International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

https://ijcjournal.org/index.php/InternationalJournalOfComputer/index

Strategies for Handling Barge-in Interruptions in Conversational AI Interfaces

Dhaval Hemant Shah*

Senior Software Engineer

Email: shahdhaval1893@gmail.com

Abstract

The article addresses the problem of handling interruptions (barge-in) in voice-based ordering interfaces operating on top of large language models. The aim of the study is, based on a review of solutions for streaming speech recognition, voice activity detection, turn-taking, and end-to-end dialogue architectures, to synthesise an integrated set of barge-in handling strategies for high-load QSR scenarios. The relevance of the research is determined by the growing share of voice orders in noisy environments and their relationship to transaction accuracy, service speed, and customer satisfaction. The novelty of the article lies in treating interruption as an end-to-end intention marker that imposes coherent requirements on stack architecture, dialogue policy, utterance design, and the transactional logic of the cart. It is demonstrated that effective barge-in handling relies on prioritising user speech over system synthesis, streaming recognition with interim hypotheses, separating stable dialogue state from the instantaneous response plan inside the language model, and phase-dependent policies for interpreting identical acoustic patterns at the stages of greeting, order configuration, confirmation, and payment. It is argued that short, interruptible system utterances, explicit invitations to interrupt, and conservative interpretation of interventions at payment steps transform interruption from a source of errors into a controlled mechanism for order correction and failure mitigation. The article is intended for researchers in dialogue systems, voice interface engineers, and product teams developing voice solutions for large-scale ordering scenarios.

Keywords: interruptions; barge-in; voice interfaces; dialogue systems; large language models; voice ordering.

Received: 11/23/2025
Accepted: 1/23/2026
Published: 2/1/2026

----

* *Corresponding author.*

1. Introduction

In the context of conversational voice systems, interruption, or barge-in, is typically understood as a user utterance that overlaps the system's current spoken turn or occurs before the system completes its current processing step. In contrast to traditional menu-based interfaces with DTMF navigation, the temporal aspect is critical here: the system must be able to distinguish its own planned speech, ambient acoustic activity, and meaningful human intervention, and then, in real time, decide whether to fall silent immediately, ignore the signal, or adjust the dialogue policy. Recent work on acoustic and contextual classification of interruptions shows that reliable barge-in handling requires combining features of the audio stream, the text of automatic speech recognition hypotheses, and information about the current step of the scenario, which makes it possible to substantially improve detection quality and reduce reaction latency compared with simpler models [1]. At the same time, classical studies of continuous interruption prediction in human–human dialogue already demonstrated that appropriate handling of barge-in increases the proportion of completed tasks and reduces session duration, making this capability not merely a convenience feature but a key determinant of dialogue system efficiency [2].

The problem of interruptions is particularly acute in voice-based ordering systems in the quick-service segment, where high load, noisy acoustic environments, and strict constraints on service time and order accuracy dominate scenarios. Under such conditions, even a slight delay in responding to a customer's utterance, or an incorrect interpretation of a phrase such as no, I changed my mind, may lead to order errors, repeated interactions with staff, and direct financial losses. Empirical studies of drive-through service operation show that consumers simultaneously maximise three criteria, speed, convenience, and accuracy, where order accuracy is increasingly identified as the main factor driving repeat usage [3]. It follows that barriers in barge-in handling cannot be regarded as purely technical artefacts: they are directly linked to satisfaction metrics, revenue, and operational efficiency in large-scale voice-ordering scenarios.

The emergence of large language models radically reshapes the contours of the problem. On the one hand, such models provide more flexible dialogue management, better understanding of informal formulations, and the ability to reconfigure the interaction plan after an interruption instantly. On the other hand, they impose new constraints on streaming operations, latency management, and alignment between the model's verbal decisions and the cart's transactional state. Recent work on LLM-based voice interfaces emphasises a shift from cascaded recognition–understanding–generation schemes towards more tightly integrated systems in which speech and text levels are trained jointly, and the model can exploit prosodic and contextual cues to choose the moment for a turn shift and process user interventions [4].

In such architectures, barge-in becomes not merely a signal for stopping speech synthesis, but a rich intention marker around which both dialogue strategies and algorithms for managing the actual order are organised. For this reason, the development of interruption-handling strategies in voice interfaces operating on top of large language models requires a holistic analysis that integrates interface design constraints, speech-processing algorithms, and the transactional logic of the application domain.

