

Collaborating with LLMs Through a Voice and Graphical User Interface

Lenja Sorokin

BMW Group Research and
Technology
Munich, Germany
lenja.sorokin@bmw.de

Khanh Huynh

BMW Group Research and
Technology
Munich, Germany
LMU Munich
Munich, Germany
khanh.huynh@bmw.de

Malin Eiband

BMW Group Research and
Technology
Munich, Germany
malin.eiband@bmw.de

Lukas Stappen

BMW Group Research and
Technology
Munich, Germany
lukas.stappen@bmw.de

Jeremy Dillmann

BMW Group Research and
Technology
Munich, Germany
jeremy.dillmann@bmw.de

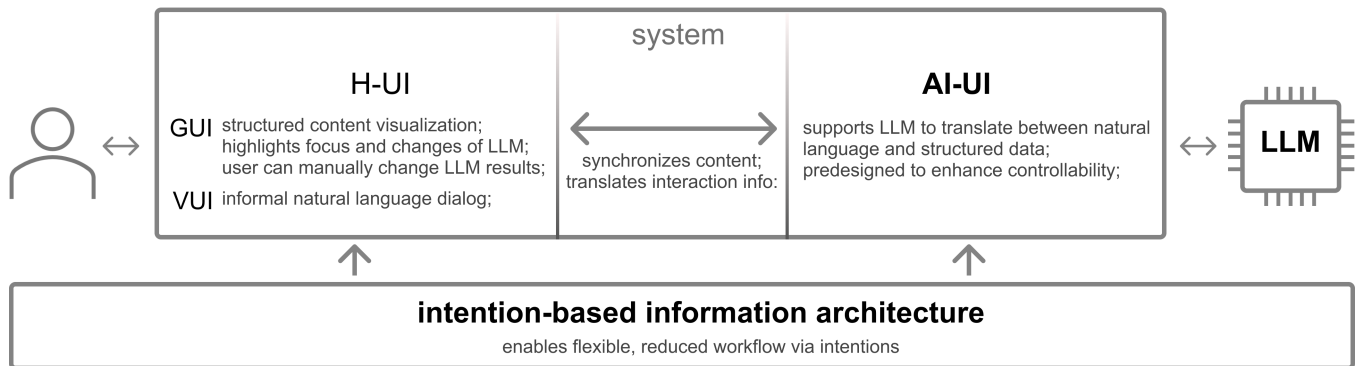


Figure 1: Integrating a Large Language Model (LLM) with a graphical and voice user interface (H-UI) introduces UX conflicts. Approach: shaping LLM as a second user with a tailored interface (AI-UI), based on an intention-based information architecture.

Abstract

The integration of Large Language Models (LLM) into systems that combine voice and graphical user interfaces causes a paradigm shift from designing for humans only to designing for collaborative interaction. However, the non-deterministic, natural language-based nature of LLMs also causes conflicts with the deterministic, structured GUI interaction. We argue that to seize collaboration and mitigate conflicts, UX designers must design the LLM as a second user, creating one interface for human users and one for the LLM. We designed an exemplary prototype as a case study, identifying through this hands-on process three unique challenges for practical UX design: (a) Since humans express objectives rather than commands, the information architecture should be intention-based; (b) the system should translate interactions between human and AI

to foster mutual understanding; and (c) the natural language dialog with the LLM should be integrated with the structured interaction typical for GUIs through targeted use of modalities.

CCS Concepts

• **Human-centered computing** → **Human computer interaction (HCI)**.

Keywords

Human-AI Collaboration, Multimodal Interaction, UX Design, Large Language Models, Graphical User Interfaces

ACM Reference Format:

Lenja Sorokin, Khanh Huynh, Malin Eiband, Lukas Stappen, and Jeremy Dillmann. 2025. Collaborating with LLMs Through a Voice and Graphical User Interface. In *Adjunct Proceedings of the 27th International Conference on Mobile Human-Computer Interaction (MobileHCI '25 Adjunct)*, September 22–25, 2025, Sharm El-Sheikh, Egypt. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3737821.3749555>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
MobileHCI '25 Adjunct, Sharm El-Sheikh, Egypt
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1970-7/25/09
<https://doi.org/10.1145/3737821.3749555>

Table 1: Conflicts arising from the nature of LLMs and the user’s expectation of GUIs

	LLM	GUI (traditional)
Behavior	non-deterministic	deterministic
Input paradigm	intention-based	command-based
Dialog style	informal, natural language	formalized, structured data

1 Introduction and Background

Large Language Models (LLMs) are fundamentally changing interaction: LLM-driven Voice User Interfaces (VUIs) offer more natural interaction, better handle ambiguity and the unpredictability of spoken language, and support complex multi-turn conversations, outperforming traditional rule-based systems [11, 26, 27]. Their advanced capabilities are driving a paradigm shift from using *systems as tools* for the execution of commands with a precise expectation of the outcome to experiencing them as collaborative partners, capable of interpreting intentions and generating context-driven content (*system as a partner*) [2, 13, 15, 36]. This idea evolved in the field of HCI, where researchers have explored how such systems and humans can jointly solve problems as active participants [28].

