amazon alexa

Panasonic
AUTOMOTIVE

Multiple Voice Assistant Dialogs and Arbitration

by Amazon and Panasonic Automotive

Voice Interoperability Initiative Architecture Series Whitepapers

August 23rd, 2023

Executive Summary

The Voice Interoperability Initiative (VII) is focused on improving the interoperability between different voice services on a single device, with the goal of providing consumers with more choice and flexibility in how they use voice assistants [1]. VII embraces the following pillars of voice interoperability:

- Customer Choice — Building voice-enabled devices that promote customer choice and flexibility through multiple, simultaneously available wake words.
- Secure Interoperability — Developing voice services that work seamlessly with others while protecting the privacy and security of customers.
- Technology Solutions — Releasing technologies and solutions that make it easier to integrate multiple voice services on a single product.
- Research and Development — Accelerating machine learning and conversational AI research to improve the breadth, quality and interoperability of voice services.

This paper is a step on the path to providing customer choice and technology solutions that make it easier to integrate multiple voice services on a single product. We discuss the benefits of multiple voice assistants on a single device, the importance of unique multi-modal cues while in a dialog with a person, and a voice assistant arbitrator to coordinate voice assistant activation. Using the discussed guidance and framework, developers can better understand how to create products that allow customers to interact with multiple voice assistants on a single device in a predictable and seamless manner.

Abstract

When a small group of people speak with one another, generally one person speaks at a time while others listen because it is difficult to speak and listen at the same time. People in the group take turns speaking and use multi-modal cues such as gaze, gestures, pauses in speech, and sentence completion to know when they may speak. Modeling this type of turn-taking behavior between a person and a voice assistant on a multi-voice assistant device may be achieved using a dialog metaphor, where interactions are confined to one person and one voice assistant at any particular time. A person asks a voice assistant for something and the voice assistant responds. The goal of a dialog with a voice assistant is typically to achieve a specific outcome or goal, such as providing information or completing a task, which can generally be conveyed in a single, sometimes compound, sentence.

When a person speaks with a voice assistant on a multi-voice assistant device whose inputs are constrained to microphones or a voice command button, and whose outputs are constrained to LEDs or a display and a speaker, it becomes necessary to use different kinds of cues to facilitate a smooth dialog. Distinct visual and sound cues are necessary to distinguish the various phases of a dialog and the active voice assistant, along with a recognizable voice.

Two common methods a person may use to cue a voice assistant into a dialog are to speak its wake phrase (a.k.a. a wake word, wake-up word, hot word or keyword) or to push its voice command button. When the selected voice assistant activates, the device typically renders a unique visual animation and plays a distinct sound or responds in a recognizable voice with “uh huh” to convey that it is listening. These and other cues can provide a signal for users to know when they may speak and when a response or action by the voice assistant is forthcoming or complete.

The stages of a dialog and the CX policies of a voice assistant are important for ascertaining whether a user may barge-in on a speaking voice assistant using another voice assistant’s wake phrase to start a new dialog, and whether a voice assistant may become active and start speaking. Also, when a dialog with a voice assistant is already underway, it is important that other voice assistants do not simultaneously speak to avoid confusing the user.

A multiple voice assistant arbitrator is useful for coordinating interactions between users and voice assistants. When an arbitrator implements a dialog metaphor, a natural and predictable turn-taking order that also supports data privacy is possible. Also, having a voice assistant’s application software first make a request with the arbitrator in order for it to activate and enter into a dialog with the user is a way to prevent multiple voice assistants from becoming simultaneously active.

This paper discusses many aspects that must be considered when combining voice assistants on a single device and offers a framework for managing dialog-driven interactions.

Contents

Intended Audience

This paper is intended for readers interested in modern voice assistant technologies on multiple voice assistant devices and covers aspects useful for decision makers, device makers, systems integrators, voice assistant developers, technical architects, and engineers. Decision makers can learn about the benefits of having multiple voice assistants simultaneously available to their customers. Device makers may benefit in learning about a framework for arbitrating multiple voice assistants to support coordinated turn taking between a user and a voice assistant. Others may benefit in learning how coordinated turn-taking in dialogs between a user and a voice assistant is an effective means for getting information and performing tasks by voice.