2. Materials and Methods

The investigation of interruption (barge-in) handling strategies in voice-based ordering interfaces is grounded in an analytical review and synthesis of existing work on acoustic interruption classification, turn-taking organisation in dialogue, streaming speech recognition architectures, and end-to-end dialogue systems based on large language models [1, 2, 4–9]. As empirical foundations, the study draws on results from contextual barge-in classification in industrial voice assistants [1], surveys of turn-taking mechanisms and overlapping speech handling [2], as well as research on voice-based ordering systems in the quick-service industry that demonstrates how architectural decisions correlate with service speed and accuracy metrics [3, 5].

At the level of speech technologies, the study systematises data on modern streaming automatic speech recognition (ASR) systems and voice activity detectors [6–8]. At the levels of the dialogue and transactional layers, it draws on work on end-to-end task-oriented dialogues and on the integration of generative models with booking and ordering services [4, 9]. This corpus of sources makes it possible to consider barge-in not as an isolated problem of stopping speech synthesis, but as an end-to-end mechanism that influences stack architecture, dialogue policy, and the behaviour of the transactional subsystem.

Methodologically, the work follows a research-through-design approach and comprises several interrelated steps. First, a reconstruction is carried out of a typical architecture for a voice-based ordering system, based on descriptions of cloud-oriented voice solutions and streaming dialogue systems [5–7, 9], with decomposition into layers: transport, voice activity detection, streaming recognition, a dialogue module based on a large language model, speech synthesis, and the transactional backend. Second, based on studies of turn-taking and barrier situations in conversation [1, 2, 4, 8], a taxonomy of interruptions characteristic of quick-service scenarios is constructed: disruptions of an ongoing response, interruptions during the model's latency phase, and interruptions associated with cart correction and payment steps.

Third, for each interruption type, a phase-based scenario analysis is performed (greeting, menu presentation, item configuration, final confirmation, payment). Within this analysis, conceptual sequence diagrams of interactions and dialogue states are constructed, specifying the potential barge-in points and the corresponding transitions between dialogue and transactional states. Finally, by aligning these models with recommendations on the architecture of voice ordering systems and end-to-end dialogues [5, 7, 9], a set of interruption-handling strategies is synthesised: policies for prioritising user speech over synthesis, rules for interpreting interruptions as a function of scenario phase and operation criticality, and principles for constructing brief, easily interruptible assistant utterances. These strategies are then subjected to a qualitative assessment of their expected impact on order accuracy, cognitive load, and the predictability of system behaviour.

3. Results and Discussion

The architecture of a voice-based ordering system leveraging large language models usually constitutes a tightly coupled set of processing layers: a transport layer for audio transmission, a voice activity detection (VAD) subsystem, a streaming speech recognition module, the core dialogue module based on a language model, a speech synthesis system, and a specialised server-side component responsible for the product catalogue, the cart, and payment processing. Studies of modern voice ordering systems in the quick-service industry indicate that precisely such a modular architecture enables simultaneous satisfaction of low-latency, scalability, and transaction-accuracy requirements, while leveraging cloud infrastructure and natural language processing methods [5]. Neural network–based voice activity detectors serve as the front line of defence against noise and define the boundaries of speech segments, which is critical for correct triggering and termination of subsequent modules [1]. Streaming ASR models are gradually shifting from classical hybrid schemes to fully neural architectures that optimise the trade-off between speed and accuracy and support continuous audio-stream processing, which is particularly important in dialogues with overlapping turns [6].

At the top of the stack is the dialogue module, based on a large language model and integrated with order-management tools. This module consumes text hypotheses, updates the dialogue and cart's structured state, and generates a response that is passed to the speech synthesis module, closing the human–machine loop. The data flow in such a system can be described as a sequence of transformations from an acoustic signal to a confirmed transaction. The user’s audio signal arrives through the communication channel into the voice activity detection subsystem, which identifies intervals containing speech and thereby determines the beginning and, equally importantly, the end of the user’s utterance. Both standalone models and those combined with end-of-query detection mechanisms are used for this task, enabling latency reduction and decreasing the probability that the system will begin responding either too early or too late [7].The selected segments are forwarded to the streaming ASR, which produces interim text hypotheses as audio frames are received, allowing the dialogue module to begin intent interpretation before the utterance is complete [8]. The language model, having access to the utterance text and dialogue context, then forms a distribution over actions and calls the server-side component for operations on the catalogue, cart, and payment services. Recent research on end-to-end dialogue systems highlights the effectiveness of such tight integration between the generative module and the transactional layer for booking and ordering tasks [9]. Finally, the result, in the form of a concise but informative formulation, is converted by the speech synthesis system into an audio response, which is then injected back into the same audio channel, creating the illusion of a natural dialogue despite an underlying chain of highly specialised modules.

