David Koski, Xavier Suau, Titus Barik, and Dominik Moritz. 2022. Symphony: Composing Interactive Interfaces for Machine Learning. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 14 pages. doi:10.1145/3491102.3502102

[10] Adream Blair-Early and Mike Zender. 2008. User Interface Design Principles for Interaction Design. *Design Issues* 24, 3 (2008), 85–107.

[11] Veronika Bogina, Alan Hartman, Tsvi Kuflik, and Avital Shulner-Tal. 2021. Educating Software and AI Stakeholders about Algorithmic Fairness, Accountability, Transparency and Ethics. *International Journal of Artificial Intelligence in Education* 32 (Apr 2021), 1–26. doi:10.1007/s40593-021-00248-0

[12] David Callele, Krzysztof Wnuk, and Birgit Penzenstadler. 2017. New Frontiers for Requirements Engineering. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, Vol. 25. Institute of Electrical and Electronics Engineer, Piscataway, New Jersey, USA, 184–193. doi:10.1109/RE.2017.23

[13] Marc Canellas and Rachel Haga. 2020. Unsafe at Any Level. *Communications of the ACM* 63, 3 (Feb. 2020), 31–34. doi:10.1145/3342102

[14] Jae-Yong Chang and Il-Min Kim. 2013. Analysis and Evaluation of Current Graph-Based Text Mining Researches. *Advanced Science and Technology Letters* 42 (Dec 2013), 100–103. doi:10.14257/astl.2013.42.23

[15] Minsuk Chang, Ben Lafreniere, Juho Kim, George Fitzmaurice, and Tovi Grossman. 2020. Workflow Graphs: A Computational Model of Collective Task Strategies for 3D Design Software. In *Proceedings of Graphics Interface*. Canadian Human-Computer Communications Society, Mississauga, Canada, 114 – 124. doi:10.20380/GI2020.13

[16] Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A Review: Knowledge Reasoning over Knowledge Graph. *Expert Systems with Applications* 141 (Mar 2020), 112948. doi:10.1016/j.eswa.2019.112948

[17] Yuh-Jen Chen. 2010. Development of a Method for Ontology-Based Empirical Knowledge Representation and Reasoning. *Decision Support Systems* 50, 1 (Dec 2010), 1–20. doi:10.1016/j.dss.2010.02.010

[18] Ryan Clancy, Ihab F. Ilyas, and Jimmy Lin. 2019. Scalable Knowledge Graph Construction from Text Collections. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Hong Kong, China, 39–46. doi:10.18653/v1/D19-6607

[19] Chaoran Cui, Yumo Yao, Chunyun Zhang, Hebo Ma, Yuling Ma, Zhaochun Ren, Chen Zhang, and James Ko. 2024. DGEKT: A Dual Graph Ensemble Learning Method for Knowledge Tracing. *ACM Transactions on Information Systems* 42, 3, Article 78 (Jan. 2024), 24 pages. doi:10.1145/3638350

[20] Fernando Delgado, Stephen Yang, Michael Madaio, and Qian Yang. 2023. The Participatory Turn in AI Design: Theoretical Foundations and the Current State of Practice. In *Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*. Association for Computing Machinery, New York, NY, USA, Article 37, 23 pages. doi:10.1145/3617694.3623261

[21] Google Docs. 2024. Retrieved March 20, 2024 from https://docs.google.com/

[22] Mateusz Dolata and Kevin Crowston. 2024. Making Sense of AI Systems Development. *IEEE Transactions on Software Engineering* 50, 1 (Jan. 2024), 123–140. doi:10.1109/TSE.2023.3338857

[23] Gonzalo Aguirre Dominguez, Keigo Kawaai, and Hiroshi Maruyama. 2021. FAILS: A Tool for Assessing Risk in ML Systems. In *28th Asia-Pacific Software Engineering Conference Workshops*. Institute of Electrical and Electronics Engineer, Piscataway, New Jersey, USA, 1–4. doi:10.1109/APSECW53869.2021.00010

[24] Johann Eder, Wolfgang Gruber, and Horst Pichler. 2006. Transforming Workflow Graphs. In *Interoperability of Enterprise Software and Applications*. Springer, London, United Kingdom, 203–214. doi:10.1007/1-84628-152-0_19

[25] Lisa Ehrlinger and Wolfram Wöß. 2016. Towards a Definition of Knowledge Graphs. *SEMANTiCS (Posters, Demos, SuCCESS)* 48, 1-4 (2016), 2.

[26] Vladimir Estivill-Castro, Eugene Gilmore, and René Hexel. 2022. Constructing Explainable Classifiers from the Start—Enabling Human-in-the Loop Machine Learning. *Information* 13, 10 (Sep 2022), 464.

[27] Michael Feffer, Michael Skirpan, Zachary Lipton, and Hoda Heidari. 2023. From Preference Elicitation to Participatory ML: A Critical Survey & Guidelines for Future Research. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*. Association for Computing Machinery, New York, NY, USA, 38–48. doi:10.1145/3600211.3604661

[28] Jules Françoise, Baptiste Caramiaux, and Téo Sanchez. 2021. Marcelle: Composing Interactive Machine Learning Workflows and Interfaces. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 39–53. doi:10.1145/3472749.3474734

[29] C. Ailie Fraser, Tricia J. Ngoon, Mira Dontcheva, and Scott Klemmer. 2019. RePlay: Contextually Presenting Learning Videos Across Software Applications. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3290605.3300527

[30] Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, et al. 2023. Leveraging Large Language Models in Conversational Recommender Systems. *arXiv preprint* 1 (2023), 24 pages. doi:10.48550/arXiv.2305.07961

[31] Wenbin Gan, Yuan Sun, and Yi Sun. 2022. Knowledge Structure Enhanced Graph Representation Learning Model for Attentive Knowledge Tracing. *International Journal of Intelligent Systems* 37, 3 (Nov 2022), 2012–2045. doi:10.1002/int.22763

[32] Matthew Gardner. 2015. Reading and Reasoning with Knowledge Graphs. Retrieved March 20, 2024 from https://www.cs.cmu.edu/~mg1/thesis.pdf

[33] Merideth Gattis and Keith J Holyoak. 1996. Mapping Conceptual to Spatial Relations in Visual Reasoning. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 22, 1 (Jan 1996), 231. doi:10.1037//0278-7393.22.1.231

[34] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for Datasets. *Communications of the ACM* 64, 12 (Nov 2021), 86–92. doi:10.1145/3458723

[35] Katy Ilonka Gero, Vivian Liu, and Lydia Chilton. 2022. Sparks: Inspiration for Science Writing using Language Models. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference*. Association for Computing Machinery, New York, NY, USA, 1002–1019. doi:10.1145/3532106.3533533

[36] Antoon Goderis, Peter Li, and Carole Goble. 2006. Workflow Discovery: the Problem, a Case Study from E-Science and a Graph-Based Solution. In *2006 IEEE International Conference on Web Services*. Institute of Electrical and Electronics Engineer, Piscataway, New Jersey, USA, 312–319. doi:10.1109/ICWS.2006.147

[37] Nick Goodson and Rongfei Lu. 2023. Intention and Context Elicitation with Large Language Models in the Legal Aid Intake Process. *arXiv preprint* 1 (2023), 11 pages. doi:10.48550/arXiv.2311.13281

[38] Google. 2024. Retrieved March 20, 2024 from https://www.google.com/

[39] Philip J. Guo, Sean Kandel, Joseph M. Hellerstein, and Jeffrey Heer. 2011. Proactive Wrangling: Mixed-Initiative End-User Programming of Data Transformation Scripts. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 65–74. doi:10.1145/2047196.2047205