However, the potential for collaborative work, where two or more partners work together toward a shared goal on shared content [12, 41], remains limited without integration with a Graphical User Interface (GUI). GUIs are essential for many tasks – particularly for collaborative content work – as they provide a comprehensive view of information and allow for direct content modification [43]. We anticipate that LLM-based VUIs will therefore progressively merge with GUIs wherever creating, viewing, or changing content is relevant, e.g. in mobile phones, smart glasses, smart home assistants, and automotive environments [1, 31, 39, 40].

Prior research in the HCI field has explored multimodal interaction, notably combining speech and gestures to support user input in GUIs [1, 6, 8, 30]. These early works demonstrated how pointing or touching could be integrated with speech to disambiguate references and enhance the interpretation of users’ commands by systems. More recent systems, such as PixelTone [22], work by Lee et al. [24] or Valletto [19] demonstrated how speech and touch could be combined for more intuitive interaction. However, these systems rely on rule-based approaches limiting natural language interaction. Introducing LLMs extends the capabilities for interpreting user intentions across multiple modalities, including unpredictable ones, more closely resembling human-to-human interaction [26, 27].

At the same time, integrating LLMs with GUIs poses new challenges: VUIs inherently fit the non-deterministic nature of LLMs and their design for informal interaction through natural language [11, 27]. In contrast, user expectations of GUIs focus on deterministic and structured interaction [33], which inherently differ from LLM behavior. Moreover, traditional user interfaces are designed exclusively for human users, assuming full initiative and control [17], while LLMs introduce substantial UX challenges like limited controllability and unexpected output [9, 38]. From these differences, we identify three core conflicts for design, outlined in **Table 1**: (1) While users expect deterministic behavior from GUIs, LLMs are non-deterministic systems [16, 34]. (2) LLMs are built for intention-based interaction, while GUIs are command-based. (3) LLMs support informal natural language dialog, while GUIs are a

highly formalized dialog system with a structured presentation of information [32].

Design guidelines exist for interacting with Artificial Intelligence (AI) [2, 4] and for combined voice and graphical user interfaces [44]. However, neither of them provides design considerations for the conflicting nature of traditional GUIs and AIs like LLMs. In this paper, we examine these conflicts through the *system as a partner* interaction paradigm. In particular, we aim to leverage the capabilities of LLMs for effective, efficient work on *alphanumeric content*, while making them controllable where possible (e.g. enabling manual changes and reversion in the collaboration history [2, 4]). To reflect the inherent limitations with regard to controllability, we design the user experience of the interaction with the LLM to resemble an interaction with a second user who is semi-autonomously working on the content. As such, a key design decision is the creation of two interfaces, one for the human user and one for the LLM as a user.

We explore how these ideas can be put into practice by building a prototype integrating an LLM and enabling interaction both via VUI and GUI modalities. Drawing on the insights from designing the prototype, we derive three challenges UX designers face in this emerging context – both in terms of the design process itself and the interface and system design –, and give suggestions on how to address them. These suggestions are intended not as definitive guidelines, but as a foundation for community discussion and exchange, aiming to inspire further exploration and development in this field.

2 Prototype Development Using a Research-through-Design Approach

We designed an LLM-based system for collaborative creation of alphanumeric content via a VUI and a GUI, such as with a mobile device or built into a car interface for a passenger. Our prototype enables the user to communicate an intention via voice interaction and then refine results via voice and touch. The LLM is used to generate and modify content, but not to generate the layout or style of the GUI. We decided on the creation and modification of a list and an email as exemplary use cases, because they are general, frequently performed tasks, in which utilizing an LLM can help substantially [45]. Also, they involve a variety of interactions, text formatting, and UI elements. In our user journey, the user has the goal to plan a sightseeing trip with a colleague on the next day. It includes creating a list of sights via the LLM, changing the list, and writing an email referring to the list (see **Figure 2a**).

The input modalities are voice and touch, the output modalities are sound and visual feedback on a central information display. A cloud-based speech processing service (Alexa) was integrated. However, the heart of the system contains the business logic in natural language and manages the communication with the LLM (OpenAI’s GPT-4o mini) as well as the user input (via voice and



(a) User journey: multimodal, LLM-supported creation and adaptation of a list and an email based on the list



(b) H-UI: list + email collaboratively created with LLM, focus and changes of LLM highlighted, "AI home" on top

Figure 2: The prototype: The flow across the user journey (a) and a screenshot of the H-UI (b)

touch). Among other things, the system enables the LLM to trigger actions (both in the business logic and the GUI) by calling predefined tools via function calling [37]. That way, the LLM can, for example, trigger the display of generated content in the GUI via a function provided by the system.

The goal of this early research phase was to investigate the conflicts listed in Table 1 and to develop a design approach and artifacts to address them. We employed a research-through-design process [46], developing a prototype informed by findings from related work, as prior research has demonstrated that prototyping *per se* is valid to provide valuable insights [10, 48].

In our process, we (1) extracted from desk research the conflicts described in Table 1 and an initial set of design principles; (2) defined high-level tasks derived from common list and email applications as well as associated intentions considering the LLMs capabilities; (3) created an information architecture based on these intentions; (4) created a basic layout adhering to the design principles and a user journey including wireframes; (5) iteratively developed a prototype, refining the information architecture with information needed by the LLM and engaging in informal discussions with over 40 professionals in UX and AI software development.