At every step of this chain, a potential collision may occur between user and system initiative, that is, a locus for interruption. During generation and playback of the response, the speech synthesis module becomes the visible surface towards which most explicit interruptions are directed: the user prematurely signals that the information has already been understood or changes their decision. At this point, the voice activity detector must immediately halt playback and redirect the audio stream to the recognition path. Interruptions of this type frequently concern informational utterances, in which the client says That’s enough, give me…, as well as corrections of specific order items when a new utterance explicitly contradicts the configuration of a dish that has just been described. The literature on turn-taking modelling in dialogue shows that timely recognition of the moment when the system must yield the floor requires not only pause analysis but also the prediction of the partner's utterance completion, which is directly related to robust handling of overlaps and interruptions [2].

A different class of barge-in events is associated with latency within the dialogue and transactional layers, when the user does not hear a response and therefore re-addresses the system or reacts emotionally to the delay. In this case, the interruption does not fall on speech output but on the model's contemplation or on waiting for a response from an external service, and must be distinguished from background noise and incidental interjections so as not to disrupt already initiated order operations. Together, these interruption types define a complex design space: the dialogue module and server-side component must be able to interpret barge-in as an unconditional priority for a new instruction in some cases, as a soft rejection of part of the information in others, and, in yet other instances, as a symptom of connectivity or latency issues. The interaction strategy must be adapted to minimise order errors and cognitive load while preserving the integrity of the system's architectural contour. A sequence diagram of the voice ordering pipeline with barge-in handling is shown in Figure 1.

