**Multimodal IVR Report**

**Abstract**

Traditional voice-only IVR systems often suffer from high cognitive load and limited flexibility, affecting usability and user satisfaction. This project presents a multimodal IVR system that combines speech and graphical interaction to support more natural and efficient user experiences. The system integrates Whisper for speech recognition, an ollama llm for dialogue management, Octave for text-to-speech, and a web-based GUI built with Flask, SocketIO, and standard web technologies. By enabling users to interact through both voice and visuals, the system aims to reduce memory burden and streamline IVR navigation. A user evaluation will compare the multimodal prototype with a baseline voice-only system to assess performance, satisfaction, and efficiency. The project demonstrates how blending voice with visual modalities can enhance IVR usability and accessibility.

**1. Introduction**

Interactive Voice Response (IVR) systems are widely used in customer service but often frustrate users due to rigid structures, deep menu hierarchies, and memory-heavy spoken prompts. Prior research highlights how cognitive overload in such systems stems from the need to recall lengthy spoken options (Bigot et al., 2013, as cited in Baeza & Kumar, 2019). Studies in human–computer interaction, particularly Oviatt (1999), demonstrate that combining voice with visual output can reduce errors, ease cognitive load, and improve user satisfaction by supporting flexible interaction styles.

This project explores whether a multimodal IVR—one combining voice and graphical input/output—can improve usability compared to a traditional voice-only system. Users interact via speech while receiving dynamic visual feedback on a web interface, with the option to respond either verbally or by selecting on-screen. This design aims to reduce frustration, support diverse user needs, and improve task efficiency.

The system integrates Whisper for speech recognition, an ollama LLM for natural language understanding and dialogue management, Octave for text-to-speech, and a real-time web interface using Flask and Socket.IO. These components work together to provide voice and visual interaction simultaneosly.

The project addresses three key questions: whether graphical support improves efficiency and satisfaction, whether multimodal interaction reduces cognitive load and error rates, and what challenges arise in integrating these components in real time. A prototype and user evaluation will compare the multimodal system to a voice-only baseline, using task performance, error rates, and user feedback to assess impact. By grounding its design in existing research and leveraging current technologies, this project aims to demonstrate that multimodal IVRs can significantly improve the usability and accessibility of automated voice services.

**2. Literature Review**

**2.1 Evolution of IVR Systems and Interfaces**

IVR systems have evolved significantly since the 1980s, transitioning from DTMF-based input and pre-recorded voice prompts to more flexible, speech-driven interfaces. Even in the mid-1990s, efforts were underway to enhance IVR capabilities through multimodality. Damhuis et al. (1994) described a consumer information server that combined IVR with fax and email outputs. While interaction was primarily through a DTMF-based IVR menu, the system allowed users to receive detailed information—such as (Karolina Gabor-Siatkowska, 20 August 2024) (Martin Damhuis, September 1994) (Luis Bravo) (Kadam, 2022) (Khushbu Mehboob Shaikh, 2024) (Vlasios Kasapakis, 2022) (David Griol, 2016) (Elias Dritsas, 2025) (Rianna R. Baeza, 2019) (Tom Bocklisch, 2017) (Shruti Bhargava, 2023) (Fernando Batista, 2016) (Oviatt, 1999) (Runting Zhong, 2022) (Prashan Wanigasekara, 2023)brochures or test results—via fax, reducing the cognitive load of lengthy voice prompts. The architecture was designed to be extensible, with planned integration of speech recognition and other advanced features. This early system illustrates how multimodal design can address the limitations of voice-only IVR by leveraging complementary output channels.

By the late 1990s and 2000s, IVRs began incorporating speech recognition, moving from keypad input to voice-driven interfaces. Technologies like VoiceXML, introduced around 2000, enabled voice input/output and declarative dialogue flows—but these systems remained largely unimodal. Griol and Molina (2016) describe how mobile and web technologies expanded IVRs into multimodal interfaces, integrating voice with touch and visual feedback via Android APIs. Their work bridges academic and industrial approaches, demonstrating how voice services can be deployed on smartphones with both speech and screen-based interactions. One example is a municipal information app that allows users to query services via speech while receiving results visually, illustrating the shift from telephony-based IVRs to multimodal mobile applications.