[40] John E. Harding and Paul Shepherd. 2017. Meta-Parametric Design. *Design Studies* 52 (Sep 2017), 73–95. doi:10.1016/j.destud.2016.09.005

[41] Jessica He, Stephanie Houde, Gabriel E. Gonzalez, Darío Andrés Silva Moran, Steven I. Ross, Michael Muller, and Justin D. Weisz. 2024. AI and the Future of Collaborative Work: Group Ideation with an LLM in a Virtual Canvas. In *Proceedings of the 3rd Annual Meeting of the Symposium on Human-Computer Interaction for Work*. Association for Computing Machinery, New York, NY, USA, Article 9, 14 pages. doi:10.1145/3663384.3663398

[42] Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems* 212 (Jan 2021), 106622. doi:10.1016/j.knosys.2020.106622

[43] Jeffrey Heer. 2019. Agency Plus Automation: Designing Artificial Intelligence into Interactive Systems. *Proceedings of the National Academy of Sciences* 116, 6 (2019), 1844–1850. doi:10.1073/pnas.1807184115

[44] Dulaji Hidellaarachchi, John Grundy, Rashina Hoda, and Ingo Mueller. 2023. The Influence of Human Aspects on Requirements Engineering-related Activities: Software Practitioners' Perspective. *ACM Transactions Software Engineering Methodology* 32, 5, Article 108 (Jul 2023), 37 pages. doi:10.1145/3546943

[45] Jordan Hollander. 2024. AI in the Hospitality Industry: Here's What You Need to Know. Retrieved November 20, 2024 from https://hoteltechreport.com/news/ai-in-hospitality

[46] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé, Miro Dudik, and Hanna Wallach. 2019. Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–16. doi:10.1145/3290605.3300830

[47] Jihyeong Hong, Yokyung Lee, Dae Hyun Kim, DaEun Choi, Yeo-Jin Yoon, Gyucheol Lee, Zucheul Lee, and Juho Kim. 2024. A Context-Aware Onboarding Agent for Metaverse Powered by Large Language Models. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference*. Association for Computing Machinery, New York, NY, USA, 1857–1874. doi:10.1145/3643834.3661579

[48] Ben Hutchinson, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell. 2021. Towards Accountability for Machine Learning Datasets: Practices from Software Engineering and Infrastructure. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, New York, NY, USA, 560–575. doi:10.1145/3442188.3445918

[49] K. Höök. 2000. Steps to Take Before Intelligent User Interfaces Become Real. *Interacting with Computers* 12, 4 (Feb 2000), 409–426. doi:10.1016/S0953-5438(99)00006-5

[50] IBM. 2024. IBM Engineering Requirements Management. https://www.ibm.com/products/requirements-management. Retrieved December 8, 2024.

[51] Asana Inc. 2024. Requirements Management 101: Your Step-by-Step Guide. https://asana.com/resources/requirements-management. Retrieved December 8, 2024.

[52] National Cancer Institute. 2024. Artificial Intelligence (AI) and Cancer. Retrieved November 20, 2024 from https://www.cancer.gov/research/infrastructure/artificial-intelligence

[53] Pankaj Jalote. 2012. *An Integrated Approach to Software Engineering*. Springer Science & Business Media, New York, NY, USA. doi:10.1007/978-1-4684-9312-2

[54] Yugyeong Jang and Kyung Hoon Hyun. 2024. Advancing 3D CAD with Workflow Graph-Driven Bayesian Command Inferences. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 36, 6 pages. doi:10.1145/3613905.3650895

[55] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems* 33, 2 (Feb 2022), 494–514. doi:10.1109/TNNLS.2021.3070843

[56] Hyoungwook Jin, Seonghee Lee, Hyungyu Shin, and Juho Kim. 2024. Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 28 pages. doi:10.1145/3613904.3642349

[57] Wei Jin and Rohini K. Srihari. 2007. Graph-Based Text Representation and Knowledge Discovery. In *Proceedings of the 2007 ACM Symposium on Applied Computing*. Association for Computing Machinery, New York, NY, USA, 807–811. doi:10.1145/1244002.1244182

[58] Jeff Johnson. 2020. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Elsevier, Amsterdam, Netherlands. doi:10.1016/C2012-0-07128-1

[59] Emma Kallina and Jatinder Singh. 2024. Stakeholder Involvement for Responsible AI Development: A Process Framework. In *Proceedings of the 4th ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*. Association for Computing Machinery, New York, NY, USA, Article 1, 14 pages. doi:10.1145/3689904.3694698

[60] Davinder Kaur, Suleyman Uslu, Kaley J. Rittichier, and Arjan Durresi. 2022. Trustworthy Artificial Intelligence: A Review. *Comput. Surveys* 55, 2, Article 39 (Jan. 2022), 38 pages. doi:10.1145/3491209

[61] Fahim Muhammad Khan, Javed Ali Khan, Muhammad Assam, Ahmed S. Almasoud, Abdelzahir Abdelmaboud, and Manar Ahmed Mohammed Hamza. 2022. A Comparative Systematic Analysis of Stakeholder's Identification Methods in Requirements Elicitation. *Institute of Electrical and Electronics Engineer Access* 10 (Feb 2022), 30982–31011. doi:10.1109/ACCESS.2022.3152073

[62] Dae Hyun Kim, Enamul Hoque, and Maneesh Agrawala. 2020. Answering Questions about Charts and Generating Visual Explanations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3313831.3376467

[63] Dae Hyun Kim, Enamul Hoque, Juho Kim, and Maneesh Agrawala. 2018. Facilitating Document Reading by Linking Text and Tables. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 423–434. doi:10.1145/3242587.3242617

[64] Dae Hyun Kim, Yoonsu Kim, Hyungyu Shin, Jinho Son, and Juho Kim. 2023. Towards Understanding the Challenges and Remedies in AI Application Development Planning. In *Korea Computer Congress*. Korean Institute of Information Scientists and Engineers, Seoul, South Korea, 1421–1423.

[65] Dae Hyun Kim, Hyungyu Shin, Shakhnozakhon Yadgarova, Jinho Son, Hariharan Subramonyam, and Juho Kim. 2024. AINeedsPlanner: A Workbook to Support Effective Collaboration Between AI Experts and Clients. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference*. Association for Computing Machinery, New York, NY, USA, 728–742. doi:10.1145/3643834.3661577

[66] Jeongyeon Kim, Daeun Choi, Nicole Lee, Matt Beane, and Juho Kim. 2023. Surch: Enabling Structural Search and Comparison for Surgical Videos. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 801, 17 pages. doi:10.1145/3544548.3580772

[67] Agnes Mercedes Kloft, Robin Welsch, Thomas Kosch, and Steeven Villa. 2024. "AI enhances our performance, I have no doubt this one will do the same": The Placebo Effect is Robust to Negative Descriptions of AI. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 24 pages. doi:10.1145/3613904.3642633

[68] Nicholas Kong, Marti A. Hearst, and Maneesh Agrawala. 2014. Extracting References Between Text and Charts via Crowdsourcing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 31–40. doi:10.1145/2556288.2557241

[69] Madhava Krishna, Bhagesh Gaur, Arsh Verma, and Pankaj Jalote. 2024. Using LLMs in Software Requirements Specifications: An Empirical Evaluation. *arXiv preprint* 1 (2024), 9 pages. doi:10.1109/RE59067.2024.00056

[70] Heng-Yu Ku, Christi A. Harter, Pei-Lin Liu, Ling Thompson, and Yi-Chia Cheng. 2007. The Effects of Individually Personalized Computer-Based Instructional Program on Solving Mathematics Problems. *Computers in Human Behavior* 23, 3 (May 2007), 1195–1210. doi:10.1016/j.chb.2004.11.017