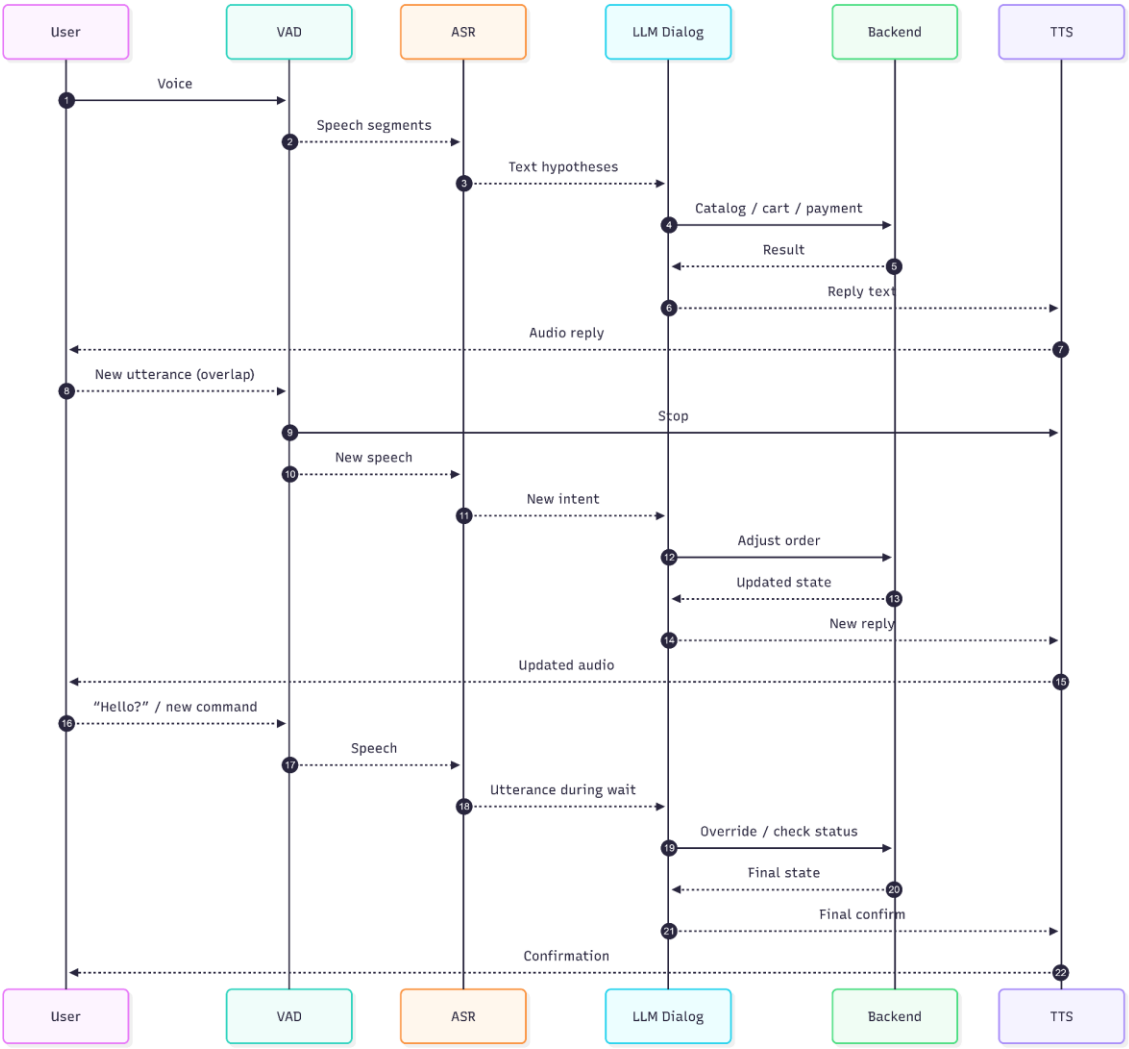


Figure 1: Sequence Diagram of Voice Ordering Pipeline with Barge-In Handling

At the level of interaction design, the key task is to choose assistant formulations and speech rhythm so that interruptions are perceived as a natural component of the dialogue rather than an error. System utterances should be short, informationally dense, and structured so that they can be safely cut off at any point. Instead of a lengthy assortment description with several consecutive clarifications, the system poses a single specific question and pauses for the user's response. It is essential to state in advance that interruptions are acceptable: a phrase such as "if everything is clear, you can interrupt me at any time" and immediately stating order lowers users' barriers to intervention and reduces the build-up of irritation.