While not part of a formal study and data collection, the discussions exposed recurring design tensions and informed design decisions. We conducted 16 sessions, each with up to 6 experts, spread over a prototype development period of 8 months. Each session consisted of (1) a short introduction to the goal of the research and the resulting conflicts as described in Table 1 (2) a demonstration of the prototype based on the user journey (3) a free exploration of the prototype by the experts and a discussion on challenges, design decisions and next steps. These discussions directly informed the respective next cycle of design decisions and development.

Questions that arose during the iterative design were documented along with the solutions we decided upon based on related work, such as human AI guidelines and prior work on multimodal interfaces, lessons learned from prototype development, and the informal expert discussions. The following chapter reflects on these insights and our design approach, but does not claim generalizability or empirical validation.

3 Insights from the Prototype: Key Design Considerations for the LLM as a Second User

As outlined in the introduction, we decided to design the users' experience of the LLM as a second user. We therefore developed two distinct types of interfaces: a Human-oriented UI (H-UI), optimized for human ergonomics, and an AI-oriented UI (AI-UI) for the unique capabilities and limitations of LLMs (see Figure 1). Using a traditional H-UI would be inefficient for LLM interactions, as it would force the LLM to use modalities optimized for human ergonomic and cognitive capabilities rather than computational efficiency. Similarly, traditional application programming interfaces (APIs) typically facilitate structured, deterministic data exchanges between software systems, which limits their ability to leverage the natural language understanding, probabilistic reasoning, and context-sensitive strengths of LLMs. Therefore, an effective AI-UI must bridge this gap by providing a structured toolkit described clearly in natural language, enabling LLMs to autonomously select and apply tools within a mixed-initiative framework.

The following sections present the insights we gained while designing and developing the H-UI and the AI-UI, which can be summarized in three key design considerations for UX practitioners: (a) Designing the interfaces via an intention-based information

architecture, (b) aligning user and LLM understanding via collaborative interactions, and (c) integrating natural language interaction driven by the LLM with structured GUI interaction

3.1 Designing the interfaces via an intention-based information architecture

Traditional UIs without LLM integration either structure the tasks according to the domain, presenting the related functions and informational objects in parallel or, if more guidance is necessary, utilize a rigid workflow. In contrast, we propose an *intention-based* information architecture facilitating a flexible and reduced workflow (via the dialog with the LLM) in combination with a permanent domain-specific representation of functions and information objects (via the GUI e.g. within apps). This promotes efficiency, effectiveness, and controllability as it offers a high-level of guidance via the LLM dialog, yet it leaves the choice to the user to utilize the guidance or select another sub-task from the GUI to work on. This is illustrated in step 4 and 5 of Figure 2a: The user utterance “create an email to Mary ...” will trigger (1) an LLM dialog providing guidance by requesting missing information, and (2) the visualization of the result in the email app, providing a structured overview of the generated content and interaction options for manual changes [2, 4]. Furthermore, an efficient workflow is promoted by automating sub-tasks and capitalizing on the LLMs potential to complement information using, for example, the dialog history or content from another app. In the email example the LLM complements the email with details from the list, illustrated in Figure 2b.

3.1.1 Design for flexible interaction based on high-level tasks. Supporting the user to interact via intentions will reduce the number of steps and the granularity of user input. Users no longer need to precisely plan workflows, instead they can iteratively refine outcomes together with the LLM. This improves interaction efficiency and reduces mental load [5]. However, when developing the prototype, we had to realize that more implicit user expectations require more precise instructions for the LLM. This asks for extra effort during design time. We had to create an overview of the tasks implicitly included in each intention. As an example, we abstracted the intention “create an email with ...” to a high-level task (create mail) and split it into sub-tasks like “open email app”, “create subject”, “specify recipient”, ..., “send mail”. We decided to automate all sub-tasks except “send mail”, for controllability reasons. We also defined where to leverage the LLM for automation and where a deterministic approach is more beneficial [42], avoiding the LLM’s unexpected behavior, latency [7, 34], and power consumption [25]. In addition, we specified the expected user input and LLM output (e.g. the recipient’s email address) and defined strategies for mitigating missing user input (e.g. looking up the recipient’s email address in the address book or asking back to the user to choose the intended recipient, as in step 4 of Figure 2a).

The resulting overview we refer to as intention-based information architecture. From our work on the prototype, we derive that designers need to specify this to support the flexible, reduced workflow as well as the structured representation of the LLM’s results, and must pay special attention to the following aspects:

- (1) Translate high-level tasks into sub-tasks.

- (2) Determine the sub-tasks that could be automated and whether the LLM or a deterministic approach is more beneficial.
- (3) Specify the information needed for automation by the LLM and the content and format of the output expected.
- (4) Provide information to mitigate incomplete task descriptions.

3.1.2 Design the AI-UI using the intention-based information architecture. Based on the resulting information architecture, both the H-UI and the AI-UI need to be designed, the latter comprising of context information including the dialog history and a set of tools usable by the LLM. In collaboration with software engineers, UX designers must divide the task hierarchy into these tools. Tools must include clear task descriptions, helping the LLM to choose the appropriate one for the current task. In addition, UX designers need to support software engineers in translating the information objects and their relationships from the information architecture into data structures provided to the LLM. An example from the email use case is specifying the structure of the input the LLM needs to give to the email creation tool, such as strings with the recipients’ email address and the message body. By providing the tools and supporting their selection, the AI-UI enables the LLM to facilitate sub-task automation and compensates for incomplete task descriptions by using context information or requesting targeted input from the user.

