Voice-based Direct Manipulation to Foster Inclusion in Intent-driven User Interfaces

Laura Colazzo, Emanuele Pucci and Maristella Matera

Politecnico di Milano, Department of Electronics, Information and Bioengineering, Milano, Italy

Abstract

This paper discusses voice-based direct manipulation to extend the conversational interactions with LLMs and, more generally, with intent-driven user interfaces. Inspired by recent work on visual direct manipulation in AI-assisted tools, we explore whether similar patterns can enhance Voice User Interfaces (VUIs). Such approaches can improve usability of intent formulation, especially in contexts where voice is the primary or the only interaction modality, with important implications for accessibility and social inclusion.

Keywords

Intent-driven UIs, Voice-based direct manipulation, LLM interfaces, Accessibility

1. Introduction

Recent advancements in Generative AI and its widespread availability are impacting professional and private lives. This new technology has also contributed to a revolutionary shift in the way humans interact with technology: a paradigm Nielsen described as "intent-based outcome specification" [1]. This is fundamentally different from other paradigms that have emerged in the history of Human-Computer Interaction (HCI). Indeed, compared to command-based interactions, the intent-based paradigm shifts the control over how the computation is performed from the user to the underlying AI model [1].

After the ChatGPT launch, back in November 2022, conversational intent-based interactions have quickly become the standard to interact with Large Language Models (LLMs). In this paradigm, humans and AI engage in multi-turn conversations, with the user focusing on expressing the desired outcome in the form of a prompt in natural language, while the model is responsible for capturing the user intent and converting it into a meaningful result [2]. Despite the exceptional LLMs popularity and the unprecedented opportunities they have unlocked in many fields, ensuring user intent is effectively and accurately captured by the LLM, starting from a prompt expressed in natural language, still poses significant usability challenges. Prompt-engineering techniques, such as few-shot prompting and chain-of-thought, have emerged as strategies to improve the alignment between the output produced by the model and the user intent. However, the effectiveness of such techniques is inherently limited [3]. Additionally, when situational or permanent disability demands voice interaction, prompt refinement may feature barriers due to the lack of adequate interactions and content manipulation mechanisms [4].

Alongside text-based prompting, other techniques that leverage the manipulation of visual elements to streamline prompting and achieve more direct interactions are emerging. This paper discusses these new strategies and introduces possible ways to bring voice-based direct manipulation into intent-driven interfaces. Shifting the focus to the voice modality is fundamental to ensure advancements in AI remain human-centered, preventing social exclusions and encouraging participation while embracing diversity. After illustrating the current panorama of conversational intent-based interaction, especially applied to the interaction with LLMs, the paper discusses how typical patterns for direct manipulation in visual intent-based interfaces can be translated into voice-based interaction mechanisms. The paper also discusses a preliminary prototype informed by a user study that involved Blind and Visually Impaired (BVI) participants.

Joint Proceedings of IS-EUD 2025: 10th International Symposium on End-User Development, 16-18 June 2025, Munich, Germany

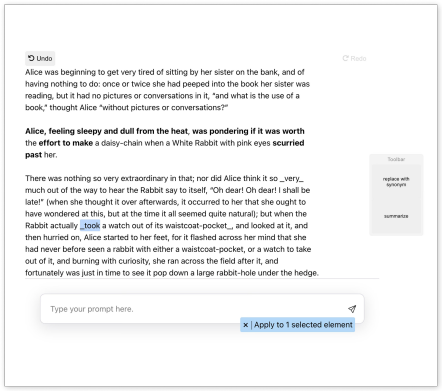


Figure 1: DirectGPT's UI [3]

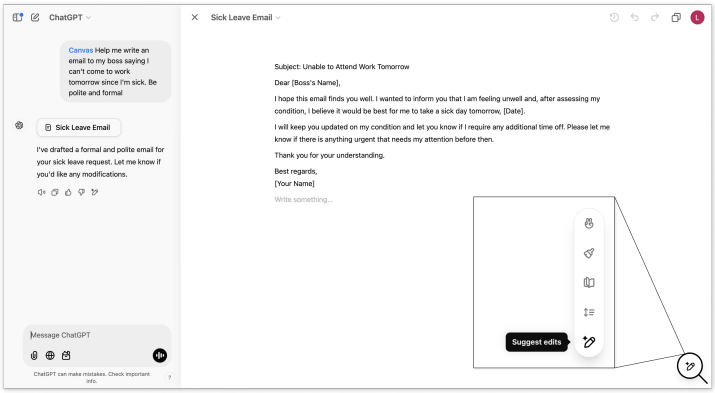


Figure 2: Canvas' UI [6]