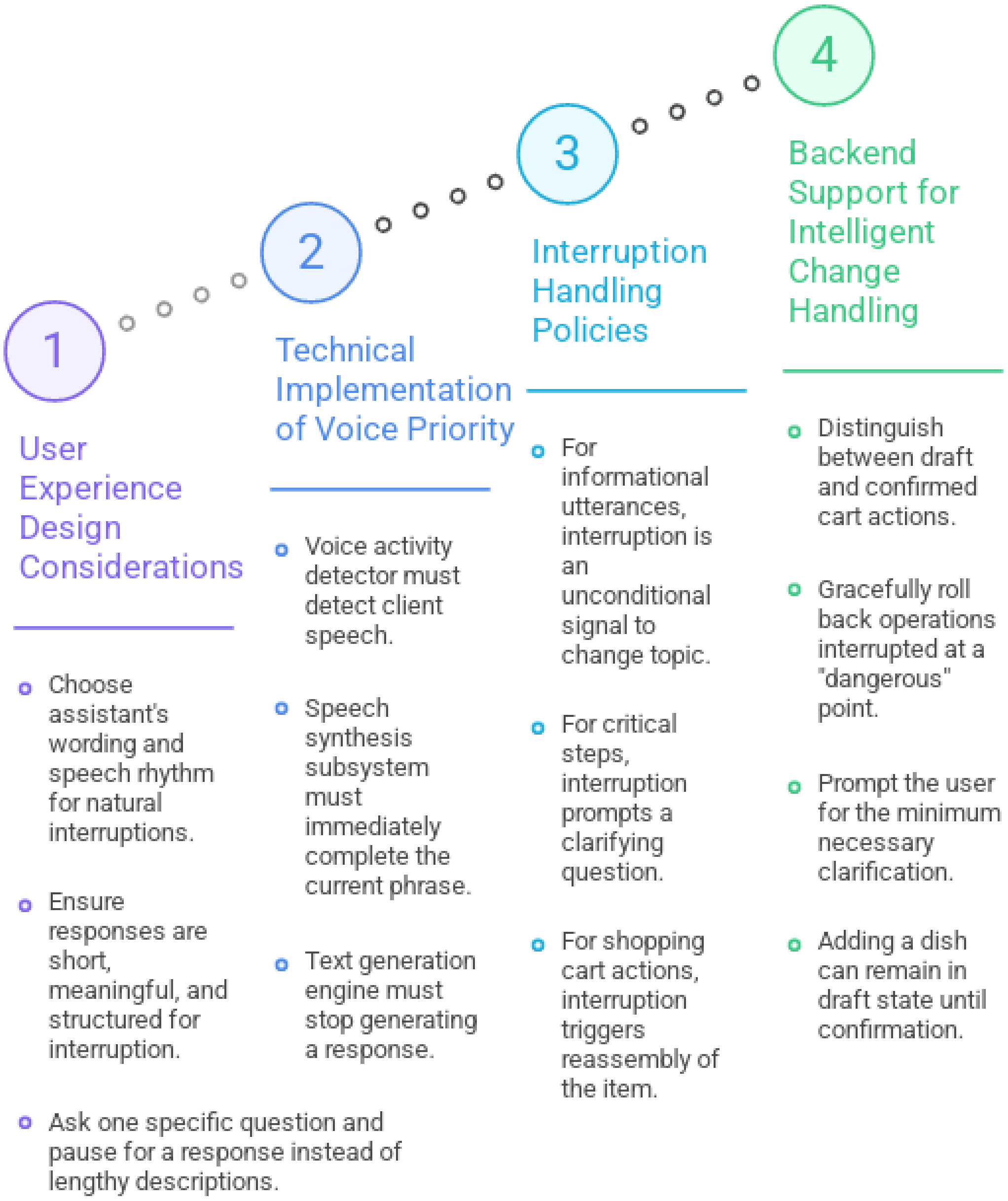
A stepwise interaction style, where each turn contains only a single question and is not overloaded with explanations, allows interruption to be embedded in the dialogue structure. If the client says That's enough, let's do a pepperoni pizza, the system does not revert to reading the menu but proceeds directly to clarifying the chosen item. When working with lists and menus, it is effective to segment content into meaningful chunks: instead of a monolithic enumeration, the assistant first offers a dish type, then several options within the category, explicitly indicating that at any time it is sufficient to say next or name a specific item. After an interruption, very short micro-confirmations help stabilise the shared understanding of the order without triggering a new wave of interventions: got it, no cheese, okay, removing the coffee, yes, moving on to desserts. By contrast, long paraphrases of what the user has just said only increase the probability of another barge-in. Successful formulations in the ordering context are typically minimalist and rely on terminology already introduced into the dialogue. In contrast, unsuccessful ones drag the conversation backwards by repeating details that have already been agreed upon, thereby provoking the client to interrupt the assistant again.

These utterance-level decisions are inseparable from the technical implementation of user-voice priority over system voice. As soon as the voice activity detector registers the onset of user speech, the speech synthesis subsystem must immediately terminate playback of the current phrase, and the text-generation mechanism must stop producing the response, even if the language model has not yet exhausted its internal plan for the utterance. Streaming ASR plays a strategic role here: interim hypotheses make it possible to recognise key intention markers even before the end of the phrase, for example, the opening words no, stop, changed my mind, that's enough, and to switch the scenario without waiting for the complete transcription.

Above the recognition and synthesis layers, interruption-handling policies are defined. For purely informational utterances, an interruption is interpreted as an unconditional signal to change the topic. For critical steps, such as voicing the total amount or initiating payment, the same intervention triggers a clarifying question. For operations on the cart, an interruption launches reconfiguration of the item in question, subject to the new constraint. As a result, even technically identical acoustic interruption patterns are interpreted differently depending on the current dialogue phase and the role of the system utterance being produced.

To ensure that these policies do not undermine order coherence, the server-side component must support meaningful handling of changes: it should distinguish between tentative (draft) and confirmed actions in the cart, be able to roll back operations interrupted at critical points, and request only the minimal clarification required from the user. Adding a dish while the system is still voicing it may remain in a tentative state until the utterance is completed or the client explicitly confirms it. If, at that moment, the user says no, without sauce, the system either corrects the latest operation or offers a simple choice: keep the burger and just change the sauce, correct? When an interruption occurs at the payment step, the logic must be even more conservative: any intervention before the transaction is finalised is interpreted as cancellation of the current charging attempt and a return to a level where the amount or order composition can be safely revised.