IVR development has evolved from code-heavy systems requiring telephony expertise to more accessible, automated approaches. Initially, IVRs were built with hard-coded logic, making updates laborious and error-prone (Shaikh & Giannakopoulos, 2024). The introduction of widget-based design tools allowed non-programmers to visually construct call flows using drag-and-drop interfaces, reducing development time and increasing adaptability. Today, AI-powered IVRs integrate natural language processing and machine learning to support open-ended, conversational input and automated flow generation. These systems improve efficiency and user experience by enabling more natural interactions and reducing the need for rigid menu structures. As Shaikh & Giannakopoulos (2024) argue, AI transforms IVRs by enhancing both usability and development workflows. Our project aligns with this shift, leveraging advanced ASR and NLU tools (e.g., Rasa or LLMs) to build flexible, responsive voice interfaces.

Modern voice assistants like Alexa, Google Assistant, and Siri can be seen as natural successors to IVR systems, enabling voice-based access to services with added multimodal capabilities—especially on smartphones and smart displays. Unlike traditional phone IVRs, these assistants support visual feedback, such as cards or lists, creating richer interactions. Their widespread use has familiarized users with voice interaction and increased expectations for natural language understanding. This shift supports our hypothesis that users will find a multimodal IVR interface both intuitive and effective. Our system builds on this evolution by combining speech with a GUI to improve usability.

**2.2 Principles and Benefits of Multimodal Interaction**

Multimodal interaction combines input and output modes like speech, text, touch, gesture, and gaze to create more natural and expressive interfaces. Oviatt (1999) dispels common myths about multimodal systems, emphasizing their potential to improve robustness and usability—not as a gimmick, but by aligning with how humans naturally communicate. For instance, Myth #1 highlights that users don’t always use all input modes at once; they choose based on context and convenience. Often, modalities are used sequentially rather than simultaneously, which supports designing flexible systems that adapt to user behavior. In IVR contexts, this means allowing users to alternate between speaking and tapping on a screen, rather than enforcing simultaneous use. Our system follows this guidance by enabling fluid switching between modalities to enhance user experience.

Multimodal systems benefit from combining complementary input modes. A classic example is Bolt’s (1980) “put-that-there” system, where speech and pointing worked together to specify both action and target. Similarly, Cohen et al.’s QuickSet (1997) allowed users to issue military simulation commands using speech and pen gestures, enabling spatial tasks that would be cumbersome using a single modality. QuickSet used semantic fusion to integrate speech and gesture inputs into a unified command, exemplifying what Oviatt (1999) calls a “highly synergistic blend.” Beyond efficiency gains (~10% faster than speech-only interaction), multimodal input also improved accuracy through mutual disambiguation: ambiguous speech could be clarified by pen input, and vice versa. This directly challenges Oviatt’s Myth #7, showing that carefully integrated recognition systems can reduce—not compound—errors.

In a voice-plus-GUI IVR, the visual modality can complement speech by delivering information that’s difficult to convey audibly—such as long account numbers. For example, a spoken summary (“Your balances are as follows”) can be paired with on-screen figures, allowing users to hear the gist and read precise details. This multimodal approach supports user preference and context, offering both convenience and clarity. It also adds redundancy: users can rely on one modality if they miss or misinterpret the other. As Dritsas et al. (2025) note, redundancy and complementarity between modalities enhance robustness, reduce ambiguity, and mirror natural human communication.

Another benefit of multimodal interfaces is improved user engagement and satisfaction. When users have control over how they input information or receive feedback, they can choose the path of least effort. For example, when speech recognition fails—common with proper nouns—users can switch to typing or selecting on-screen suggestions. Kadam (2022) highlights such flexibility as essential for usability and outlines challenges like keeping voice and visual components synchronized. She recommends visual cues (e.g., “listening” indicators or partial transcriptions) to reflect system state and help users stay oriented. Our IVR system applies this guidance by showing live speech transcriptions and feedback indicators on the GUI, improving transparency and user control.

It’s important to clarify what multimodal interaction is not. Oviatt (1999) debunks the assumption (Myth #4) that speech is always the primary mode in multimodal systems. In reality, users may prioritize visual input depending on context—for example, using touch in a quiet office and voice while driving. Similarly, Myth #2 assumes that speech + pointing is the dominant modality pair, but studies show that modality preference varies by task. Our IVR system reflects these insights by supporting flexible interaction flows: users may read first then speak, or hear a prompt and tap. Designing for such variability enhances usability and aligns with how people naturally communicate.

