Automatic agent chaining for multimodal task support

Anonymous Author(s)
Affiliation
Address
email

Abstract

The future of human-computer interaction is moving toward systems where Large Language Models (LLMs) act as autonomous agents, capable of self-planning and adapting to complex, domain-specific tasks. However, a significant gap remains in developing agentic architectures that can seamlessly integrate into real-world, multimodal task support systems. We present our initial work on a novel agentic architecture for process task guidance, designed to assist human technicians in complex physical tasks. Our system develops automatic agent chaining features via dynamic planner that recruits specialized agents for task solving. To evaluate this approach, we collected a novel multimodal dataset of human-agent interactions during a toy assembly task and benchmarked our agentic system against a non-agentic baseline. Our findings show that the agentic solution significantly improves response quality and reduces incorrect outputs.

1 Introduction

Agentic architectures empower Large Language Models (LLMs) to act as autonomous agents, capable of making decisions, interacting with their environment, and achieving complex goals. This shift towards active interactions with the environment opens up new possibilities for human-agent collaborations and personalized assistance. We refer to agents, motivated by the very early works in the area of Artificial Intelligence, to mean any piece of program capable of perceiving, making decisions, and taking actions [1, 2]. Agentic systems in this work refer to 'agents' working with the user to collaboratively accomplish a user-centric goal.

The focus of this work is the exploration and development of an agentic solution for *process task guidance*. Process task guidance systems collaborate with a human task performer (i.e., technicians) to help them successfully accomplish a task. Agentic task guidance involves developing solutions that can provide intelligent and timely support to users. Users request assistance via queries and visual demonstrations. These queries can be complex and difficult to answer, requiring the system to understand user intentions, context, and rationale; validate its own responses; look up task specifications or other relevant information; and provide guidance relevant to the posed query. Agentic task assistance involves systems that provide responses via speech, text, and/or tool calls (functions that control peripherals to provide better task guidance).

Our agentic system consists of Planners, context enhancers, validators, and task or modality experts to assist users in accomplishing their goals. In this work, we propose a novel agentic architecture and a dataset for evaluating agentic solutions for process task guidance. The task requires constant environment perception and close coordination with the human task performer. Our agentic architecture involves automatic plan generation and chaining to automatically identify the support needed to assist the human in solving the task. We validated the agentic solution through quantitative and qualitative methods to identify potential issues in contextual robustness and chain-of-thought explainability, setting up promising avenues for future work. We also leverage human evaluations and LLM-as-judge for analyzing the results.

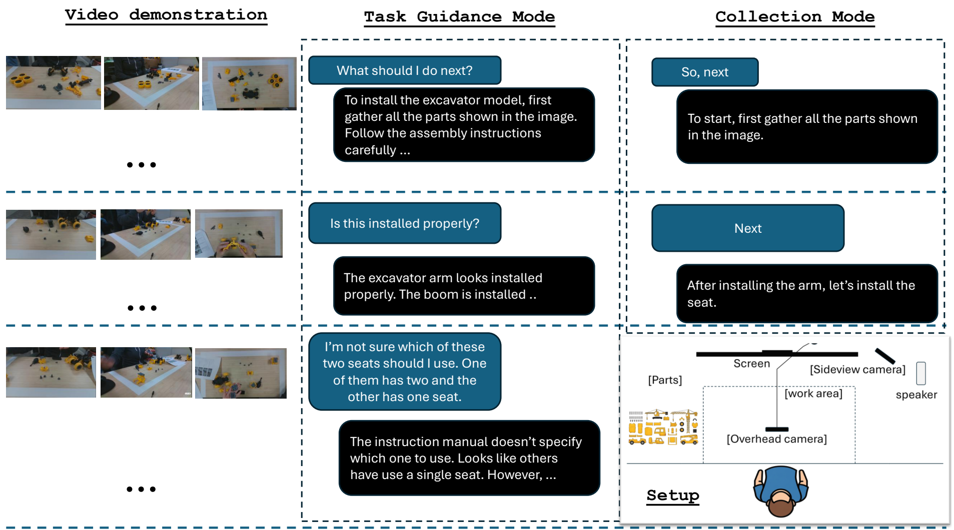

Figure 1: Shows sample conversations from the dataset. We collect the data in two modes: Task Guidance Mode and Collection Mode. The setup for data collection is shown in bottom right.