Introduction

Voice assistants on devices are like personas. The more they adhere to the norms of human etiquette in verbal conversations, the more that interactions with them become predictable, natural, and seamless. When multiple personas are involved in a conversation, the rules of etiquette and natural turn-taking become increasingly important, and this applies to user interactions with multiple voice assistant devices as well.

This paper discusses the various aspects that should be taken into consideration when implementing multiple voice assistants on a single device such that the rules of etiquette and natural turn-taking are understood and may be applied to user dialogs with voice assistants. The reader is provided with a comprehensive understanding of the importance of dialog management and how cues and the conveyance of the voice assistant states for interactions with multiple voice assistants on a single device relate to dialog management. We begin with a description of the various types of voice assistants and the benefits of integrating multiple voice assistants on a single device. Special attention is given to the limited availability of multi-modal cues on these devices and how natural turn-taking techniques can aid in mitigating this limitation. To facilitate effective turn-taking between users and voice assistants, the adoption of a dialog metaphor is discussed. Furthermore, the implementation of a voice assistant arbitrator and accommodative application program behaviors are proposed as mechanisms to ensure that activation of only one voice assistant occurs at any given time to facilitate seamless user interactions and safeguard data privacy. Additionally, operating-system based methods for voice assistant arbitration are also discussed as are aspects related to voice assistant initialization and enablement.

The more common voice and button activation input modalities of voice assistants are covered, whereas other modalities such as multisensory wake detections used for video pose detection and text input are not. Multiple voice assistant scenarios spanning multiple devices or multiple zones with multiple users, such as those gaining popularity in motor vehicles, are not covered as these are baselined in single user interactions with a multi-voice assistant device. Additionally, it is assumed that the speech enhancement and noise removal algorithms in the audio front end of the device adequately isolate the user's speech from any background

conversations or noise and as such, are not factored into the discussion. False positives from wake word (or phrase) detectors and custom wake phrases are also not discussed because their role for providing high quality detections is independent of the role of a voice assistant arbitrator [4].

Less technical readers may skip the sections on Voice Assistant Dialog States, Arbitrating Multiple Voice Assistants, and Operating System-Based Arbitration.

What is a voice assistant?

A voice assistant is a type of software program that utilizes natural language processing (NLP) and natural language generation (NLG) technologies to recognize and respond to voice commands spoken by a user. Commands may include things like setting a reminder, playing music, or checking the weather. These machine learning and artificial intelligence technologies enable voice assistants to understand the context of the user's request and provide more personalized and relevant responses in a way that is natural and easy to understand.

Today's voice assistants utilize stored information and prebuilt knowledge graphs or access online data sources to provide responses or perform actions. These actions may include things like providing answers to questions about past events or people, adding items to shopping lists, making purchases, playing music, providing in-vehicle navigation and car control, setting smart home devices, and more. Voice assistants provide a hands-free experience that "frees up" the user's hands and generally their eyes as well.

Conversational voice assistants are more advanced in that they are designed to carry out more complex interactions with users. These interactions are designed to simulate a human-like conversation, where the user can ask follow-up questions, provide additional information, or clarify their requests. These follow-up questions typically do not require that the user first speaks the voice assistant's wake word. For example, the Alexa app provides a setting called Follow-Up Mode, which empowers the user to continue the conversation with Alexa without the use of a wake word.

Types of Voice Assistants

A voice assistant (VA) may come pre-installed on a device or be installed by the user at a later point in time. It is usually linked to and backed by services hosted over the internet by the VA provider. Voice assistants installed on the same device may perform different tasks, offer different features or capabilities, or have similar capabilities. They may also support varying degrees of interoperability. For example, a VA may handoff a user's voice request to another VA that is able to fulfill the request [6]. Also, one of the voice assistants on the device may be designated as the default VA, which activates when a gesture is performed by the user, such as pressing a specific button [7].

A voice assistant may also utilize a display on the device to present information in response to user utterances, enrich the dialog, and allow for selections or settings. This applies to smart display devices, modern automotive infotainment systems, and specialized solutions designed for people with certain types of disabilities.