3.1.3 Design the interfaces for a ubiquitous LLM. In command-based systems, UX designers structure the information architecture within the boundaries of one app. However, the intention of a user per se is independent of an app and can include several (high-level) tasks, for instance, planning your birthday party might involve finding a location, planning the buffet, sending invitations, and finding decoration ideas. Part of the appeal of conversational user interfaces (CUI) stems from the ability to interact independently of the narrow boundaries of functionality of a particular application [47]. However, in order to effectively work on content, we still require the functionality and content visualization in well-established apps. During the development of the prototype, it became clear that the LLM needs to be able to operate across apps to unleash the potential of intention-based interaction in this setup. As an example, for creating the email based on the list, it needed to still have access to information from the list, autonomously switch to the email app and to show this by highlighting the app, as in step 5 of Figure 2a. Insights from prototype development indicate that the LLM should:

- (1) Maintain a shared context or knowledge base across apps to reuse content from previous interactions.
- (2) Autonomously select, open and switch apps as needed.
- (3) Have a ubiquitous visual representation for interaction, for instance, to highlight, overlay, or augment elements, as a layer on top of the deterministic system. In environments like vehicles, this might extend beyond displays to physical controls or objects being controlled by the LLM.
- (4) In line with the guideline of Amershi et al. [2] to “Provide global controls”, a central, task-independent “AI home” should provide users with easy access to overarching functions to control the state and chat history (e.g. buttons for

AI on/off, deleting the dialog history, and reverting the last interaction, as illustrated in [Figure 2b](#)).

3.2 Aligning user and LLM understanding via collaborative interactions

The goal is to support the human to understand the LLM and vice versa, and to impede a mismatch of data shown in the GUI and the context the LLM works with. Mutual understanding ensures effective collaboration, preventing and handling errors, and facilitating a seamless flow of interaction [43]. However, as a human and an LLM work via different interfaces on the content they are not automatically aware of each other's actions. The system therefore, needs to transfer the information about each interaction step to the other user. This information consists of the following components: `<ACTION><OBJECT><CONTEXTUAL INFORMATION>`. Sending information about the actions and objects promotes a common focus, as whenever we show a dialog partner what our focus is, we also influence their focus.

3.2.1 Inform the user of the LLM's focus and changes via the H-UI. When the LLM performs an action in the system (such as creating and modifying a list in [Figure 2](#)), the user's attention should be guided towards the working area of the LLM and the result of the last interaction step to promote a common focus and facilitate reviewing of changes, aligning with prior work on transparency in human-AI collaboration [2, 4, 43] and joint attention in interactive systems [21]. To make this possible, the user needs to be able to know which app the LLM is currently using via the H-UI (e.g. by opening and highlighting the app). For a more detailed focus, displayed objects created or modified by the LLM should be recognizable. Highlighting changes, as the "45 min." in [Figure 2b](#), facilitates their review, thereby supporting efficient correction [2]. Inline with transparency goals [2, 4], users should be able to access additional information on how the prompt was interpreted by the LLM and what actions were taken (e.g. in an overlay showing the dialog reply, accessible via the "AI home"; see [Section 3.1.2](#)). Finally, consider informing the user about the context used by the LLM e.g. this could also be done via displaying relevant information bits in an overlay via the "AI home" [4].

3.2.2 Make user interaction understandable for the LLM. System support for LLM understanding of user interactions can be split into two cases. Firstly, if the user interacts with the system without utilizing the LLM, such as when deleting the list items via touch in step 2 of [Figure 2a](#), the system needs to inform the LLM of the result via the AI-UI. Manual changes of content and other results where the LLM was not involved need to be written in the history of the LLM's context, so the user and the LLM are on the same page.

Secondly if the user prompts the LLM with their intention, the system should make it easier for the LLM to understand the prompt, as well as support the human user to formulate their prompts. The following aspects could be considered:

- (1) Reference objects across modalities: Transfer object references (e.g. list item selected via touch) via the AI-UI to the LLM prompt. Complement voice input with multimodal signals (e.g. touch, gesture and/or gaze) to clarify objects or locations referenced by the user (see chapter 3.3.2) [24].

- (2) Simplify GUI references: Design GUI elements to be easily referenced in prompts, using an unambiguously referenceable layout, distinctive color, concise labels, or visually distinct icons. For textual content, use paragraph or line numbers for easy reference.
- (3) Provide relevant contextual information: Since there is evidence that LLM output is improved by context [29], identify context crucial to user intentions (e.g. the user's location, time, and weather conditions) and transfer it to the LLM via the AI-UI. When the user is interacting with the display, explicitly provide visible content either as textual context or image input for the LLM.
- (4) Align language consistently: Employ the users' typical choice of words not only for the GUI but also for the tool description in the AI-UI to increase the likelihood of selecting the right tool and wording. There is evidence that LLMs are sensitive to the exact wording of the prompt for calling tools [37].