2 Data

In this study, we collected data from an assembly task in which participants interacted with an AI system to complete the construction of various toys. Participants were recruited through the company via email invitations sent to employees at the office. All participants were fluent in English and had normal or corrected-to-normal vision. Informed consent was obtained from all participants prior to the experiment, and the study was approved by the company's internal review board.

During the experiment, participants assembled four different toys while interacting with the AI system, which was presented in multiple modalities (detailed in Section 3). Participants captured images of their current assembly scene and asked verbal questions. The system responded with verbal guidance to assist in the task. All interactions were recorded for analysis. We collected a variety of data types from each participant, as summarized in the following table 2.

The audio recordings were transcribed using the Whisper automatic speech recognition (ASR) model. The images were standardized to a fixed resolution and annotated with toy type. The interaction logs were synchronized using timestamps to align queries, responses, and assembly progress.

3 Architecture

The perception component is the sensory system responsible for perceiving information from its environment. The system perceives the input via vision (video) and audio (speech). The perception module ingests, processes, and structures the raw data into a form suitable for the planners. The vision perception module converts the video into frames with logic to sample frames from the video. The speech perception system is a Whisper-based speech-to-text conversion agent. The perception system also keeps track of the conversation history and passes it along to the planner modules. The conversation history is limited by the context length limitation of the underlying models.

The planner is the agent's cognitive core responsible for agent recruitment for answering a given query. The planner has access to the agents in the environment and their task expertise. The agents that the planner chooses belong to the i) context enhancers, ii) solvers, and iii) validators classes. The agents are published to the environment with agent cards containing details about the agent, input format, output format, and relevant prompts.

The context enhancer acts as the agent's long-term memory, which enriches the solver's understanding by providing relevant external knowledge, preventing them from relying solely on the LLM's pre-trained (and potentially outdated) information. The core component of context enhancement is

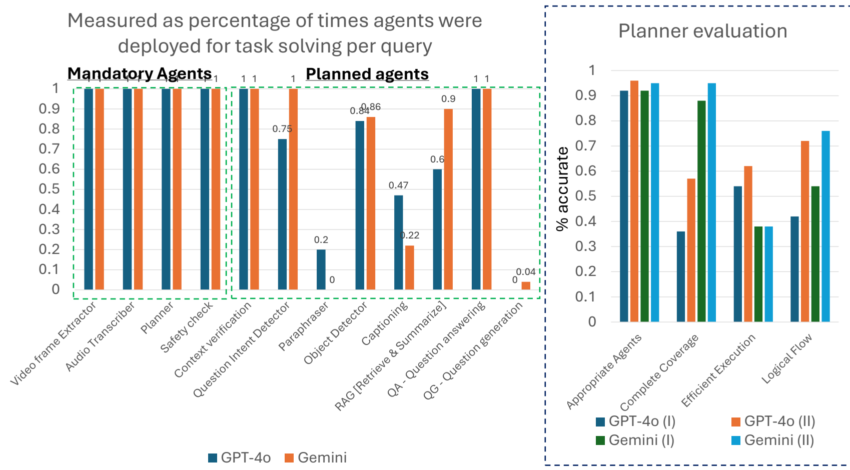

Figure 2: The left panel shows the percentage of times specific agents were deployed by a planner, with mandatory agents successfully deployed at a 100% rate. The right panel displays human-rated planner evaluations, highlighting strategic differences between different LLM backends. GPT-4o and Gemini models are compared both without (I) and with (II) a query type agent, demonstrating how this specific agent affects overall performance.