[71] Yoonjoo Lee, John Joon Young Chung, Tae Soo Kim, Jean Y Song, and Juho Kim. 2022. Promptiverse: Scalable Generation of Scaffolding Prompts Through Human-AI Hybrid Knowledge Graph Annotation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 96, 18 pages. doi:10.1145/3491102.3502087

[72] Yoonjoo Lee, Kihoon Son, Tae Soo Kim, Jisu Kim, John Joon Young Chung, Eytan Adar, and Juho Kim. 2024. One vs. Many: Comprehending Accurate Information from Multiple Erroneous and Inconsistent AI Generations. In *Proceedings of the*

2024 ACM Conference on Fairness, Accountability, and Transparency. Association for Computing Machinery, New York, NY, USA, 2518–2531. doi:10.1145/3630106.3662681

[73] Alexander Lex, Christian Partl, Denis Kalkofen, Marc Streit, Samuel Gratzl, Anne Mai Wassermann, Dieter Schmalstieg, and Hanspeter Pfister. 2013. Entourage: Visualizing Relationships between Biological Pathways using Contextual Subsets. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec 2013), 2536–2545. doi:10.1109/TVCG.2013.154

[74] Belinda Z Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. 2023. Eliciting Human Preferences with Language Models. *arXiv preprint* 1 (2023), 26 pages. doi:10.48550/arXiv.2310.11589

[75] Ching Liu, Juho Kim, and Hao-Chuan Wang. 2018. ConceptScape: Collaborative Concept Mapping for Video Learning. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3173574.3173961

[76] Qiao Liu, Liuyi Jiang, Minghao Han, Yao Liu, and Zhiguang Qin. 2016. Hierarchical Random Walk Inference in Knowledge Graphs. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval.* Association for Computing Machinery, New York, NY, USA, 445–454. doi:10.1145/2911451.2911509

[77] Ryan Louie, Ananjan Nandi, William Fang, Cheng Chang, Emma Brunskill, and Diyi Yang. 2024. Roleplay-doh: Enabling Domain-Experts to Create LLM-simulated Patients via Eliciting and Adhering to Principles. *arXiv preprint* 1 (2024), 34 pages. doi:10.48550/arXiv.2407.00870

[78] Josh Lovejoy. 2019. Human-centered AI Cheat-Sheet. https://uxdesign.cc/human-centered-ai-cheat-sheet-1da130ba1bab. Retrieved December 8, 2024.

[79] Lucy Ellen Lwakatare, Aiswarya Raj, Jan Bosch, Helena Holmström Olsson, and Ivica Crnkovic. 2019. A Taxonomy of Software Engineering Challenges for Machine Learning Systems: An Empirical Investigation. In *Agile Processes in Software Engineering and Extreme Programming.* Springer International Publishing, Cham, Switzerland, 227–243.

[80] Silverio Martínez-Fernández, Justus Bogner, Xavier Franch, Marc Oriol, Julien Siebert, Adam Trendowicz, Anna Maria Vollmer, and Stefan Wagner. 2022. Software Engineering for AI-Based Systems: A Survey. *ACM Transactions Software Engineering Methodology* 31, 2, Article 37e (April 2022), 59 pages. doi:10.1145/3487043

[81] John McCormick. 2020. AI-Enabled Cheetos Offer Promise of the Perfect Puff). Retrieved November 20, 2024 from https://www.wsj.com/articles/ai-enabled-cheetos-offer-promise-of-the-perfect-puff-11608158547

[82] Colin D. McLaren and Mark W. Bruner. 2022. Citation Network Analysis. *International Review of Sport and Exercise Psychology* 15, 1 (Jan 2022), 179–198. doi:10.1080/1750984X.2021.1989705

[83] Microsoft. 2022. Microsoft Responsible AI Standard, v2 General Requirements. https://blogs.microsoft.com/wp-content/uploads/prod/sites/5/2022/06/Microsoft-Responsible-AI-Standard-v2-General-Requirements-3.pdf. Retrieved December 8, 2024.

[84] Piotr Mirowski, Kory W. Mathewson, Jaylen Pittman, and Richard Evans. 2023. Co-Writing Screenplays and Theatre Scripts with Language Models: Evaluation by Industry Professionals. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, Article 355, 34 pages. doi:10.1145/3544548.3581225

[85] Steven Moore, Q. Vera Liao, and Hariharan Subramonyam. 2023. fAIlureNotes: Supporting Designers in Understanding the Limits of AI Models for Computer Vision Tasks. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, 19 pages. doi:10.1145/3544548.3581242

[86] Per Rådberg Nagbøl, Oliver Müller, and Oliver Krancher. 2021. Designing a Risk Assessment Tool for Artificial Intelligence Systems. In *The Next Wave of Sociotechnical Design.* Springer International Publishing, Cham, Switzerland, 328–339. doi:10.1007/978-3-030-82405-1_32

[87] Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. 2019. Graph-based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network. In *IEEE/WIC/ACM International Conference on Web Intelligence.* Association for Computing Machinery, New York, NY, USA, 156–163. doi:10.1145/3350546.3352513

[88] Angela M O'donnell, Donald F Dansereau, and Richard H Hall. 2002. Knowledge Maps as Scaffolds for Cognitive Processing. *Educational Psychology Review* 14 (Mar 2002), 71–86. doi:10.1023/A:1013132527007

[89] OpenAI. 2024. Embeddings. Retrieved March 20, 2024 from https://platform.openai.com/docs/guides/embeddings

[90] OpenAI. 2024. New Embedding Models and API Updates. Retrieved March 20, 2024 from https://openai.com/blog/new-embedding-models-and-api-updates

[91] Srishti Palani, Yingyi Zhou, Sheldon Zhu, and Steven P. Dow. 2022. InterWeave: Presenting Search Suggestions in Context Scaffolds Information Search and Synthesis. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology.* Association for Computing Machinery, New York, NY, USA, Article 93, 16 pages. doi:10.1145/3526113.3545696

[92] Jeongeon Park, Bryan Min, Xiaojuan Ma, and Juho Kim. 2023. Choicemates: Supporting Unfamiliar Online Decision-Making with Multi-Agent Conversational Interactions. *arXiv preprint* 1 (2023), 19 pages. doi:10.48550/arXiv.2310.01331

[93] Soya Park, April Yi Wang, Ban Kawas, Q. Vera Liao, David Piorkowski, and Marina Danilevsky. 2021. Facilitating Knowledge Sharing from Domain Experts to Data Scientists for Building NLP Models. In *Proceedings of the 26th International Conference on Intelligent User Interfaces.* Association for Computing Machinery, New York, NY, USA, 585–596. doi:10.1145/3397481.3450637

[94] Savvas Petridis, Benjamin D Wedin, James Wexler, Mahima Pushkarna, Aaron Donsbach, Nitesh Goyal, Carrie J Cai, and Michael Terry. 2024. Constitution-Maker: Interactively Critiquing Large Language Models by Converting Feedback into Principles. In *Proceedings of the 29th International Conference on Intelligent User Interfaces.* Association for Computing Machinery, New York, NY, USA, 853–868. doi:10.1145/3640543.3645144

[95] David Piorkowski, Soya Park, April Yi Wang, Dakuo Wang, Michael Muller, and Felix Portnoy. 2021. How AI Developers Overcome Communication Challenges in a Multidisciplinary Team: A Case Study. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1, Article 131 (Apr 2021), 25 pages. doi:10.1145/3449205

[96] David Piorkowski, Soya Park, April Yi Wang, Dakuo Wang, Michael Muller, and Felix Portnoy. 2021. How AI Developers Overcome Communication Challenges in a Multidisciplinary Team: A Case Study. *Proceedings of the ACM on Human-Computer Interacteration* 5, CSCW1, Article 131 (April 2021), 25 pages. doi:10.1145/3449205