3.3 Integrating natural language interaction driven by the LLM with structured GUI interaction

Our goal here is, on the one hand to leverage the full potential of the open-ended dialog of an LLM, and on the other hand, to leverage a GUI to display content and facilitate efficient interaction. The resulting challenge is that we need to integrate an informal dialog (with the LLM via VUI) [17] with a highly formalized, structured dialog system (the GUI).

3.3.1 Output: "formalize" the dialog where necessary. When designing for the most suitable output modality, we found that when the LLM's result contains content creation or changes to content, it should be displayed in the GUI. Voice output is the better solution only when the user's attention needs to be drawn [20] and the LLM has not altered content but rather replied as part of a multi-turn chat-like informal dialog (e.g. when requesting additional information or stating that the intent could not be processed) [27]. When designing for GUI output, the genuine style of an LLM reply, natural language, is not suitable. Instead, the reply should be formalized to fit well-established structures for the type of information. As an example, an email created by an LLM should be displayed in a typical GUI format with the recipient, subject, and content. The data is structured in the information architecture, which is the basis for the GUI template as well as for the tool description of the AI-UI. This ensures the content structures in the H-UI and the AI-UI are aligned.

3.3.2 Input: Design for the most efficient modality and seamlessly combine them. Voice input is particularly beneficial for communicating intentions that the system cannot anticipate, as there is no need for a predefined graphical representation, and through the integration of the LLM, the unpredictability can be managed [27]. This might be intentions which contain one or more high-level tasks like "create an email" as well as more specific intentions like "I need a more inspiring title". Also, especially when there is no hardware keyboard available, voice input is great for alphanumeric input. On the other hand, voice is in many cases very inefficient in comparison to a formalized input via e.g. touch. As an example,

when the user wants to delete several items from the list, he would need to read out or describe each item in addition to the action (delete). Via touch, this might be done a lot faster by tapping the delete buttons of the items, as in step 2 of Figure 2a. Whenever the user references a GUI element, formalized input is beneficial, as it is fast and unambiguous. It is especially suitable for selection (e.g. of a GUI element or of a position within a GUI element) and for activation of a predictable function.

The full potential of a multimodal LLM-based system for increasing efficiency is unleashed when combining the advantages of the input modalities in one interaction: e.g. the user selects a list element via touch and changes it via voice using the LLM as in step 3 of Figure 2a. As shown in prior work by Bolt [6] and Lee et al. [24], such modality combinations not only enhance efficiency but can also be used to decrease ambiguity e.g. when using gaze, touch, and gesture in addition to the language command to optimize interpretation of which object the user is talking about.

4 Discussion and Future Work

Having outlined specific design considerations from our prototype development, we now reflect on the broader implications and future work. The insights address critical challenges in bridging the gap between the non-deterministic nature of LLMs [16, 34], built for an informal, intention-based natural language dialog [13], and the structured, formalized interactions and deterministic responses typically expected in GUIs [18, 23, 33]. Following the *system as a partner* paradigm, we integrate the LLM as a second user. As a consequence, we propose to design two interfaces, an H-UI for the human user and an AI-UI tailored to the unique capabilities and limitations of the LLM. The AI-UI is a key enabler for the intention-based user input, the translation between informal LLM dialog and formalized GUI interaction, the exchange of contextual information, and for supporting a common focus between user and LLM. In addition, the AI-UI is a lever for controllability, allowing to steer the LLM towards using an adequate tool and creating appropriate output.

Both interfaces, AI-UI and H-UI, need to be designed for mutual understanding. In the H-UI, the collaboration is made visible for the user [43]. However, much of UX design for LLM integrated systems does not have direct visual impact as traditional UI design would have, but is rather about the orchestration of the collaboration, and is implemented in the background of the system [2, 7]. It is built on the intention-based information architecture, which allows the UX designer to optimize for efficiency, effectiveness, and controllability by defining what is to be automated, when to support formalized GUI interaction vs. informal LLM dialog, and to specify the structure of the AI-UI. The intention-based information architecture also promotes the careful decision for which task the LLM provides significant added value over its drawbacks, such as latency, lack of controllability [7, 34], and high economic and ecological costs [25]. In many cases, such as when the result expected by the user can be predefined, a deterministic approach is more advantageous. A targeted use of the two paths and their deep integration is necessary.

4.1 Towards evaluation of design considerations and generalizability to other use cases

We focused on collaborative work on alphanumeric content via the use cases of creating lists and emails with a voice interface and a touch screen as a GUI. A flexible combination of interaction modalities and exploiting generation and structuring based on minimal user input appeared to enhance the UX in our setting. However, the insights presented may not be directly applicable to other use cases or contexts. More research is needed to evaluate the generalizability of the design considerations and their applicability to a broader range of scenarios, as for collaborative work on non-alphanumeric content like images. Additionally, our work does not provide an in-depth, formal evaluation of the proposed approach with UX designers, and the design considerations have not been tested with end-users. Collecting user feedback and assessing the real-world effectiveness and UX of the LLM-integrated multimodal interface is an important next step for refining our insights and confirming their practical implications.