Retrieval-Augmented Generation (RAG) [3]. The indexing process relies on multimodal encoders which vectorize the contents and store them in vector stores (FAISS [4]). The retrieval process searches the vector database to find the most relevant chunks of text using the query and visual frames. The retrieved text chunks are added to the prompt that's sent to the solver, giving it the specific context it needs to form an answer. The solver is the core execution unit that performs a specific, well-defined task. The specific solvers in the system are vision expert systems such as object detectors, captioning, QA, Intent recognition, and question generation modules. The validators ensure that the system-generated responses are aligned with the system's goals and are safe. Additional context verifiers ensure that the prompt generated by the context enhancers is faithful to the query and the conversational context.

4 Experiments

The performance of an agentic baseline system is compared against a non-agentic one. The non-agentic system's approach is straightforward: it sends a prompt that includes both the user's query and a detailed instruction manual directly to a Large Language Model (LLM). In contrast, the agentic system would typically involve more complex processes as described in Section 3.

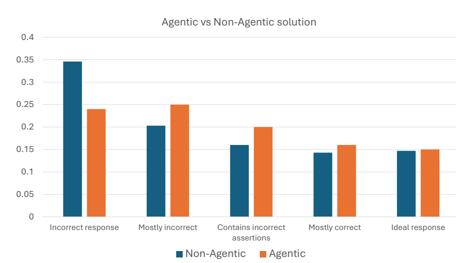

Figure 3: The agentic system demonstrates higher average response quality.

To measure the effectiveness of each system, human users annotated the generated responses. This evaluation uses a 5-point Likert scale, where a score of 1 signifies a completely incorrect or irrelevant response and 5 indicates a perfect answer. This method is similar to evaluation frameworks used in other benchmarks where an LLM judge assigns a correctness score to model outputs on the same 1 to 5 scale. However, in this work we employed experts to evaluate the responses.

A significant gap in research on LLM-based Multi-Agent Systems is the failure to adequately evaluate the outputs from the planner component. The plans themselves—the sequences of reasoning and actions an agent decides to take—are inherently challenging to evaluate automatically. This difficulty mirrors findings which reveal that assessing an agent's reasoning is substantially harder than evaluating its final, direct answer. Models consistently score lower on providing a justification for their answer compared to just giving the answer itself. This suggests that while we can often tell if a final output is correct, judging the quality and logic of the intermediate plan" or reasoning" that produced it remains a major hurdle in the field.

5 Results

This section presents the results of an experiment designed to evaluate the performance of different large language models (LLMs) as planners in an agentic workflow and to assess the impact of strategic agent deployment on task completion. Our analysis focuses on four key metrics: Appropriate Agents, Complete Coverage, Efficient Execution, and Logical Flow (refer A.2).

Our findings indicate that the choice of LLM significantly influences the planning strategy in an agentic system. We compared the planning behavior of GPT-4o and Gemini models across a series of human-annotated evaluations. While both models demonstrated comparable performance in selecting Appropriate Agents and maintaining a sound Logical Flow, their approaches to planning diverged significantly. GPT-4o model demonstrated a more strategic and efficient planning style. It selectively deployed agents to solve tasks, prioritizing an optimized workflow. This is evidenced by its higher scores in Efficient Execution. Gemini, in contrast exhibited a more exhaustive and expansive planning strategy. It tended to create broader plans that utilized more agents, even when not strictly necessary for efficiency. This behavior is reflected in its lower score for Efficient Execution. Despite these differences, both LLMs achieved similar high scores for Appropriate Agents and Logical Flow, suggesting that they are equally capable of identifying and sequencing the necessary agents for a given task.

The experiment also highlighted the critical role of specific agents within the workflow. The presence or absence of certain agents can have a significant impact on overall performance. For example, removing the query type categorizer from the agentic workflow resulted in a statistically significant drop in performance ($p < 0.001$, Wilcoxon rank-sum test). This finding underscores the importance of a comprehensive and well-structured agent toolkit, where each agent contributes to the overall success of the task.