[97] Ritika Poddar, Rashmi Sinha, Mor Naaman, and Maurice Jakesch. 2023. AI Writing Assistants Influence Topic Choice in Self-Presentation. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, Article 29, 6 pages. doi:10.1145/3544549.3585893

[98] Maria Priestley, Fionntán O'donnell, and Elena Simperl. 2023. A Survey of Data Quality Requirements That Matter in ML Development Pipelines. *Journal of Data and Information Quality* 15, 2 (June 2023), 39 pages. doi:10.1145/3592616

[99] Filippo Radicchi, Santo Fortunato, and Alessandro Vespignani. 2012. Citation Networks. *Models of science dynamics: Encounters between Complexity Theory and Information Sciences* 1 (Jan 2012), 233–257. doi:10.1007/978-3-642-23068-4_7

[100] Google People + AI Research. 2019. Google People + AI Guidebook. Retrieved January 20, 2024 from https://pair.withgoogle.com/guidebook/

[101] P. A. Kulkarni S. S. Sonawane. 2014. Graph based Representation and Analysis of Text Document: A Survey of Techniques . *International Journal of Computer Applications* 96, 19 (Jun 2014), 1–8. doi:10.5120/16899-6972

[102] Raafat George Saadé, Danielle Morin, and Jennifer D.E. Thomas. 2012. Critical Thinking in E-learning Environments. *Computers in Human Behavior* 28, 5 (2012), 1608–1617. doi:10.1016/j.chb.2012.03.025

[103] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. 2021. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, 15 pages. doi:10.1145/3411764.3445518

[104] Google Scholar. 2024. Retrieved March 20, 2024 from https://scholar.google.com/

[105] Semantic Scholar. 2024. Retrieved March 20, 2024 from https://www.semanticscholar.org/

[106] Karima Sedki and Louis Bonneau de Beaufort. 2012. Cognitive Maps for Knowledge Represenation and Reasoning. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, Vol. 1. Institute of Electrical and Electronics Engineer, Piscataway, New Jersey, USA, 1035–1040. doi:10.1109/ICTAI.2012.175

[107] Orit Shaer, Angelora Cooper, Osnat Mokryn, Andrew L Kun, and Hagit Ben Shoshan. 2024. AI-Augmented Brainwriting: Investigating the Use of LLMs in Group Ideation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, Article 1050, 17 pages. doi:10.1145/3613904.3642414

[108] Omar Shaikh, Kristina Gligoric, Ashna Khetan, Matthias Gerstgrasser, Diyi Yang, and Dan Jurafsky. 2024. Grounding Gaps in Language Model Generations. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Association for Computational Linguistics, Mexico City, Mexico, 6279–6296. doi:10.18653/v1/2024.naacl-long.348

[109] Zheyuan Ryan Shi, Claire Wang, and Fei Fang. 2020. Artificial Intelligence for Social Good: A Survey. *arXiv preprint* 1 (2020), 78 pages. doi:10.48550/arXiv.2001.01818

[110] Janice C. Sipior. 2020. Considerations for Development and Use of AI in Response to COVID-19. *International Journal of Information Management* 55 (Dec 2020), 102170. doi:10.1016/j.ijinfomgt.2020.102170

[111] Sicheng Song, Juntong Chen, Chenhui Li, and Changbo Wang. 2023. GVQA: Learning to Answer Questions about Graphs with Visualizations via Knowledge Base. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, Article 464, 16 pages. doi:10.1145/3544548.3581067

[112] Statistica. 2023. Artificial Intelligence - Worldwide. Retrieved January 20, 2024 from https://www.statista.com/outlook/tmo/artificial-intelligence/worldwide

[113] Monika Steidl, Michael Felderer, and Rudolf Ramler. 2023. The Pipeline for the Continuous Development of Artificial Intelligence Models—Current State of Research and Practice. *Journal of Systems and Software* 199 (May 2023), 111615. doi:10.1016/j.jss.2023.111615

[114] Mark Steyvers and Joshua B. Tenenbaum. 2005. The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. *Cognitive Science* 29, 1 (Jan 2005), 41–78. doi:10.1207/s15516709cog2901_3

[115] Debbie Stone, Caroline Jarrett, Mark Woodroffe, and Shailey Minocha. 2005. *User Interface Design and Evaluation.* Elsevier, Amsterdam, Netherlands.

[116] Hariharan Subramonyam, Jane Im, Colleen Seifert, and Eytan Adar. 2022. Solving Separation-of-Concerns Problems in Collaborative Design of Human-AI Systems through Leaky Abstractions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, Article 481, 21 pages. doi:10.1145/3491102.3517537

[117] Hariharan Subramonyam, Colleen Seifert, and Eytan Adar. 2021. ProtoAI: Model-Informed Prototyping for AI-Powered Interfaces. In *Proceedings of the 26th International Conference on Intelligent User Interfaces.* Association for Computing Machinery, New York, NY, USA, 48–58. doi:10.1145/3397481.3450640

[118] Hariharan Subramonyam, Colleen Seifert, and MI Eytan Adar. 2021. How Can Human-Centered Design Shape Data-Centric AI. In *Proceedings of NeurIPS Data-Centric AI Workshop.* Curran Associates, New York, NY, USA, 3 pages.

[119] Hariharan Subramonyam, Colleen Seifert, Priti Shah, and Eytan Adar. 2020. texSketch: Active Diagramming through Pen-and-Ink Annotations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3313831.3376155

[120] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. 2023. Sensecape: Enabling Multilevel Exploration and Sensemaking with Large Language Models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology.* Association for Computing Machinery, New York, NY, USA, 18 pages. doi:10.1145/3586183.3606756

[121] Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. 2022. Learning and Evaluating Graph Neural Network Explanations based on Counterfactual and Factual Reasoning. In *Proceedings of the ACM Web Conference 2022.* Association for Computing Machinery, New York, NY, USA, 1018–1027. doi:10.1145/3485447.3511948

[122] Mei Tan, Hansol Lee, Dakuo Wang, and Hari Subramonyam. 2024. Is a Seat at the Table Enough? Engaging Teachers and Students in Dataset Specification for ML in Education. *Proceedings of the ACM on Human-Computer Interacteration* 8, CSCW1, Article 81 (April 2024), 32 pages. doi:10.1145/3637358

[123] Chien-Lin Tang, Jingxian Liao, Hao-Chuan Wang, Ching-Ying Sung, and Wen-Chieh Lin. 2021. ConceptGuide: Supporting Online Video Learning with Concept Map-based Recommendation of Learning Path. In *Proceedings of the Web Conference 2021.* Association for Computing Machinery, New York, NY, USA, 2757–2768. doi:10.1145/3442381.3449808

[124] TheFinance.sg. 2024. How AI Chatbots are Revolutionizing Customer Service Across Industries). Retrieved November 20, 2024 from https://thefinance.sg/2024/11/17/how-ai-chatbots-are-revolutionizing-customer-service-across-industries/

[125] Jakob Tholander and Martin Jonsson. 2023. Design Ideation with AI - Sketching, Thinking and Talking with Generative Machine Learning Models. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference.* Association for Computing Machinery, New York, NY, USA, 1930–1940. doi:10.1145/3563657.3596014

[126] W. M. P. van der Aalst, A. Hirnschall, and H. M. W. Verbeek. 2002. An Alternative Way to Analyze Workflow Graphs. In *Advanced Information Systems Engineering.* Springer Berlin Heidelberg, Berlin, Heidelberg, 535–552. doi:10.1007/3-540-47961-9_37