This coordinated operation of dialogue strategies and transactional mechanisms allows interruptions to be used not as a source of chaos but as a full-fledged channel for expressing intentions, reducing the user's cognitive load while simultaneously protecting the system from latent errors that would otherwise surface only at the order fulfillment stage. Design principles for natural interruptions in voice assistants are shown in Figure 2.



User Experience Design Considerations

- Choose assistant's wording and speech rhythm for natural interruptions.
- Ensure responses are short, meaningful, and structured for interruption.
- Ask one specific question and pause for a response instead of lengthy descriptions.

Technical Implementation of Voice Priority

- Voice activity detector must detect client speech.
- Speech synthesis subsystem must immediately complete the current phrase.
- Text generation engine must stop generating a response.

Interruption Handling Policies

- For informational utterances, interruption is an unconditional signal to change topic.
- For critical steps, interruption prompts a clarifying question.
- For shopping cart actions, interruption triggers reassembly of the item.

Backend Support for Intelligent Change Handling

- Distinguish between draft and confirmed cart actions.
- Gracefully roll back operations interrupted at a "dangerous" point.
- Prompt the user for the minimum necessary clarification.
- Adding a dish can remain in draft state until confirmation.

Figure 2: Designing for Natural Interruptions in Voice Assistants

At the level of the large language model, the central task is to separate two distinct entities: the stable dialogue state and the instantaneous response plan. The dialogue state includes the cart structure, the current scenario step, the history of key decisions, and a concise, interpretable context summary. The response plan, by contrast, is merely a temporary blueprint of the formulation that the model constructs at a specific moment in the conversation. If everything is stored only as a sequence of messages within the model itself, an interruption destabilises this fragile construction: the model continues to rely on an outdated plan the user has already cancelled, becoming vulnerable to accumulating contradictions.Separating state from words makes it possible, after barge-in, to discard the unfinished response while preserving a correct representation of what has actually been agreed: which cart items are fixed, which ingredient constraints apply, and at which scenario question the dialogue is in fact paused.

To ensure predictable model behaviour under interruptions, system prompts must explicitly specify priorities and conflict-resolution rules. In a basic formulation, this implies a direct instruction to always treat the user's most recent utterance as authoritative, even if it contradicts what the assistant has said earlier, and to interpret explicit negations as cancellation or correction of the previous step. Additional schemas are defined to handle typical situations: how to react when a new command partially diverges from an already confirmed order; how to interpret soft forms of refusal and hesitation; and whether the cart summary should be automatically recomputed after each substantial change or only at the user's request.

In addition to these rules, a tool-invocation layer is constructed that serves as a transactional intermediary between the language model and the ordering system. Each user utterance is first interpreted as a set of operations on the cart: addition, removal, modification of an item, display of contents, confirmation of checkout, and only then is it accompanied by a brief verbal response. This ordering of operations prevents divergence between what the voice assistant promises and what is actually recorded in the system, since the final verbal confirmation always reflects changes that have already been carried out.

Particular tension arises when interruption yields a fragmentary phrase that only hints at an intention: the user says no, without…, stop, let's…, wait, better… and then falls silent or digresses. In such cases, the language model must behave with maximal caution. Instead of hastily inferring the missing meaning on the user's behalf and silently changing the order, a strategy of minimal follow-up is appropriate: clarify which specific item the correction concerns, which ingredient should be removed, or to which scenario step the user is proposing to return. Embedded rules should encourage the model not to discuss the entire menu anew or reformulate the order as a whole, but to link the fragmentary utterance in a targeted way to the most recently modified object, stating this explicitly in a short phrase and, if necessary, posing one additional question. In this regime, interruptions cease to threaten state coherence: rather than breaking the dialogue, they become gentle control impulses that the language model can localise, refine, and safely integrate into the overall ordering process. Maintaining Dialogue State During Interruptions is shown in Figure 3.