The literature strongly supports a multimodal approach to IVR, highlighting benefits like increased robustness and user-friendliness. Multimodal systems give users greater expressive power by allowing them to interact in the most natural way for the moment (Oviatt, 1999). For designers, this means supporting input fusion (e.g., combining speech and GUI input) and output synchronization (e.g., aligning voice and visual feedback). When well-integrated, such systems improve both task performance and user satisfaction. For instance, Baeza & Kumar (2019) found that users perceived voice assistants with screens as more useful, thanks to reduced cognitive load and easier information handling. The next section will explore similar findings to inform our IVR design.

**2.3 Multimodal Voice Assistants and Empirical Findings**

Smart displays like the Amazon Echo Show serve as real-world examples of multimodal IVR systems. Baeza & Kumar (2019) studied how visual feedback influences the perceived usefulness of such systems. Participants interacted with a voice assistant under different conditions—some included relevant visuals, others had irrelevant or no visuals—while also performing tasks to simulate cognitive load. The results showed that users found the assistant significantly more useful and had better recall when brief spoken responses were paired with relevant on-screen information. Visuals acted as memory aids, reducing cognitive effort. Based on these findings, the authors recommend designing concise voice outputs supported by meaningful visuals. Our system design follows this advice by ensuring that any lengthy or complex information is displayed textually or graphically, rather than delivered solely via a long voice prompt.

The e-commerce domain offers a compelling context for exploring multimodal voice interaction, particularly in scenarios that combine speech with visual catalogs. Wanigasekara et al. (2023) introduced the “Visual Item Selection” task to address a key limitation in voice assistants: the inability to refer to items shown on-screen using natural voice expressions such as “the second one” or “that blue lamp.” This limitation stems from the lack of grounding between spoken language and visual content. To overcome this, the authors proposed a multimodal system that integrates computer vision and speech processing, leveraging image embeddings from models like CLIP to align voice commands with on-screen elements. Their approach achieved over 70% accuracy in zero-shot reference resolution on a shopping dataset, with further gains through fine-tuning. The study demonstrates that incorporating visual context into voice interfaces enables more natural and flexible user interactions, such as issuing commands that rely on visual descriptors or spatial references. These findings underscore the broader value of visual grounding in multimodal dialogue systems and inform the design of interfaces where users interact with both speech and dynamic visual content.

Bhargava et al. (2023) address the challenge of enabling voice assistants to interpret deictic references to on-screen text such as “call this number” or “open that link.” Traditional assistants often lack access to screen content, making such references ambiguous. To overcome this, the authors developed a modular system that uses text extracted via OCR or accessibility APIs to ground voice commands in the visible GUI. Their model, Screen Reference Resolver (SRR), handles multiple similar items (e.g., several phone numbers) by considering contextual cues like location, text labels, and entity categories. The approach demonstrated strong performance (reduced relative top-1 error by 43% on Descriptive and 83% on Category-level) on their ScreenRef dataset and highlights how integrating GUI context enables more natural and efficient voice interaction. While their work targets smartphone-based assistants, the core principle—maintaining alignment between screen content and speech understanding—is equally relevant to multimodal IVR systems. It informs our own design, where system-generated GUI elements (e.g., account names or time slots) must be linked to user utterances through reference resolution.

Multimodal dialogue systems have also been studied for their impact on user engagement and emotional responsiveness. Gabor-Siatkowska et al. (2024) present *Terabot*, a therapeutic agent initially built with voice interaction and text-based emotion recognition. They found that voice-only interactions sometimes led to mismatches between user emotions and system responses, disrupting conversational flow. To address this, they proposed adding eye-tracking to infer user engagement or affective state, allowing the agent to adapt its dialogue strategy. This highlights the broader value of using implicit feedback (e.g., gaze) to improve responsiveness in multimodal systems.Similarly, Bravo et al. (2025) review AI-based multimodal dialogue systems (including those in customer service) and note that integrating modalities like facial expressions, vocal tone, and gesture enhances emotion and intent recognition. Although our IVR system does not include emotion detection, it aligns with this trend. Future versions could, for example, detect repeated help requests as signs of frustration and simplify the interaction or escalate to a human agent.

Evaluating multimodal systems requires going beyond traditional IVR metrics like call completion or handling time. Additional measures may include how users switch modalities or whether they attend to visual elements. Dzardanova and Kasapakis (2022), in a study on Immersive Virtual Reality (IVR), used eye-tracking to examine how non-verbal cues influenced user attention across different visual stimuli. While we do not have access to eye-tracking equipment, their approach highlights the value of understanding where users focus during multimodal interaction.