[127] Candace Walkington, Anthony Petrosino, and Milan Sherman. 2013. Supporting Algebraic Reasoning through Personalized Story Scenarios: How Situational Understanding Mediates Performance. *Mathematical Thinking and Learning* 15, 2 (Apr 2013), 89–120. doi:10.1080/10986065.2013.770717

[128] Zhiyuan Wan, Xin Xia, David Lo, and Gail C Murphy. 2019. How does Machine Learning Change Software Development Practices? *IEEE Transactions on Software Engineering* 47, 9 (Sept. 2019), 1857–1871. doi:10.1109/TSE.2019.2937083

[129] Dakuo Wang, Elizabeth Churchill, Pattie Maes, Xiangmin Fan, Ben Shneiderman, Yuanchun Shi, and Qianying Wang. 2020. From Human-Human Collaboration to Human-AI Collaboration: Designing AI Systems That Can Work Together with People. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, 1–6. doi:10.1145/3334480.3381069

[130] Lina Weichbrodt. 2021. Domain-Specific Machine Learning Monitoring. Retrieved March 20, 2024 from https://medium.com/mlops-community/domain-specific-machine-learning-monitoring-88bc0dd8a212

[131] Krzysztof Wnuk. 2017. Involving Relevant Stakeholders into the Decision Process about Software Components. In *2017 IEEE International Conference on Software Architecture Workshops.* Institute of Electrical and Electronics Engineer, Piscataway, New Jersey, USA, 129–132. doi:10.1109/ICSAW.2017.68

[132] Suraya Ya'acob, Nazlena Mohamad Ali, and Norshita Mat Nayan. 2013. Understanding Big Picture and Its Challenges: Experts and Decision Makers Perspectives. In *Advances in Visual Informatics.* Springer International Publishing, Cham, Switzerland, 311–322. doi:10.1007/978-3-319-02958-0_29

[133] Youfu Yan, Yu Hou, Yongkang Xiao, Rui Zhang, and Qianwen Wang. 2025. KNowNEt:Guided Health Information Seeking from LLMs via Knowledge Graph Integration. *IEEE Transactions on Visualization and Computer Graphics* 31, 1 (Jan 2025), 547–557. doi:10.1109/TVCG.2024.3456364

[134] Yang Yang, Jian Shen, Yanru Qu, Yunfei Liu, Kerong Wang, Yaoming Zhu, Weinan Zhang, and Yong Yu. 2021. GIKT: A Graph-Based Interaction Model for Knowledge Tracing. In *Machine Learning and Knowledge Discovery in Databases.* Springer International Publishing, Cham, Switzerland, 299–315. doi:10.1007/978-3-030-67658-2_18

[135] Nur Yildirim, Mahima Pushkarna, Nitesh Goyal, Martin Wattenberg, and Fernanda Viégas. 2023. Investigating How Practitioners Use Human-AI Guidelines: A Case Study on the People + AI Guidebook. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, 13 pages. doi:10.1145/3544548.3580900

[136] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: Story Writing With Large Language Models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces.* Association for Computing Machinery, New York, NY, USA, 841–852. doi:10.1145/3490099.3511105

[137] Angie Zhang, Alexander Boltz, Jonathan Lynn, Chun-Wei Wang, and Min Kyung Lee. 2023. Stakeholder-Centered AI Design: Co-Designing Worker Tools with Gig Workers through Data Probes. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, Article 859, 19 pages. doi:10.1145/3544548.3581354

[138] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. Greaselm: Graph Reasoning Enhanced Language Models for Question Answering. *arXiv preprint* 1 (2022), 16 pages. doi:10.48550/arXiv.2201.08860

[139] Xiaoyu Zhang, Jianping Li, Po-Wei Chi, Senthil Chandrasegaran, and Kwan-Liu Ma. 2023. ConceptEVA: Concept-Based Interactive Exploration and Customization of Document Summaries. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, Article 204, 16 pages. doi:10.1145/3544548.3581260

[140] Zheng Zhang, Jie Gao, Ranjodh Singh Dhaliwal, and Toby Jia-Jun Li. 2023. VISAR: A Human-AI Argumentative Writing Assistant with Visual Programming and Rapid Draft Prototyping. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology.* Association for Computing Machinery, New York, NY, USA, Article 5, 30 pages. doi:10.1145/3586183.3606800

[141] Dangzhi Zhao and Andreas Strotmann. 2015. *Analysis and Visualization of Citation Networks.* Morgan and Claypool Publishers, San Rafael, California, USA. doi:10.2200/S00624ED1V01Y201501ICR039

[142] Faguo Zhou, Fan Zhang, and Bingru Yang. 2010. Graph-Based Text Representation Model and its Realization. In *Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering(NLPKE-2010).* Institute of Electrical and Electronics Engineer, Piscataway, New Jersey, USA, 1–8. doi:10.1109/NLPKE.2010.5587861

[143] Zoom. 2024. Retrieved March 20, 2024 from https://zoom.us/

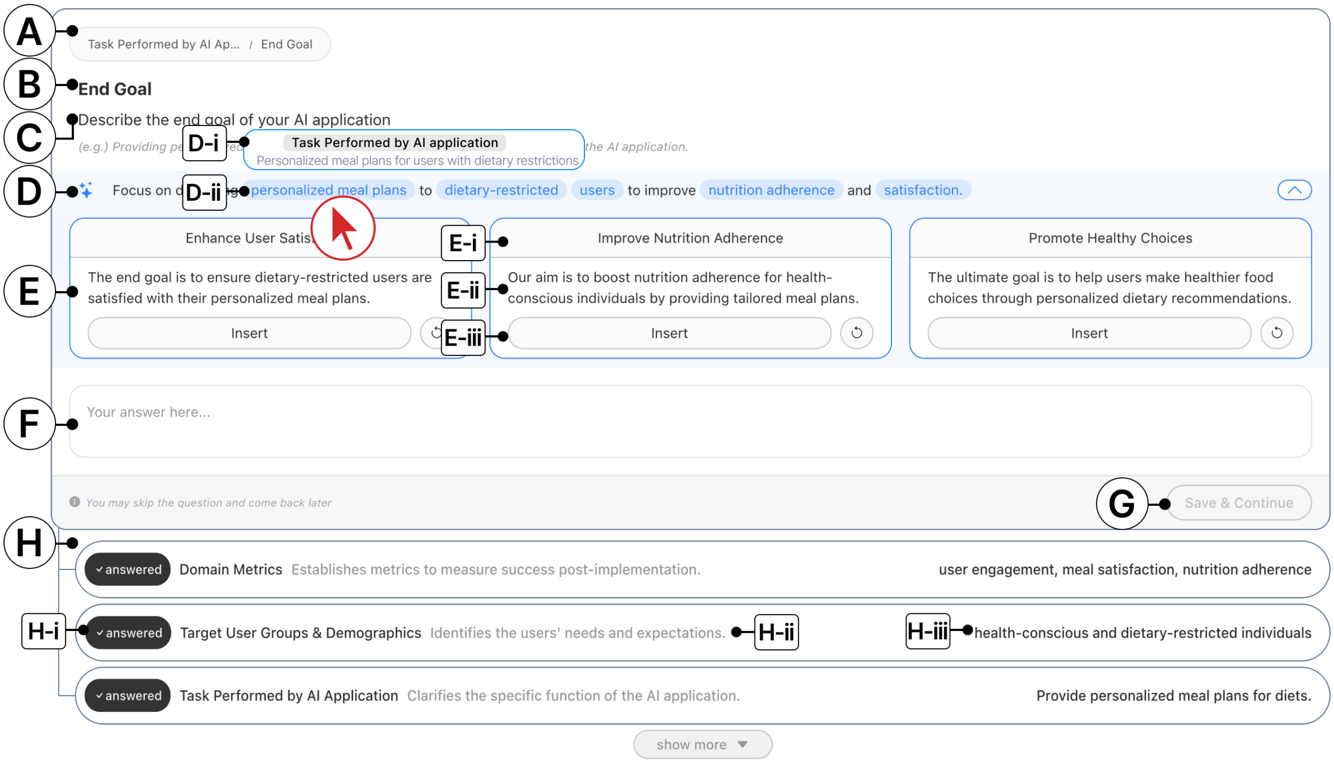
A Detailed Descriptions of Interface Panels

This section includes detailed descriptions about the four interface components: (1) Main Panel, (2) Table of Contents, (3) Overview Panel, and (4) AI Chatbot.