4.2 Predefining the H-UI vs. allowing for a generative UI

A basic mechanism we propose is the formalization of the LLM's output for a predefined H-UI. This allows for a classical GUI style with domain-specific apps offering well-established, pre-designed information visualization and interaction patterns. However, it limits the graphical output to the predefined GUI. To address this, we have proposed to reduce the UI to voice for these cases, seizing the flexibility of the informal natural language dialog. Another option would be to enable the LLM to freely combine information objects relevant for the user's intent and add UI controls to manipulate them or navigate to related content. This extension of our concept to a generative UI is an intriguing thought, yet shortcomings in usability common to adaptive user interfaces have to be considered [35]. A main remaining difficulty is likely user orientation in a changing layout. A potential approach could be to use a pre-designed GUI only for the main use cases and integrate graphical "sandboxes" which may contain a purpose driven, LLM-generated UI, which could also be semi-controlled via the AI-UI by using tools for its creation.

4.3 Specifying the AI-UI during design time vs. approaches for runtime

A second basic mechanism is the AI-UI, a purpose-built interface for interaction between the system and the LLM. Creating the AI-UI during design time again demands for pre-designing interaction, limiting the system to predictable use cases, at least for the GUI. In future work, it could be considered to automatically create the AI-UI from existing GUIs during runtime, provided the GUI code is accessible. Another approach would be to let the LLM use a GUI [14], as e.g. explored by Anthropic in the paradigm of computer-use [3]. This might be powerful for enabling the LLM to access a broad range of unforeseeable systems. However, it seems inefficient for intensive system use, as it requires transferring large amounts of data, such as images and potentially video, to the LLM. For predictable, frequent use cases with systems within the developer's

sphere of influence, a purpose-built interface, such as the AI-UI, is more efficient and controllable, as the LLM's actions are executed via specifically designed tools. Both approaches have their areas of application, and a combination might be most suitable in certain cases and should be explored as well.

In summary, pre-designing the AI-UI and H-UI can enhance controllability and efficiency for predictable use cases. However, to fully harness the generative potential of the LLM and accommodate a wider range of use cases, it is essential to explore the combination of pre-design with approaches that grant the LLM greater autonomy. This highlights the designer's responsibility to balance controllability with the LLM's capabilities.

5 Conclusion

UX designers need to address the shift from traditional UIs designed exclusively around human users to systems where the LLM can also initiate actions and work semi-autonomously on content. We contribute specific design considerations for combined graphical and voice interaction with LLMs in a *system as a partner* paradigm.

Understanding the conflicts outlined in this paper is crucial for designing this collaborative interaction to match UX goals like effectiveness, efficiency, and controllability. The design considerations offer practical suggestions for approaching the design process and for effectively shaping interaction between the human user and the LLM. With these insights, we aim to stimulate a discussion in the community of UX designers on why and how to address the complexities of integrating LLMs into combined voice and graphical systems, taking advantage of LLMs as partners while mitigating their shortcomings.