2. Beyond Text-based Prompting in Human-LLM Interaction

Although standard prompting, especially text-based prompting, is currently the most common interaction mechanism in Generative AI systems [5], novel approaches are emerging. According to the taxonomy of the most common patterns in Generative AI UIs proposed by Luera and et al. [5], prompting, whether it is text-based, visual, audio or multimodal, is the most prominent example of what the authors define "user-guided interactions" in the landscape of Generative AI. In addition to chat-based interfaces, other categories feature a *canvas area* at the center of the screen where the majority of interactions occur. More specifically, *information visualization canvases* represent a subcategory where elements inside the central area can be directly manipulated to alter the state of the system.

DirectGPT [3] (Figure 1) is an example of a system featuring an information visualization canvas interface [5], implemented on top of an LLM. In DirectGPT, users are allowed to build prompts by manipulating visual components inside the UI, using mechanisms such as multi-modal prompting and multi-selection [5]. The authors have demonstrated how direct manipulation principles can be effectively combined with traditional text-based prompting strategies to streamline the interaction with LLMs and make it more direct, while overcoming some of the limitations of purely conversational UIs. Indeed, as the authors point out, while it is true that traditional conversational LLM interfaces enable users to create content like text, code, or images, it may take several conversational turns to achieve a satisfactory result, thus slowing down interactions. Furthermore, error-recovering strategies are not supported and unsatisfactory results negatively affect subsequent interactions. For this iterative adjustment process to be truly effective, then, users must be able to reference specific elements from the model's previous response, something purely-conversational interfaces do not allow.

DirectGPT, as opposed to conversational interfaces, builds upon the idea that to enable direct manipulation the objects of interest must be continuously accessible inside the interface. For this reason, the long history of messages is replaced by a fixed area at the center of the screen, where the model output is continuously displayed after each modification. Furthermore, the dynamic population of a toolbar of commands makes available the most recent user prompts in the form of buttons, fostering their re-usability. Increased directness is also achieved through physical actions, such as dragging or highlighting, augmenting the expressivity and the flexibility of purely textual prompts. Additionally, error recovery is achieved through undo and redo commands, displayed as buttons located at the top of the interface. The system also highlights the elements affected by the LLM-generated modifications, allowing users to quickly assess the effect of their prompts and immediately revert them if needed.

Content creation is one of the primary applications of Generative AI. As Luera and et al. [5] point out in their survey, the most common UI layouts for AI-assisted content creation tasks, involving the generation or editing of visual, written or audio content, are conversational and canvas UIs, with the latter being more frequently adopted for the generation of visual content. However, this pattern is not absolute and canvas UIs have also been developed for AI-assisted writing tasks. An example is

given by Canvas [6] (Figure 2): a tool integrated in ChatGPT, introduced by OpenAI in October 2024. Canvas is designed to support users in their writing and coding projects, with a UI that goes beyond the traditional chat layout. Indeed, as emphasized in the article about its launch, working with a chat can be limiting for tasks that require editing and revisions. The tool, instead, introduces a canvas area, separate from the chat, that ChatGPT users can leverage to directly engage with the text or code to be edited. Supported actions include the possibility to highlight specific portions of the text to restrict the context of the action to be performed; the chance for the user to directly edit the text or code, for those cases that do not require the support of the underlying LLM; the access to a menu of shortcuts for quickly tuning parameters such as the length or the reading level of the text; the option to perform undo and redo actions through dedicated buttons; the opportunity to check the difference between the current version of the text and the previous one. These features facilitate the collaboration between the user and ChatGPT, with greater flexibility compared to standard interfaces used for the same tasks.