A.1 Main Panel

The *Main Panel* (Figure 12; Figure 6C), which is at the center of the interface, displays the title (Component B) and the question (Component C) the client should answer. The client would type in their answer into the answer field given in Component F and press the "Save & Continue" button (Component G) to proceed to the next question. Upon clicking on this button, the system fetches the next question from the Planning Information Graph to be displayed to the client.

To aid the client as they answer the question, the interface displays tips (Component D) and suggestions (Component E) created by the Guidance Generator (Section 4.3) based on the information available in the children nodes in the Planning Information Graph.



End Goal

Describe the end goal of your AI application

(e.g.) Providing pe… the AI application.

Task Performed by AI application
Personalized meal plans for users with dietary restrictions

Focus on … personalized meal plans to dietary-restricted users to improve nutrition adherence and satisfaction.

Enhance User Satis…
The end goal is to ensure dietary-restricted users are satisfied with their personalized meal plans.

Improve Nutrition Adherence
Our aim is to boost nutrition adherence for health-conscious individuals by providing tailored meal plans.

Promote Healthy Choices
The ultimate goal is to help users make healthier food choices through personalized dietary recommendations.

Your answer here...

You may skip the question and come back later

Domain Metrics Establishes metrics to measure success post-implementation. user engagement, meal satisfaction, nutrition adherence

Target User Groups & Demographics Identifies the users' needs and expectations. health-conscious and dietary-restricted individuals

Task Performed by AI Application Clarifies the specific function of the AI application. Provide personalized meal plans for diets.

Figure 12: The Main Panel (Figure 6 Panel A). In addition to the basic question-answering interface (Components B, C, F, G), it includes features helping the client answer the given question (Components D, E) and navigation features (Components A, H).

The system highlights phrases of the provided tips that are related to previously answered questions (Component D-ii) so that the client can contextualize the current question in the planning flow. Upon hovering over the highlighted phrase, the client can get a summary of the question and the answer (Component D-i) that the phrase is drawing information from. We initially hide the suggestions to promote client's independent thinking about the question before being influenced by the suggestions. Upon clicking on the downward arrow button (⌄), the system displays two to three suggestions, each of which is displayed as a theme (Component E-i) and the suggested answer (Component E-ii). The client can click on the *insert* button (Component E-iii) add the suggested answer to the answer box in Component F, which they can further edit to match their intent. If the client wishes to view more suggestions, they can click on the *refresh* button (↻).

Other than scrolling up and down to revisit previously visited questions, the Main Panel also offers additional navigation features that leverage the graph structure. The breadcrumb navigation bar (Component A) informs the client about which graph traversal path they have taken to arrive at the current question, which are questions that answering the current question can help with. The clients can freely navigate back to the questions mentioned in the breadcrumb navigation bar by clicking on them, which would be equivalent to traversing to parent nodes in the Planning Information Graph. The other navigation feature included in the system is the *graph-based navigation* (Component H). The graph-based navigation displays the children node questions that can help answer the current question (Component H-i) as well as how they are related to the current question (Component H-ii). If the question has already been answered, a summary of the provided answer is also provided to to help answer the current question (Component H-iii). The graph-based navigation also includes tags showing the status of the shown question: answered vs unanswered, AI-generated to provide further support, and/or prerequisite for answering the current question. When clients have difficulty answering the current question, they can click on one of the questions in this display to move to another question that may provide insights and information that can help them tackle the current question later.

A.2 Table of Contents

Table of Contents (Figure 6A), which is on the left side of the interface, helps the client navigate through the questions as well as see the answered state of questions and the relationships between the questions. The Table of Contents is a nested list of sections and questions inside them, in the order that the client has visited them. To show the completion state, the panel shows a circled check mark (✓⃝) for completed sections (e.g., 'Dataset' in Figure 6A) and check mark (✓) for answered questions (e.g., 'Task Performed by AI Application' in Figure 6A). Furthermore, to give a sense of relationships between the questions, it highlights the current questions in blue (e.g., 'Task Performed by AI Application' in Figure 6A) and related questions (i.e., neighbors in the Planning Information Graph) in gray (e.g., 'End Goal' in Figure 6A). As a final note, the Table of Contents includes a progress bar showing the overall progress at the top.

A.3 Overview Panel

To keep the client aware of the big picture while answering each of the questions, PlanTogether includes an Overview Panel (Figure 6B; examples in Figure 13) at the top of the interface. The panel provides a tabular summary of the responses of the selected section as well as the progress on each of the sections underneath.

Initially, the tabular summary is empty but gradually fills as clients answer more questions in the Main Panel, offering a sense of progress and concretization of the pre-collaboration plan. Once a question is answered, its corresponding portion in the tabular summary is filled in. Inspired by the practice of highlighting links between visualizations and text to reduce the split-attention problem [63, 68], PlanTogether highlights relevant portions of the Overview Panel based on the current question the client is answering. For instance, in Figure 13b, as the client works on questions related to dataset size, the 'Size' portion of the Overview Panel is highlighted.

The sectional progress bar underneath displays the percentage of the questions answered in each of the sections. The client views the tabular summary of other sections by clicking on the section names on the sectional progress bar.

A.4 AI Chatbot

The system includes an AI chatbot (Figure 6D; Figure 2c) to reduce cognitive load from switching to use LLM-based chatbot services and the prompting efforts needed to capture the contents of the current plan. Upon clicking on the 'AI Chatbot' button (Figure 6D), the AI chatbot displays a chat interface (Figure 2c) with a chat history and an input field.

B Planning Information Graph Specification

This section includes detailed specification of the Planning Information Graph Components: nodes and edges.

B.1 Nodes

A *node* of the Planning Information Graph, which represents each piece of information that the client needs to determine the pre-collaboration plan, includes (Figure 7A):

- *Title*: A short summary of the contents of each node.
- *Question*: The question asked to the client at each node.
- *Answer*: The answer that the client provides to the question. The node answers remain empty until filled in by the client.
- *Type*: A classification of the nodes based on the role of the information they cover: *core node* or *supplemental node*.
  - *Core*: Information that is part of the information needs of the AI experts from their clients according to Kim et al. [65] (e.g., nodes in Figure 7).
  - *Supplemental*: Information that is not part of the information needs of the AI expert, but helpful in eliciting and guiding clients as they provide information for other nodes in the graph; supplemental nodes are dynamically generated on a needs-basis by the system.
- *Embedding Vector*: The embedding vectors generated by using the `text-embedding-3-small` model [90] on the question. Our system uses cosine similarity [89] on these embedding vectors for efficient yet accurate comparison of the semantic information included in the nodes.
- *Answer Form*: The type of form (e.g., free-form, list) that PlanTogether would use to receive answers from the client. We include further details about the answer form in the supplemental material.