References

- [1] Mohammad M Alsuraishi and Dimitris I Rigas. 2008. Speech Displaces the Graphical Crowd. *J. Comput.* 3, 6 (2008), 47–58.
- [2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fournery, 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* (Glasgow, Scotland UK) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3290605.3300233
- [3] Anthropic. n.d.. Computer Use Tool. <https://docs.anthropic.com/en/docs/agents-and-tools/tool-use/computer-use-tool>. Accessed: June 2025.
- [4] Gagan Bansal, Jennifer Wortman Vaughan, Saleema Amershi, Eric Horvitz, Adam Fournery, Hussein Mozannar, Victor Dibia, and Daniel S Weld. 2024. Challenges in Human-Agent Communication. *arXiv preprint arXiv:2412.10380* (2024). doi:10.48550/arXiv.2412.10380
- [5] Ashish Bastola, Hao Wang, Judsen Hembree, Pooja Yadav, Zihao Gong, Emma Dixon, Abolfazl Razi, and Nathan McNeese. 2023. Llm-based smart reply (lsr): Enhancing collaborative performance with chatgpt-mediated smart reply system. *arXiv preprint arXiv:2306.11980* (2023). doi:10.48550/arXiv.2306.11980
- [6] Richard A Bolt. 1980. "Put-that-there" Voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. 262–270. doi:10.1145/965105.807503
- [7] Daniel Buschek, Lukas Mecke, Florian Lehmann, and Hai Dang. 2021. Nine potential pitfalls when designing human-ai co-creative systems. *arXiv preprint arXiv:2104.00358* (2021). doi:10.48550/arXiv.2104.00358
- [8] Philip R. Cohen, Michael Johnston, David McGee, Sharon Oviatt, Jay Pittman, Ira Smith, Liang Chen, and Josh Clow. 1997. QuickSet: multimodal interaction for distributed applications. In *Proceedings of the Fifth ACM International Conference on Multimedia* (Seattle, Washington, USA) (MULTIMEDIA '97). Association for Computing Machinery, New York, NY, USA, 31–40. doi:10.1145/266180.266328
- [9] Zijian Ding, Arvind Srinivasan, Stephen MacNeil, and Joel Chan. 2023. Fluid transformers and creative analogies: Exploring large language models' capacity for augmenting cross-domain analogical creativity. In *Proceedings of the 15th Conference on Creativity and Cognition*. 489–505. doi:10.1145/3591196.3593516
- [10] Daniel Nygaard Ege, Mark Goudswaard, James Gopsill, Martin Steinert, and Ben Hicks. 2024. What, how and when should I prototype? An empirical study of design team prototyping practices at the IDEA challenge hackathon. *Design Science* 10 (2024), e22. doi:10.1017/dsj.2024.16
- [11] Toru Furusawa and Masatoshi Saitoh. 2024. A Demonstration of Voice-Interactive AI Agents for Vehicles Utilizing Multiple LLMs. In *2024 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 249–251. doi:10.1109/SMARTCOMP61445.2024.00058
- [12] Katrin Glinka and Claudia Müller-Birn. 2023. Critical-Reflective Human-AI Collaboration: Exploring Computational Tools for Art Historical Image Retrieval. *Proc. ACM Hum.-Comput. Interact.* 7, CSCW2, Article 263 (Oct. 2023), 33 pages. doi:10.1145/3610054
- [13] Catalina Gomez, Sue Min Cho, Shichang Ke, Chien-Ming Huang, and Mathias Unberath. 2025. Human-AI collaboration is not very collaborative yet: A taxonomy of interaction patterns in AI-assisted decision making from a systematic review. *Frontiers in Computer Science* 6 (2025), 1521066. doi:10.3389/fcomp.2024.1521066
- [14] Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2024. Navigating the digital world as humans do: Universal visual grounding for gui agents. *arXiv preprint arXiv:2410.05243* (2024). doi:10.48550/arXiv.2410.05243
- [15] Barbara J Grosz. 1996. Collaborative systems (AAAI-94 presidential address). *AI magazine* 17, 2 (1996), 67–67.
- [16] Joe Ee Ho, Boon Yaik Ooi, and Markus Westner. 2024. Application Integration Framework for Large Language Models. In *2024 5th International Conference on Artificial Intelligence and Data Sciences (AiDAS)*. IEEE, 1–6. doi:10.1109/AiDAS63860.2024.10730541
- [17] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 159–166. doi:10.1145/302979.303030
- [18] Hu Jin, Shuo Wang, Nian-Wei Chen, and Zhen Ye. 2009. Finite state machine for automatic GUI testing. In *2009 International Conference on Computational Intelligence and Software Engineering*. IEEE, 1–4. doi:10.1109/CISE.2009.5364314
- [19] Jan-Frederik Kassel and Michael Rohs. 2018. Valletto: A Multimodal Interface for Ubiquitous Visual Analytics. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI EA '18). Association for Computing Machinery, New York, NY, USA, 1–6. doi:10.1145/3170427.3188445
- [20] Emine Merve Kaya and Mounya Elhilali. 2014. Investigating bottom-up auditory attention. *Frontiers in human neuroscience* 8 (2014), 327. doi:10.3389/fnhum.2014.00327
- [21] Stefan Kopp and Nicole Krämer. 2021. Revisiting human-agent communication: The importance of joint co-construction and understanding mental states. *Frontiers in Psychology* 12 (2021), 580955. doi:10.3389/fpsyg.2021.580955
- [22] Gierad P. Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder, and Eytan Adar. 2013. PixelTone: a multimodal interface for image editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) (CHI '13). Association for Computing Machinery, New York, NY, USA, 2185–2194. doi:10.1145/2470654.2481301
- [23] Konstantin Läufer. 1995. Interactive Web Applications Based on Finite State Machines. In *Invited paper/Proc. Symp. Information Systems Analysis and Synthesis (ISAS), Baden-Baden, Germany*. Citeseer.
- [24] Jaewook Lee, Sebastian S. Rodriguez, Raahul Natarajan, Jacqueline Chen, Harsh Deep, and Alex Kirlik. 2021. What's This? A Voice and Touch Multimodal Approach for Ambiguity Resolution in Voice Assistants. In *Proceedings of the 2021 International Conference on Multimodal Interaction* (Montréal, QC, Canada) (ICMI '21). Association for Computing Machinery, New York, NY, USA, 512–520. doi:10.1145/3462244.3479902
- [25] MAIKEL Leon. 2024. The escalating AI's energy demands and the imperative need for sustainable solutions. *WSEAS Transactions on Systems* 23 (2024), 444–457.
- [26] Jan Leusmann, Chao Wang, and Sven Mayer. 2024. Comparing Rule-based and LLM-based Methods to Enable Active Robot Assistant Conversations. *Proceedings of the CUI@ CHI 2024: Building Trust in CUIs-From Design to Deployment* (2024), 05–11. <https://cui.acm.org/workshops/CHI2024/wp-content/uploads/2024/04/Comparing-Rule-based-and-LLM-based-Methods-to-Enable-Active-Robot-Assistant-Conversations.pdf>
- [27] Amama Mahmood, Junxiang Wang, Bingsheng Yao, Dakuo Wang, and Chien-Ming Huang. 2025. User interaction patterns and breakdowns in conversing with LLM-powered voice assistants. *International Journal of Human-Computer Studies* 195 (2025), 103406. doi:10.1016/j.ijhcs.2024.103406
- [28] Bahar Mahmud, Guan Hong, and Bernard Fong. 2023. A Study of Human-AI Symbiosis for Creative Work: Recent Developments and Future Directions in Deep Learning. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 2, Article 47 (Sept. 2023), 21 pages. doi:10.1145/3542698
- [29] Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, et al. 2025. A Survey of Context Engineering for Large Language Models. *arXiv preprint arXiv:2507.13334* (2025). doi:10.48550/arXiv.2507.13334
- [30] J. G. Neal, C. Y. Thielman, Z. Dobes, S. M. Haller, and S. C. Shapiro. 1989. Natural language with integrated deictic and graphic gestures. In *Proceedings of the Workshop on Speech and Natural Language* (Cape Cod, Massachusetts) (HLT '89).