Our analysis further revealed that agentic solutions consistently outperformed non-agentic solutions. Agentic workflows provide the ability to increase context, verify information, and plan more effectively, leading to superior task completion rates. The data also showed a clear distinction between the deployment of mandatory agents (e.g., Video frame Extractor, Audio Transcriber) and planned agents (e.g., Paraphraser, Object Detector). While mandatory agents were deployed at near 100% accuracy, the deployment of planned agents varied based on the LLM's planning strategy, further emphasizing the distinct behaviors of GPT-4o and Gemini.

Beyond the quantitative scores, our qualitative analysis revealed key insights into the user experience. Participants reported that the agentic system's responses felt more natural and context-aware, as the system seemed to "remember" previous steps and conversations. While the non-agentic system often repeated information from the manual, the agentic system's ability to use context enhancers resulted in more concise and personalized guidance. We also observed a clear preference for the system's ability to "speak" with them, which allowed them to keep their hands free to continue the physical task.

6 Future work

This work demonstrates the viability of an agentic approach for multimodal process task guidance systems. Our primary findings indicate that while an agentic system outperforms a simple baseline, its performance is highly contingent on the quality of the planner. A major limitation of our current work is that we leverage the zero-shot abilities of a pre-trained LLM for the planner module and do not perform any fine-tuning or explicit training on the planning task. Future work will focus on improving the planner, enhancing the real-time abilities and additional capabilities to enhance the proactive abilities.

References

[1] Oliver G Selfridge. Pandemonium: A paradigm for learning. In *Neurocomputing: foundations of research*, pages 115–122. 1988.

[2] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems*. Number 1. Oxford university press, 1999.

[3] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[4] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv preprint arXiv:2401.08281*, 2024.

[5] Rick Dakan and Joseph Feller. The 4d framework for ai fluency. 2025.

[6] Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, et al. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20270–20281, 2023.

[7] Ramesh Manuvinakurike, Sovan Biswas, Giuseppe Raffa, Richard Beckwith, Anthony Rhodes, Meng Shi, Gesem Gudino Mejia, Saurav Sahay, and Lama Nachman. Human in the loop approaches in multi-modal conversational task guidance system development. *arXiv e-prints*, pages arXiv–2211, 2022.

[8] Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint arXiv:2307.05300*, 2023.

[9] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *ICLR*, 2024.

[10] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024.

[11] Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje Karlsson, Jie Fu, and Yemin Shi. Autoagents: A framework for automatic agent generation. In *IJCAI*, 2024.

[12] Genghan Zhang, Weixin Liang, Olivia Hsu, and Kunle Olukotun. Adaptive self-improvement llm agentic system for ml library development. *arXiv preprint arXiv:2502.02534*, 2025.

[13] Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. *In The Thirteenth International Conference on Learning Representations.*, 2025.

[14] LangChain. LangChain. `https://github.com/langchain-ai/langchain`, 2022. Accessed: 2025-08-15.

[15] deepset. deepset studio, 2025. Accessed: 2025-08-21.

[16] Significant Gravitas. Auto-gpt. `https://github.com/Significant-Gravitas/Auto-GPT`, 2023. Accessed: 2025-08-15.

[17] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.

[18] Aymeric Roucher, Albert Villanova del Moral, Thomas Wolf, Leandro von Werra, and Erik Kaunismäki. 'smolagents': a smol library to build great agentic systems. `https://github.com/huggingface/smolagents`, 2025.

[19] Jesper Simonsen and Toni Robertson. *Routledge international handbook of participatory design*, volume 711. Routledge New York, 2013.

[20] Mona Sloane, Emanuel Moss, Olaitan Awomolo, and Laura Forlano. Participation is not a design fix for machine learning. In *Proceedings of the 2nd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–6, 2022.