B.2 Edges

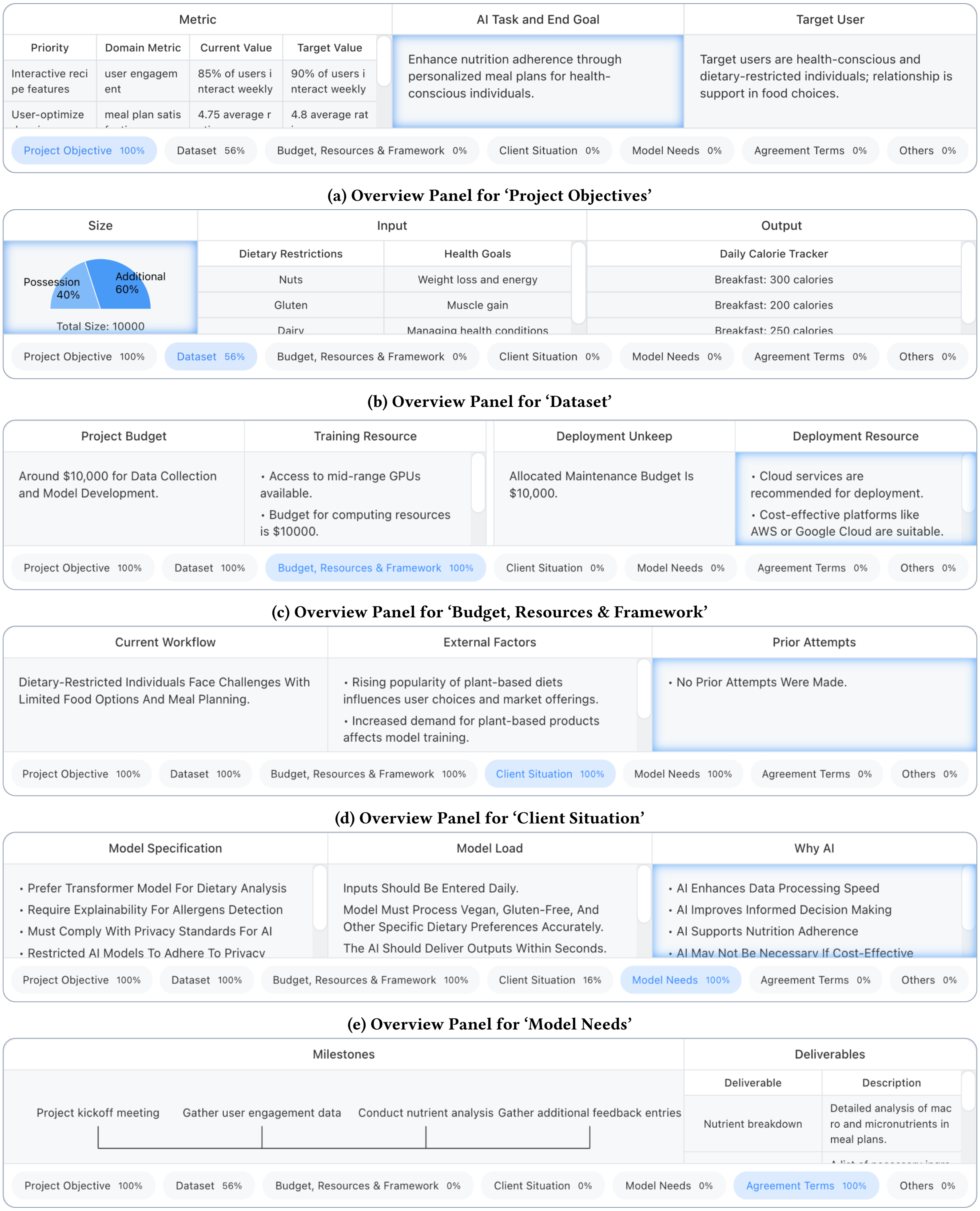
An *edge* of the Planning Information Graph, which represents the information dependency between two nodes (i.e., the parent node has information dependencies on the current node), includes (Figure 7B):

- *Parent node*: The source node of the edge.
- *Child node*: The destination node of the edge.
- *Type*: Type of the information dependency between the parent and the child nodes: *prerequisite* or *contextual*.
  - *Prerequisite*: 'A → B' indicates that answering the child node B is a prerequisite for questioning about the parent node A. For example, in Figure 7, we need to have asked about what the 'Domain Metrics' is before we can ask about its 'Measurement Method'.
  - *Contextual*: 'A → B' indicates that the answer to child node B can help contextualize information in the parent node A and hence help answer the parent node. For example, in Figure 7, knowing about the 'End Goal' can help determine what the 'Domain Metric' could be suitable for evaluating the AI application.
- *Weight*: The degree of information dependencies between the parent and child nodes. The edge weight is computed using three factors: semantic similarity of the parent and child nodes, how much the child node helps reason about the parent node, and how often the user chose to traverse to the specific child node from the parent. The Planning Information Graph Manager utilizes this information to decide on the traversal order and suggestion. The edge weights also determine the ordering of the questions in the graph-based navigation (Figure 12H).

The weight of an edge in the Planning Information Graph is computed as a weighted average of three factors: semantic similarity of the parent and children nodes, reasoning usage score, and user selection score:

$$w = \lambda_{\text{semantic}} w_{\text{semantic}} + \lambda_{\text{reasoning}} w_{\text{reasoning}} + \lambda_{\text{user}} w_{\text{user}},$$

where w_* represents each of the three factors and λ_* represents the relative weights of the three factors when computing the average. Each of the three factors take on a value between 0 and 1, with values closer to 0 indicating weaker information dependencies and values closer to 1 indicating stronger information dependencies. The system uses these edge weights to decide on traversal order and suggestions; the edge weights also help determine the ordering of the questions in the graph-based navigation (Figure 12H).



Metric | AI Task and End Goal | Target User

Priority	Domain Metric	Current Value	Target Value
Interactive reci pe features	user engagem ent	85% of users i nteract weekly	90% of users i nteract weekly
User-optimize	meal plan satis	4.75 average r	4.8 average rat

Enhance nutrition adherence through personalized meal plans for health-conscious individuals.

Target users are health-conscious and dietary-restricted individuals; relationship is support in food choices.

Project Objective 100% | Dataset 56% | Budget, Resources & Framework 0% | Client Situation 0% | Model Needs 0% | Agreement Terms 0% | Others 0%

(a) Overview Panel for 'Project Objectives'

Size | Input | Output

Total Size: 10000

Dietary Restrictions	Health Goals
Nuts	Weight loss and energy
Gluten	Muscle gain
Dairy	Managing health conditions

Daily Calorie Tracker
Breakfast: 300 calories
Breakfast: 200 calories
Breakfast: 250 calories

Project Objective 100% | Dataset 56% | Budget, Resources & Framework 0% | Client Situation 0% | Model Needs 0% | Agreement Terms 0% | Others 0%

(b) Overview Panel for 'Dataset'

Project Budget | Training Resource | Deployment Unkeep | Deployment Resource

Around $10,000 for Data Collection and Model Development.

- Access to mid-range GPUs available.
- Budget for computing resources is $10000.

Allocated Maintenance Budget Is $10,000.

- Cloud services are recommended for deployment.
- Cost-effective platforms like AWS or Google Cloud are suitable.

Project Objective 100% | Dataset 100% | Budget, Resources & Framework 100% | Client Situation 0% | Model Needs 0% | Agreement Terms 0% | Others 0%

(c) Overview Panel for 'Budget, Resources & Framework'

Current Workflow | External Factors | Prior Attempts

Dietary-Restricted Individuals Face Challenges With Limited Food Options And Meal Planning.

- Rising popularity of plant-based diets influences user choices and market offerings.
- Increased demand for plant-based products affects model training.