- Association for Computational Linguistics, USA, 410–423. doi:10.3115/1075434.1075499
- [31] Americo Talarico Neto, Renata Pontin M. Fortes, and Adalberto G. da Silva Filho. 2008. Multimodal interfaces design issues: the fusion of well-designed voice and graphical user interfaces. In *Proceedings of the 26th Annual ACM International Conference on Design of Communication* (Lisbon, Portugal) (SIGDOC '08). Association for Computing Machinery, New York, NY, USA, 277–278. doi:10.1145/1456536.1456597
- [32] Jakob Nielsen. 2023. AI: First New UI Paradigm in 60 Years. <https://www.nngroup.com/articles/ai-paradigm/> Accessed: March 2025.
- [33] Donald A. Norman. 2010. Natural user interfaces are not natural. *Interactions* 17, 3 (May 2010), 6–10. doi:10.1145/1744161.1744163
- [34] Shuyin Ouyang, Jie M Zhang, Mark Harman, and Meng Wang. 2025. An empirical study of the non-determinism of chatgpt in code generation. *ACM Transactions on Software Engineering and Methodology* 34, 2 (2025), 1–28. doi:10.1145/3697010
- [35] Tim F. Paymans, Jasper Lindenberg, and Mark Neerincx. 2004. Usability trade-offs for adaptive user interfaces: ease of use and learnability. In *Proceedings of the 9th International Conference on Intelligent User Interfaces* (Funchal, Madeira, Portugal) (IUI '04). Association for Computing Machinery, New York, NY, USA, 301–303. doi:10.1145/964442.964512
- [36] Aung Pyae. 2025. The Human-AI Handshake Framework: A Bidirectional Approach to Human-AI Collaboration. *arXiv preprint arXiv:2502.01493* (2025). doi:10.48550/arXiv.2502.01493
- [37] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems* 36 (2023), 68539–68551. doi:10.48550/arXiv.2302.04761
- [38] Ben Shneiderman. 2020. Human-centered artificial intelligence: Three fresh ideas. *AIIS Transactions on Human-Computer Interaction* 12, 3 (2020), 109–124. doi:10.17705/1thci.00131
- [39] Lukas Stappen, Jeremy Dillmann, Serena Striegel, Hans-Jörg Vögel, Nicolas Flores-Herr, and Björn W Schuller. 2023. Integrating generative artificial intelligence in intelligent vehicle systems. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 5790–5797. doi:10.1109/ITSC57777.2023.10422003
- [40] S Subhash, S Siddesh, N Srivatsa Prajwal, A Ullas, and B Santhosh. 2021. Developing a Graphical User Interface for an Artificial Intelligence-Based Voice Assistant. *International Journal of Organizational and Collective Intelligence (IJOICI)* 11, 3 (2021), 49–67. doi:10.4018/IJOICI.2021070104
- [41] Loren G Terveen. 1995. Overview of human-computer collaboration. *Knowledge-Based Systems* 8, 2-3 (1995), 67–81. doi:10.1016/0950-7051(95)98369-H
- [42] Aliaksei Vertsel and Mikhail Rumiantsev. 2024. Hybrid llm/rule-based approaches to business insights generation from structured data. *arXiv preprint arXiv:2404.15604* (2024). doi:10.48550/arXiv.2404.15604
- [43] Michael Vössing, Niklas Kühl, Matteo Lind, and Gerhard Satzger. 2022. Designing Transparency for Effective Human-AI Collaboration. *Information Systems Frontiers* 24, 3 (June 2022), 877–895. doi:10.1007/s10796-022-10284-3
- [44] Tianshu Wang, Ling Luo, and Xin Zhang. 2021. Evaluation Model of Voice User Interface. In *Advances in Usability, User Experience, Wearable and Assistive Technology: Proceedings of the AHFE 2021 Virtual Conferences on Usability and User Experience, Human Factors and Wearable Technologies, Human Factors in Virtual Environments and Game Design, and Human Factors and Assistive Technology, July 25-29, 2021, USA*. Springer, 894–899. doi:10.1007/978-3-030-80091-8_106
- [45] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. *ACM Transactions on Knowledge Discovery from Data* 18 (2023), 1–32. doi:10.1145/3649506
- [46] John Zimmerman, Jodi Forlizzi, and Shelley Evenson. 2007. Research through design as a method for interaction design research in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '07). Association for Computing Machinery, New York, NY, USA, 493–502. doi:10.1145/1240624.1240704
- [47] Victor W Zue and James R Glass. 2000. Conversational interfaces: Advances and challenges. *Proc. IEEE* 88, 8 (2000), 1166–1180. doi:10.1109/5.880078
- [48] Selin İnel, Öğuz Turan Buruk, Mehmet Cengiz Onbaşlı, and Öğuzhan Özcan. 2018. Snowflakes: A Design Speculation for a Modular Prototyping Tool for Rapidly Designing Smart Wearables. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI EA '18). Association for Computing Machinery, New York, NY, USA, 1–6. doi:10.1145/3170427.3188676