[21] Seth Bullock, Nirav Ajmeri, Mike Batty, Michaela Black, John Cartlidge, Robert Challen, Cangxiong Chen, Jing Chen, Joan Condell, Leon Danon, et al. Artificial intelligence for collective intelligence: a national-scale research strategy. *The Knowledge Engineering Review*, 39:e10, 2024.

[22] Yifan Li, Yuhang Chen, Anh Dao, Lichi Li, Zhongyi Cai, Zhen Tan, Tianlong Chen, and Yu Kong. Industryeqa: Pushing the frontiers of embodied question answering in industrial scenarios. *arXiv preprint arXiv:2505.20640*, 2025.

[23] Pierre Sermanet, Tian Ding, Jeffery Zhao, Fei Xia, Debidatta Dwibedi, Keerthana Gopalakrishnan, Christine Chan, Gabriel Dulac-Arnold, Sudeep Maddineni, Nikhil J. Joshi, and et al. Robovqa: Multimodal long-horizon reasoning for robotics. In *ICRA*, pages 645–652, 2024.

[24] Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

[25] Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, Peifeng Wang, Silvio Savarese, et al. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. *arXiv preprint arXiv:2504.09037*, 2025.

[26] Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. An overview of catastrophic ai risks. *arXiv preprint arXiv:2306.12001*, 2023.

A Appendix

A.1 Why is an Agentic Solution a Fit for Process Task Guidance?

Physical tasks requiring process guidance are complex. They contain multiple steps and require users to identify both objects and actions with which they may be unfamiliar. Nevertheless, users have some actions at their disposal to navigate that complexity: they can manipulate their physical workspace (e.g., holding objects, pointing, etc.) and verbalize questions (e.g., "What do I do next?") about the task when in need of guidance. LLMs (including VLMs) are a useful match for both of these physical and verbal actions, which makes physical tasks and their navigation well-suited to an LLM-based agentic solution. Such a solution supports the needs of both humans and AI in socio-technical systems; the human and the AI system can, supported with an agentic solution, delegate tasks to one another, describe goals in ways that produce useful outputs, discern how useful those outputs are, and diligently take responsibility for those outputs (following [5]).

To understand what could be usefully automated, we carefully studied tasks that are central to technicians' work, and the context in which those tasks are embedded. We collected a dataset consisting of questions technicians asked in the course of using a remote-controlled task guidance system (similar to [6, 7]). The guidance could be provided via answers to the queries, proactively while the technician is performing the process, or even post-hoc after the step is completed via feedback. The responses themselves can be provided in numerous ways, including via text (chat), voice output, peripheral controls (navigating the dashboard a technician sees to the relevant parts, visualizations), augmented reality, or even robot controls. Given the nature of the process and the size of the objects, the most viable guidance we could provide in real-time was via speech and peripheral controls. We leave expanding the modality of generated responses for future work.

#processes	
Minutes	
# questions	
...	

LLM-as-Judge	
Response ratings	
...	

Table 1: Dataset stats and results

To understand how augmentation creates more value than working separately, we identified the logistical and cognitive burden of following a lengthy process specification, invoking multiple difficult-to-recognize components while also physically manipulating those components and necessary tools. This burden, we determined, could be lessened for the human user by an agentic solution that could describe each step of the process and help identify each component needed for that step through spoken language interactions.

Humans should remain in control of manipulating the physical components and completing the steps. The agent's role is to keep the human 'on task' without requiring them to scroll through an online interface or turn pages in a printed guidebook, thereby freeing their hands for the physical task.

We developed tools to help agents execute software/dashboard-related tasks, providing support for tasks carried out in the physical space. Our design consideration was that agents should not carry out the process in the physical space themselves yet, but only provide guidance via speech and peripheral control to the humans carrying out the process. This is in part because humans are far better and faster at carrying out the steps involved in the processes. Developing robotic solutions is also expensive and time-consuming with steep learning curves. As VLAs (Vision Language Action) models become more prevalent and integration with Agentic flow becomes easier, it remains an area of interest for process task guidance.