- No Prior Attempts Were Made.

Project Objective 100% | Dataset 100% | Budget, Resources & Framework 100% | Client Situation 100% | Model Needs 100% | Agreement Terms 0% | Others 0%

(d) Overview Panel for 'Client Situation'

Model Specification | Model Load | Why AI

- Prefer Transformer Model For Dietary Analysis
- Require Explainability For Allergens Detection
- Must Comply With Privacy Standards For AI
- Restricted AI Models To Adhere To Privacy

Inputs Should Be Entered Daily.

Model Must Process Vegan, Gluten-Free, And Other Specific Dietary Preferences Accurately.

The AI Should Deliver Outputs Within Seconds.

- AI Enhances Data Processing Speed
- AI Improves Informed Decision Making
- AI Supports Nutrition Adherence
- AI May Not Be Necessary If Cost-Effective

Project Objective 100% | Dataset 100% | Budget, Resources & Framework 100% | Client Situation 16% | Model Needs 100% | Agreement Terms 0% | Others 0%

(e) Overview Panel for 'Model Needs'

Milestones | Deliverables

Project kickoff meeting | Gather user engagement data | Conduct nutrient analysis | Gather additional feedback entries

Deliverable	Description
Nutrient breakdown	Detailed analysis of mac ro and micronutrients in meal plans.

Project Objective 100% | Dataset 56% | Budget, Resources & Framework 0% | Client Situation 0% | Model Needs 0% | Agreement Terms 100% | Others 0%

(f) Overview Panel for 'Agreement Terms'

Figure 13: Example views of the Overview Panels (Figure 6B) for each of the sections.

We empirically use $\lambda_{semantic} = 0.4$, $\lambda_{reasoning} = 0.3$, and $\lambda_{user} = 0.3$, in our implementation, but there may be more optimal weights. For the user study, we froze the edge weights based on the data we collected through our preliminary evaluation (Section 5) to provide a uniform experience.

Semantic Similarity. The semantic similarity of the parent and child nodes (w_{semantic}) represents how similar the contents the two nodes cover are. This score is computed by taking the cosine similarity of the embedding vectors of the two nodes and normalizing the value to be between 0 and 1.

Reasoning Usage Score. The reasoning usage score ($w_{\text{reasoning}}$) represents how often the system uses the answer from the child node when providing guidance for the parent node. This score is computed using the formula:

$$w_{\text{reasoning}} = \frac{b + r}{2b + t},$$

where b is a buffer variable to stabilize the score for the first clients using the system, r is the number of times the system used the answer from the child node for reasoning about the parent node, and t is the total number of times the system attempted to reason about the parent node with the answer from the child node. If the child node is never used for reasoning, the score will converge to 0; if the child node is always used for reasoning, the score will converge to 1.

Note that the buffer variable initiates the score at 0.5 and keeps the score close to 0.5 initially. Larger values of the buffer variable reduce the score variability for the first users but may require more users to converge.

User Selection Score. The user selection score (w_{user}) represents the propensity of the user to select a child node over other children nodes when given a choice through the supporting questions display (navigation feature in Figure 12H). This score is computed using the formula:

$$w_{\text{reasoning}} = \frac{b + \sum_{u \in U} s_u}{2b + |U|},$$

where b is a buffer variable playing the same role as in the reasoning usage score, U is the set of users who have reached the edge (i.e., shown as a traversal choice in the supporting questions display when skipping the current node), and s_{u} is the inverted user selection rank (i.e., if the child node is the n-th child node to be selected for traversal by the user u, the score is $1/n$). If the edge was never selected but the user traversed along other presented edges, the score is set to 0. The user selection score of an edge that users will never choose to traverse on, despite being given the choice, would converge to 0; the user selection score of an edge that users always choose to traverse on first would converge to 1.

C Graph Traversal Algorithm

This section provides detailed algorithm of the five-step traversal (Figure 8 based on depth-first search traversal algorithm and the downward/upward graph traversal mechanism (Section 4.2.1).

To aid graph traversal by checking the completion of each step and to avoid revisiting nodes due to cycles in the graph, we keep the states of the nodes as one of the following:

- *Unvisited* (U): the node has not been visited.
- *Answered* (A): the node has been visited and the client has provided an answer for the node.
- *Unanswered* (X): the node has been visited but the client was unable to provide an answer for the node.
- *Deferred* (D): the node has been visited but the client answer for the node has been deferred to traverse to prerequisite children nodes.

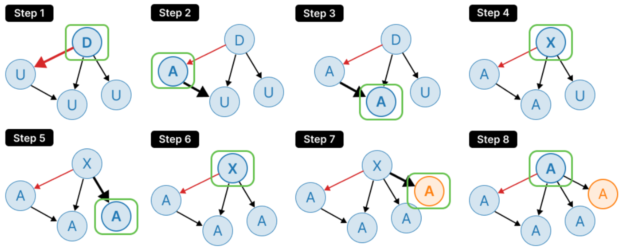


Figure 14: An example of traversal based on the five-step graph traversal algorithm. Green boxes indicate the current nodes of traversal and the letters on the nodes indicate the state of each node. Thick arrows indicate the direction of traversal at each step. The algorithm is based on the depth-first search algorithm, but it first visits the prerequisite nodes (Step 2) and then the core contextual nodes (Step 5). Afterward, it will generate and visit supplemental nodes (Step 7), presenting the current node in between (Steps 1, 4, 6, 8).

Figure 14 shows how this recursive algorithm would play out at a higher level with an example graph.

Step 1: Traverse to prerequisite children nodes. Before PlanTogether can inquire the client about a node, all prerequisite nodes must have been answered. Hence, the module marks the current as 'deferred' (D) and traverses through each of the prerequisite nodes that have not been answered. The order in which the prerequisite nodes are visited is based on the edge weights, the highest first. Step 1 is complete once all prerequisite nodes are answered.

Step 2: Present the current node. After the traversal to the prerequisite children nodes is complete, the system passes the current node to the Guidance Generator along with information from any answered children and parent nodes to provide personalized guidance to the client as they attempt to answer the current node. The client will either be able to answer the node, in which case the state of the node will be marked as 'answered', or will not be able to fully answer the node, in which case the state of the node will be marked as 'unanswered'.

Step 3: Traverse to core contextual children nodes. Regardless of whether the current node is 'answered' or 'unanswered', the system traverses to the core contextual children nodes. While we allow the client to select the order of traversal, the traversal options are presented in a way that prioritizes nodes with higher edge weights. Step 3 is complete once all core children nodes are answered. If the current node is 'answered', the system is done with the current node and returns to its parent node.

Step 4: Re-present the current node. Next, the system attempts again to elicit information about the node from the client with the additional information. As before, the system passes the current node and information from all children and parent nodes to the Guidance Generator to provide further guidance based on the additional information. If the client is able to fully answer the node, the system is done with the current node and returns to its parent node.

Step 5: Traverse to supplemental children nodes. The system now attempts to utilize the supplemental children nodes to help the clients answer the current node. The module traverses to the existing supplemental children nodes.

If there are no more supplemental children nodes, the module generates a new supplemental children node that can provide additional context in answering the current node. Specifically, it attempts to generate a title and a question for the new node by leveraging the reasoning ability of the LLM and prompts it with the question of the current node and the (question, answer) pairs of the children nodes. The module computes the cosine similarity of the embedding vectors of the generated node and other nodes in the section. If any of the other nodes exceeds the empirically set similarity threshold of 0.7 and is not a child of the current node, the module adds the detected node as a new child node of the current node. If the generated node is unique, the module creates a new supplemental node marked in the interface as 'AI generated'. Throughout this process, the system returns to Step 4 to elicit information about the current node.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009