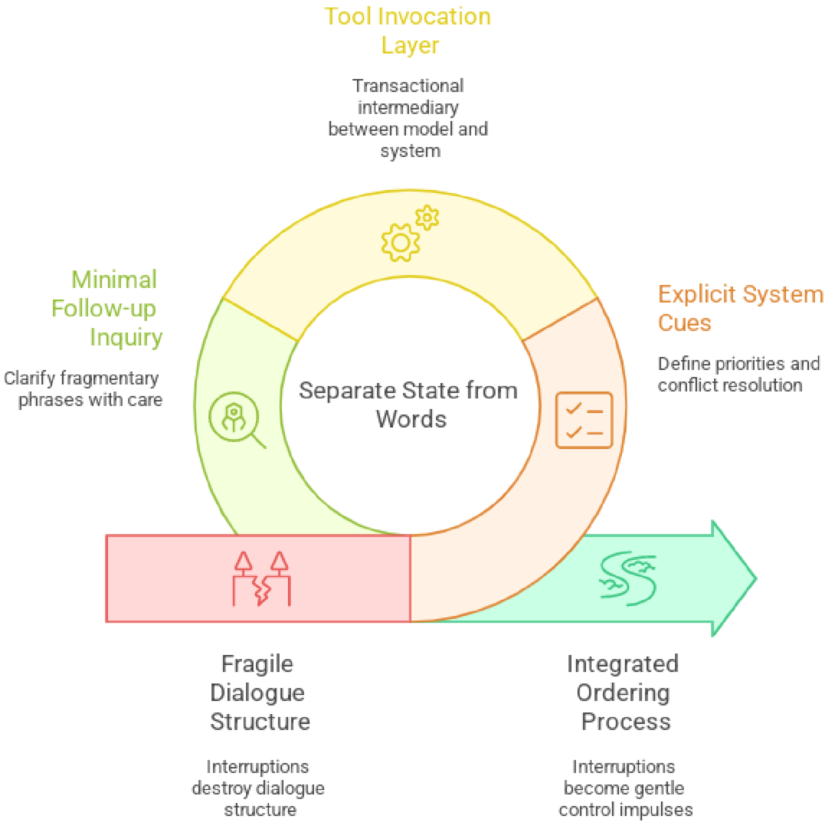

Tool Invocation Layer
Transactional intermediary between model and system
Minimal Follow-up Inquiry
Clarify fragmentary phrases with care
Explicit System Cues
Define priorities and conflict resolution
Separate State from Words
Fragile Dialogue Structure
Interruptions destroy dialogue structure
Integrated Ordering Process
Interruptions become gentle control impulses

Figure 3: Maintaining Dialogue State During Interruptions

The phase structure of dialogue in a voice-based ordering system allows interruptions to be considered not abstractly but as a sequence of stable patterns. At the greeting stage, the main task is to transition quickly to the core of the interaction upon the first user intervention: if the person interrupts a lengthy assistant introduction and immediately states the order, the system must, without taking offence and without returning to the prologue, move directly to item clarification.

During menu and special-offer presentation, interruptions with typical formulas such as that’s all, give me… become the norm, and assistant utterances are therefore designed so that a single such signal is sufficient to switch immediately from list reading to adding a specific dish. In the item-configuration phase, more delicate barriers appear: specifications such as no cheese, soy cream, double espresso, and corrections to options that have already been voiced must modify exactly that part of the cart to which they refer, rather than provoking a complete dialogue reset. Here, the link between the current scenario step and the object that the language model considers active is particularly useful.

Final order confirmation opens the space for another interruption type: the assistant enumerates the final cart, and the user intervenes with utterances such as remove the coffee or add one more sauce. In such situations, the system is obliged to pause the enumeration, carefully change the order composition, recompute the total, and only then succinctly repeat the updated result, avoiding any return to the original, now incorrect version.

At the payment stage, the requirements for caution are higher still: any intervention during voicing of the payment method, amount, or confirmation code is interpreted as a signal to halt the transaction and return to a safe point where no funds have yet been debited and the order composition can be painlessly adjusted. Interruption-handling logic in these late phases follows a conservative principle: it is preferable to request clarification once too many times than to allow double charging or to provide the client with an outcome that does not align with their last explicit intention.

The practical implementation of the described strategies requires alignment among architectural, interaction design, and model-level decisions. At the technical level, a full-duplex audio channel, streaming modes for both recognition and synthesis, and precise commands for immediate stop and resume of playback are necessary so that any user interruption truly interrupts the system rather than lagging by fractions of a second.

At the level of dialogue texts and scenarios, it is essential to predefine short, interruptible assistant utterances, explicit invitations to interrupt, and robust templates for follow-up questions after ambiguous interventions. The large language model must be configured through system prompts and tool sets so that it first adjusts the structured state of the cart and scenario step, and only then formulates a response that reflects the factual order state. Finally, an overarching cycle of verification and improvement should be established: predesigned test sessions with frequent interruptions, load testing that records reaction time and error frequency, pilot deployments with analysis of real dialogues, and controlled experiments comparing different barge-in handling strategies in terms of customer satisfaction, dialogue duration, and final order accuracy.