3. Voice-based Direct Manipulation

Our research explores how *direct manipulation principles*, most commonly associated with GUIs, here can be extended to voice LLMs' interfaces. The direct-manipulation paradigm was first introduced to describe interactive systems that apply three main principles [7]: *(i)* continuous representation of the object of interest; *(ii)* physical actions or labeled button presses instead of complex syntax; *(iii)* rapid, incremental, reversible operations whose impact on the object of interest is immediately visible. Our research aims to identify how these principles can inspire the definition of new mechanisms to achieve more expressive and direct vocal interactions in the context of LLM-powered voice-based UIs (VUIs). The research is informed by a user study we performed between November 2023 and July 2024 [4] to explore the accessibility potential and challenges of voice interaction when accessing LLMs. Our research began with an exploratory interview with a BVI expert in assistive technologies, which provided firsthand insights into the needs of BVI users interacting with LLMs. The conversation focused on the use of tools such as ChatGPT for information access. Building on the interview findings, we first designed and distributed an online questionnaire targeting BVI individuals that received 116 responses; we then conducted individual semi-structured interviews. A thematic analysis highlighted the main aspects emerged from the gathered data. Related voice interaction patterns were identified as solutions to the main challenges, and validated in focus groups with 11 participants.

Overall, the study revealed a range of design opportunities in voice interaction. It also emphasized the need for better ways to manage and organize conversations, making it easier for users to navigate content, revisit important information, and maintain context across sessions. Additionally, the findings highlighted the importance of supporting more effective consumption of generated responses, by enabling smoother navigation through long or complex outputs and direct manipulation of specific output portions. These design insights informed the development of a prototype incorporating the new voice interaction patterns. We are now building on this experience to extend *direct manipulation* principles to the vocal channel. The discussion presented below is inspired by the previously-discussed examples of canvas UIs and considers some well-known challenges of VUI design from the literature.

Continuous representation of the object of interest. The system could let the user hear the most updated version of the output—the LLM-generated text—aloud. In this way, the effect of user-requested commands could be immediately perceived, without the need for the user to explicitly ask to have the full text read out loud any time a new transformation is applied. However, while this mechanism could be effective when the model's response is relatively short, for more lengthy outputs, it could decrease usability. In such circumstances, providing a summary of the applied changes might be more appropriate, while still preserving the chance for the user to listen to the full modified text upon request.

Physical actions or labeled button presses instead of complex syntax. While the definition of visual abstractions acting like shortcuts is relatively easy and effective in GUIs, in voice interfaces it requires careful consideration. Such an abstraction could be implemented as pre-defined vocal commands that act as proxies for more complex and longer prompts. A challenge is, however, to inform

the users about the availability of such shortcuts; once aware, they should also be able to recall them. To address these issues, the system could actively remind of the availability of such commands, asking at strategic points in the conversation whether the user wants to hear the list of such shortcuts. A less intrusive alternative would be to provide it only upon request. In both cases, to help the user remember all the proposed options, no more than three elements should be provided at a time [8]. Overall, the pattern appears to be consistent with the findings of previous work [9]. Additionally, identifying a voice-based version of gestures that in GUIs are naturally associated with a semantic meaning, such as a selection mechanism to restrict the context of a given prompt to a smaller portion of text, is far more challenging. It requires the ability to physically point to a specific element of text, an action that does not have a direct equivalent in the voice domain. However, equipping the user with a fine-grained navigation mechanism, through which the text can be linearly explored and potentially modified along the way, may give users an analogous degree of expressiveness in crafting prompts through speech. By considering the limitations of human short-term memory, this pattern might give users the chance to focus on smaller and more manageable portions of text for an easier editing and review experience. One may argue that having speech as primary means for expressing intent inevitably introduces a certain degree of indirectness, forcing the user to precisely describe in words what the expected outcome is and which portions of the previous model's output the prompt should affect. With this navigation mechanism, instead, users could gain finer-grained control over the generation process, having access to an in-place tool for a more precise and accurate definition of intents that removes the need for extensive descriptions. However, to prevent users from feeling lost and help them form a mental model of the text being explored, this method should be combined with a suitable strategy that promotes *location awareness*, such as explicitly assigning each navigation node a progressive numerical identifier.

Rapid, incremental, reversible operations whose impact on the object of interest is immediately visible. While reversibility can be easily achieved by introducing vocal commands that undo or redo recent modifications, the possibility of performing incremental operations quickly is the most critical part of the interaction. This is because vocal commands can be relatively rapid to issue, but the efficiency with which the user is able to evaluate their effect on the system strongly depends on the way such actions are acknowledged. When the target text can be visually inspected, it is easier for the user to check the difference between the current version of the output and the previous one, especially if the interface provides a dedicated mechanism to inspect and browse the editing history, like the one available in Canvas. In VUIs, instead, the efficiency with which users can evaluate the effect of issued commands depends on the trade-offs discussed for the first principle.