Current works don't study the agentic interactions and focus instead on final task performance evaluations, which is a key gap our work addresses.

A.2 Other details

Appropriate Agents: This metric assesses whether the LLM's generated plan included the correct and relevant agents necessary to address the given query.

Complete Coverage: This evaluates how effectively the plan, when executed, would lead to a comprehensive and complete answer to the user's question.

Efficient Execution: This measures the plan's efficiency by determining if the LLM selected and deployed only the essential agents, avoiding unnecessary steps or redundant agent calls.

Logical Flow: This metric judges the coherence of the plan by examining whether the sequence of agent deployments and actions makes logical sense from start to finish.

B Related work

An agentic system adaptively improves itself for problems that cannot be adequately specified in advance [1, 2]. There are several ways agentic systems achieve this. An agentic system could be comprised of a multi-agent cognitive synergist, where multi-turn self-collaboration with different personas is employed by a central planner that dynamically identifies personas to combine strengths and knowledge for complex problem-solving [8]. An alternative approach doesn't rely on personas but rather a group of agents recruited to form an expert group, with each agent solving a task democratically which is then acted upon in the environment by experts [9].

Recent advancements in agentic systems are shifting the focus from single LLM instances to structured, autonomous, and self-improving multi-agent frameworks. A primary trend is the formalization of collaboration to enhance efficiency and reliability. For instance, MetaGPT [10] tackles this by encoding human-like Standardized Operating Procedures (SOPs) into a multi-agent system, assigning specific roles and structured workflows to mirror an efficient organization. This approach is further automated by frameworks like AutoAgents [11], which dynamically generates a team of specialized agents and a collaboration plan tailored to a specific task, thereby reducing the manual design effort.

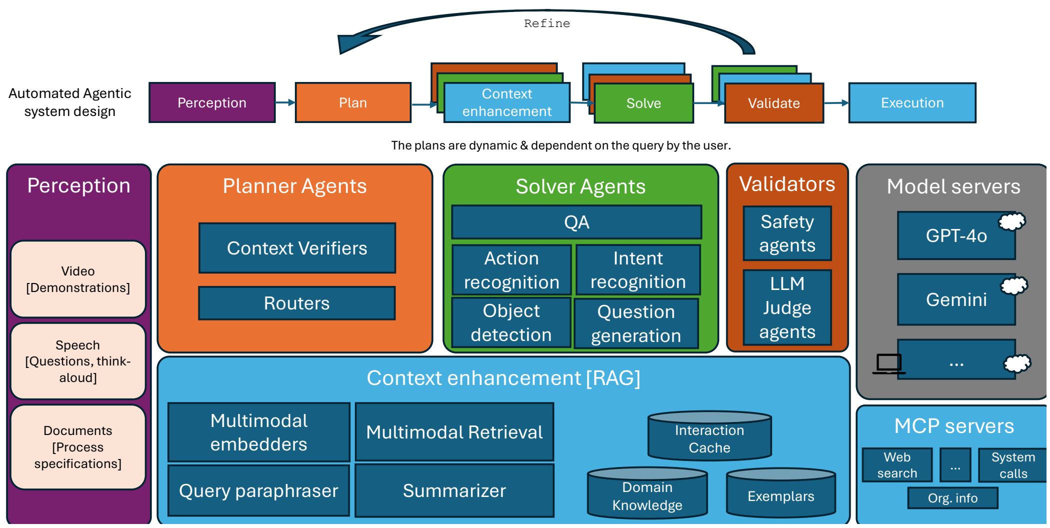

Figure 4: The agentic architecture for process task guidance is shown here. The system employs a planner to orchestrate various specialized agents, including context enhancers, solvers, and validators, to provide accurate and relevant support to the user.

These developments aim to create more coherent and capable systems that can handle complex, multi-step problems with greater autonomy.