4. Conclusion

The article demonstrates that handling interruptions in voice-based ordering systems, particularly under quick-service conditions, cannot be treated as a narrowly technical problem at the level of speech recognition or voice activity detection. Barge-in functions as an end-to-end marker of user intention that permeates the entire stack, from voice activity detectors and streaming recognition to the dialogue module based on a large language model and the transactional cart logic. The analysis shows that only through coordinated operation of these layers is it possible to simultaneously ensure low latency, high-order accuracy, and predictable system behaviour in noisy environments with overlapping user and system turns. Key elements here include prioritising the human voice over the assistant voice, streaming recognition with interim hypotheses, separating stable dialogue state from the instantaneous response plan within the language model, and context-dependent policies for interpreting the same acoustic interruption pattern as a function of the scenario phase and the role of the current system utterance.

At the same time, the work shows that barge-in handling strategies are inseparable from the design of user interaction and transactional reliability. Short, easily interruptible assistant utterances, explicit invitations to interrupt, a stepwise dialogue style, and minimalist follow-up questions after fragmentary interventions transform interruption from a source of chaos into a natural and controllable mechanism for order correction. On the server side, this is mirrored by distinguishing between tentative and confirmed actions, conservative treatment of any interventions at critical stages such as payment, and structuring the operation sequence so that verbal confirmations always follow changes that have already been applied to the cart.

Taken together, architectural choices, dialogue principles, and large language model configurations form an integrated framework for voice interfaces in which interruptions not only fail to disrupt state coherence but are systematically exploited to reduce cognitive load, improve accuracy, and robustly achieve target satisfaction and operational efficiency metrics in large-scale voice ordering scenarios.

References

[1] D. Bekal, S. Srinivasan, S. Ronanki, S. Bodapati, and K. Kirchhoff, "Contextual Acoustic Barge-In Classification for Spoken Dialog Systems," *Interspeech 2022*, pp. 1091–1095, Sep. 2022, doi: https://doi.org/10.21437/interspeech.2022-408.

[2] G. Skantze, "Turn-taking in Conversational Systems and Human-Robot Interaction: A Review," *Computer Speech & Language*, vol. 67, p. 101178, Dec. 2020, doi: https://doi.org/10.1016/j.csl.2020.101178.

[3] M. Şehirli, "A New Qualitative Measurement Of Customer Expectations And Satisfaction And Cross-Brand Comparison In The Automotive After Sales Services Industry," *International Journal of Management Economics and Business*, vol. 19, no. 4, pp. 883–909, Sep. 2023, doi: https://doi.org/10.17130/ijmeb.1292817.

[4] C.-H. H. Yang, A. Stolcke, and L. Heck, "Spoken Conversational Agents with Large Language Models," *arXiv*, Dec. 2025, doi: https://doi.org/10.48550/arxiv.2512.02593.

[5] S. M. Devaraj, "AI and Cloud-Enabled Voice Ordering Systems: The Future of QSR Customer Interaction," *Zenodo*, vol. 13, no. 1, pp. 1–13, Jun. 2023, doi: https://doi.org/10.5281/zenodo.14762605.

[6] H. Ahlawat, N. Aggarwal, and D. Gupta, "Automatic Speech Recognition: A survey of deep learning techniques and approaches," *International Journal of Cognitive Computing in Engineering*, vol. 6, pp. 201–237, Jan. 2025, doi: https://doi.org/10.1016/j.ijcce.2024.12.007.

[7] B. Li *et al.*, "Towards Fast and Accurate Streaming End-to-End ASR," *arXiv*, Apr. 2020, doi: https://doi.org/10.48550/arxiv.2004.11544.

[8] A. Addlesee, Y. Yu, and A. Eshghi, "A Comprehensive Evaluation of Incremental Speech Recognition and Diarization for Conversational AI," *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 3492–3503, Jan. 2020, doi: https://doi.org/10.18653/v1/2020.coling-main.312.

[9] L. Qin *et al.*, "End-to-end Task-oriented Dialogue: A Survey of Tasks, Methods, and Future Directions," *arXiv*, Nov. 2023, doi: https://doi.org/10.48550/arxiv.2311.09008.