4. Preliminary Prototype

To assess the potential of the patterns presented in the previous section, we integrated them into a new paradigm for vocal interaction with LLMs [4]. For this purpose, we considered a use case in which the user's goal is to iteratively refine a message received from the model within an ongoing chat (e.g. the body of an LLM-generated email to adjust its tone and structure). A sketch of the interaction flow is provided in Figure 3. Being our focus on voice interfaces, visual elements in the figure are used to depict the objects of the vocal interactions.

At first, the user chooses a message from a chat that will act as a target for editing actions. To support the refinement process, an editing mode is introduced. This is conceptually analogous to the canvas area seen in some visual LLM interfaces and can be accessed using a dedicated vocal command, such as "Enter edit mode". Being such a space virtually separated from the main chat with the model, it can be leveraged by the user to iteratively carry out a series of LLM-assisted transformations to the corpus of the text, while keeping the main chat clean and concise. In fact, in VUIs, the longer the conversation history, the more frustrating it becomes to fully traverse it [4]. Moreover, if some messages in the chat only represent transient intermediate results, they can contribute to a more cluttered history of messages, which might be even more tedious to navigate. Given that in edit mode the exchange of messages between the user and the model only serves the purpose of producing a new, satisfactory

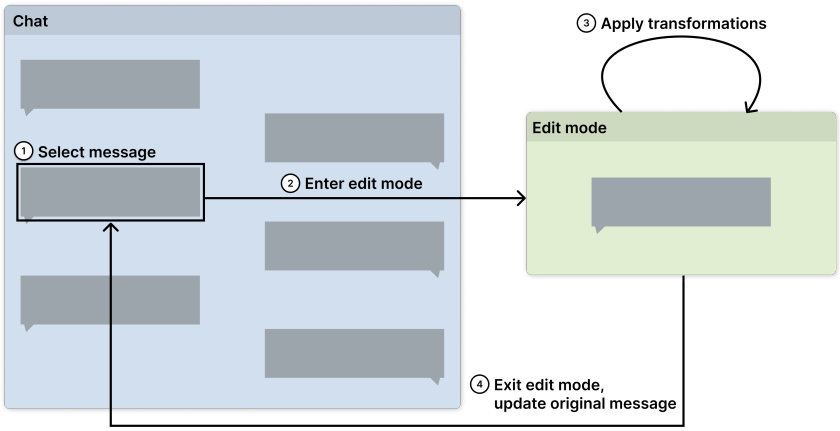
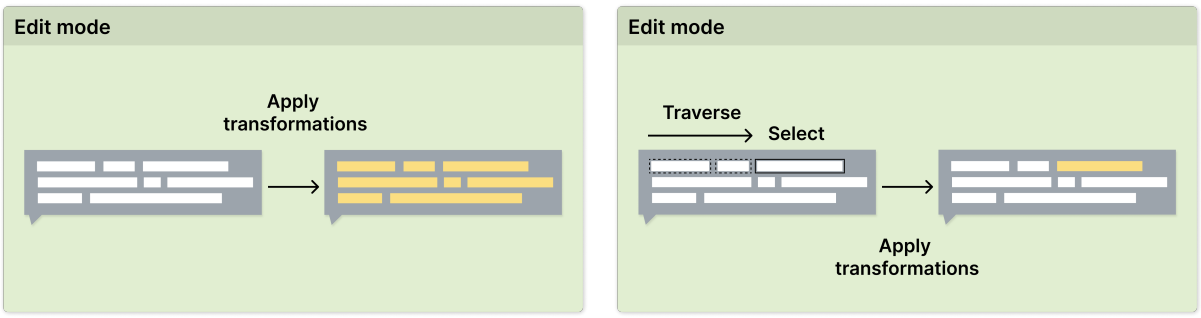


Figure 3: Interaction flow



(a) Global editing actions

(b) Localized editing actions

Figure 4: Editing actions

version of the message selected as the target for modification, only the final output of such a procedure persists to the main chat once the user is out of edit mode.