Simultaneously, another key research thrust is enabling agents to learn and adapt without direct human intervention. The work on Adaptive Self-Improvement [12] introduces a mechanism for agents to reflect on their past actions, identify errors, and build an internal knowledge base to improve future performance through a closed-loop learning process. This capacity for autonomous evolution is being applied in practical domains, as demonstrated by ADAS [13], a system that automates the creation of decision-making agents for complex simulations. By generating diverse agent personas and behaviors automatically, ADAS enables sophisticated modeling of socio-technical systems. Together, these works illustrate a move toward building more dynamic, intelligent, and scalable agentic ecosystems that can learn, collaborate, and be deployed with increasing ease.

B.1 Tools for Agentic System Development

The landscape of agentic system development is defined by a range of open-source tools that offer different levels of abstraction and control. Frameworks like LangChain [14] and Haystack by deepset [15] provide the foundational building blocks for creating custom applications. LangChain offers a highly modular and extensive set of components for chaining" together LLM calls with data sources and APIs, making it a versatile choice for a wide array of tasks, particularly Retrieval-Augmented Generation (RAG). Similarly, Haystack provides a robust, pipeline-centric architecture of nodes for building production-grade semantic search and question-answering systems. These frameworks empower developers by providing the essential glue" to construct complex, context-aware applications from the ground up.

In contrast to these foundational frameworks, other tools offer more opinionated or autonomous approaches. AutoGPT [16] pioneered the concept of a fully autonomous agent, demonstrating how an LLM could use a self-prompting loop of thought, reasoning, and tool use to pursue high-level goals without direct human oversight. For more structured workflows, Microsoft's AutoGen [17] introduces a multi-agent conversational paradigm, where specialized agents collaborate by talking" to each other to solve problems, offering a flexible way to orchestrate complex tasks. At the same time, tools like Smol Agents [18] focus on a specific application—code generation—by acting as a developer-in-a-box" that scaffolds an entire codebase from a single, high-level user prompt. Together, these tools span the spectrum from providing granular components to delivering fully autonomous or specialized agent solutions.

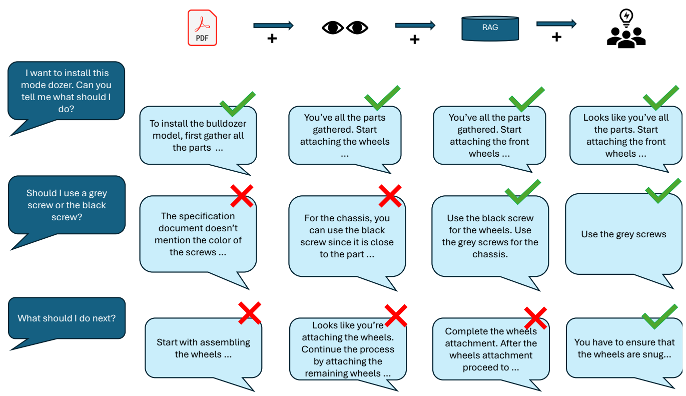

Figure 5: An example of an agentic architecture for a multimodal task support system. The planner dynamically recruits agents to handle perception, context, and validation, ensuring a robust and accurate response.

As agentic systems become more prevalent and complex, there is a need for human-centered participatory designs for such systems [19]. While not a total solution to all the potential harms of integrating AI systems into the works [20], participatory design methods can help researchers to more inclusively consider the needs of the people and communities who will be using these technologies, gather critical data on acceptance, adoption and continued use, and integrate community concerns and priorities into design and engineering practices [21].

B.2 Datasets and Surveys

[22], [23] [24], [25], [26]

Data Type	**Description**
Audio	Transcribed spoken questions from participants.
Image	Scene snapshots taken by participants during each toy assembly.
Response	Verbal or textual guidance provided by the LLM.
Meta Information	Timestamps, toy type, query-response pairs, and assembly stage.
Demographics	Participant age, sex, profession, experience with AI, and other background info.

Table 2: Summary of collected data types and their descriptions.