In edit mode, we can classify user requests to modify text into two categories:

- *Global editing actions* (Figure 4a): they affect the target message in its entirety. Examples include requests to summarize the message or change its tone. Any time a request of this kind is sent to the model, the full modified message is read aloud. In the case of particularly long messages, instead, only a summary of modifications applied is provided.
- *Localized editing actions* (Figure 4b): they affect only a restricted portion of the text, such as a single sentence or word. Examples include rewriting a given sentence using a different style or replacing a word with a synonym. To perform this class of actions the user must be able to both traverse the message using a finer-grained and adjustable granularity (e.g. at sentence level), and select the specific portion to be transformed. This behaviour can be achieved through the previously-discussed navigation pattern, implemented as a navigation mode that can be activated using a vocal command like “Enter navigation mode”. Users can optionally specify the desired navigation granularity, otherwise the most appropriate one is selected based on the length of the message. In navigation mode, users can also access helper commands that facilitate message traversal. Landing on a given node automatically triggers a series of actions. At first, the node’s identifier is read aloud followed by its content. Then, the user receives suggestions from the system for possible actions to take. At this stage, any request involving an editing action automatically considers the node currently in focus as the target of the transformation to be applied. Moreover, while traversing the message, any time a request to modify a node is issued, the updated version of its content is read aloud.

Additional complementary strategies to improve the overall interaction include:

- *Auditory cues*: sounds can be used to reinforce and complement verbal feedback about specific actions. E.g., a specific non-verbal sound can be reproduced after the message "You are now in edit mode", when entering this mode. Such an additional layer of feedback could help increase the users' awareness of the system's state and guide them more intuitively through the interaction.
- *Keyboard shortcuts*: they can be introduced as an alternative to commands that already have a voice-based counterpart. For instance, inside navigation mode, users could use the right arrow key to move to the next node instead of pronouncing the word "Next". The availability of such keyboard shortcuts could streamline the interaction. However, to preserve the hands-free modality, these mechanisms should only represent a form of redundancy.
- *Help command*: to facilitate the exploration of all actions and shortcuts available from a given stage of the interaction, a help command can additionally be introduced.

5. Conclusion

This paper has illustrated preliminary insights on defining vocal interaction patterns for direct manipulation in chat-based interfaces. Sketching the interaction, as illustrated in the previous section, represents the first step toward validating the feasibility of the proposed patterns. The next steps involve testing the interaction with users. A fast-prototyping approach was envisioned for this purpose, leveraging OpenAI's custom GPTs as tools for demonstrating the identified interaction patterns. This approach presents several advantages, including access to a pre-built interface with a ready-to-use voice mode available on top of OpenAI's models. The configuration required to share and run the prototype is thus limited to the definition of a well-formatted system prompt. Despite its limitations, the low resource demand of this solution makes it promising for quickly testing and improving the interaction. However, more sophisticated prototyping techniques will also be explored before approaching the validation step, with the aim of defining a more robust artifact that can lead to a higher quality of the gathered data.

Declaration on Generative AI

The authors used Writefull for grammar and spelling checking. However, the authors extensively reviewed and edited the text; therefore, they take full responsibility for the publication's content.

References

[1] J. Nielsen, AI: First New UI Paradigm in 60 Years, 2023. URL: https://www.nngroup.com/articles/ai-paradigm/.

[2] J. Gao, S. A. Gebreegziabher, K. T. W. Choo, T. J.-J. Li, S. T. Perrault, T. W. Malone, A Taxonomy for Human-LLM Interaction Modes: An Initial Exploration, in: CHI EA '24', ACM, 2024.

[3] D. Masson, S. Malacria, G. Casiez, D. Vogel, DirectGPT: A Direct Manipulation Interface to Interact with Large Language Models, in: Proc. of CHI '24, ACM, 2024.

[4] E. Pucci, L. Piro, S. Andolina, M. Matera, From Conversational Web to Inclusive Conversations with LLMs, in: C. Conati, G. Volpe, I. Torre (Eds.), Proc. of AVI 2024, ACM, 2024, pp. 87:1–87:3.

[5] R. Luera, R. A. R. et al., Survey of User Interface Design and Interaction Techniques in Generative AI Applications, 2024.

[6] OpenAI, Introducing Canvas, 2024. URL: https://openai.com/index/introducing-canvas/.

[7] Shneiderman, Direct Manipulation: A Step Beyond Programming Languages, Computer 16 (1983).

[8] Z. Wei, J. A. Landay, Evaluating Speech-Based Smart Devices Using New Usability Heuristics, IEEE Pervasive Computing 17 (2018) 84–96.

[9] W. Y. Luebs, G. W. Tigwell, K. Shinohara, Understanding Expert Crafting Practices of Blind and Low Vision Creatives, in: CHI EA '24', ACM, 2024.