A voice assistant may only be temporarily available on a device that can host multiple voice assistants simultaneously. The most prominent example of this type of integration comes from the automotive industry, where the in-vehicle infotainment (IVI) system supports native voice assistants and the connection of a smart phone that, when connected, projects onto the IVI display and utilizes the vehicle's microphones and speakers. For example, owners of modern cars supporting Apple CarPlay can connect their iPhone to interact with Siri through the IVI system and simultaneously interact with the native voice assistants. The processing of the voice requests for Siri and the subsequent synthesis of the voice responses occurs on the connected smartphone or on Siri's cloud-based voice services. Whereas, the processing of voice requests for the native voice assistants and the synthesis of the voice responses occurs locally on the IVI system or the respective cloud-based voice services.

Many modern voice assistants are fully "reactive" in their interactions with the user, only partaking in a voice dialog in response to an activation of the VA. However, as some voice assistants become more personalized and closely integrate with the lifestyle and daily activities of the user, they may behave proactively in certain situations. For example, a VA may automatically start a dialog with the user on behalf of an earlier setup of a reminder, calendar events, or having expressed interest in getting weather alerts. A more advanced form of a proactive VA is a personal health assistant that periodically asks the user how they are feeling and based on their response, may inquire more or less frequently and make suggestions to address their health condition. Voice assistants may also offer a setting such as Do-not-Disturb to prevent the VA from initiating a dialog.

Voice assistants may depend on the availability of an internet connection or be able to run locally without it. In the past, often assistants embedded in devices and smartphones fully relied on cloud-based backend services. However, present-day neural network technologies for NLP and NLG combined with faster hardware make it possible to handle more voice dialogs locally on the device.

Benefits of Multiple Voice Assistants

There are several benefits to having multiple voice assistants on a single device. Different voice assistants may have different features, strengths, and weaknesses. Offering multiple voice assistants on a single device gives customers greater choice and a broader range of capabilities, and the flexibility to choose the one that works best for a particular task or situation. For example, some voice assistants may be better at navigation, or shopping and purchasing, or controlling smart home devices or in-vehicle components. Also, voice assistants vary in their ability to perform certain tasks. By offering choice, customers may choose the best experience.

Having multiple voice assistants also benefits companies that want to have their own branded voice assistant with specific features applicable to their product. These product-specific features are complemented by the different features offered by other voice assistants on the same device.

Device makers also benefit from being able to have multiple voice assistants on a single device because it reduces the number of distinct products, SKUs, and overall software development and support costs.

Challenges Facing Combined Voice Assistants

While having multiple voice assistants on a single device may offer greater choice and flexibility, it also presents a number of usability and technical challenges that must be addressed to provide a seamless, effective and high-quality customer experience.

Voice Assistant Discovery

One of the main challenges with the use of multiple voice assistants on a single device is that users may become confused about which voice assistant to use for different tasks, leading to frustration and a poor user experience. This problem is alleviated by introducing the available voice assistants and their features when the device powers up and by offering a means for learning more.

Invoking Voice Assistants

Invoking a specific voice assistant should involve simple, easy-to-remember methods. If a voice assistant can be invoked using a wake phrase, it should be distinct from the wake phrase of every other voice assistant on the device to minimize the chance that a device mistakes one wake phrase for another. If users are allowed to choose alternative wake phrases to invoke a voice assistant, it is important to be aware of other voice assistant wake phrases and alternative options. Also, wake phrases themselves should also be easy to remember to avoid customer confusion.

If the multiple voice assistant device affords distinct buttons to invoke each of the voice assistants, the button should be clearly labeled to indicate the specific voice assistant it invokes. This also applies to virtual buttons shown on a touch-sensitive display. Whereas, if multiple voice assistants share a common button for invocation (i.e., the button is overloaded), the distinct patterns (e.g., single- verses double-press) to invoke a particular voice assistant should be easy to perform and made clear during voice assistant discovery. The overloaded button should also provide quality haptic feedback (e.g., a click) and its invocation should coincide with distinct visuals that confirm the invoked voice assistant.

Voice Assistant Activation

Voice assistants may have different policies governing activation. For example, a voice assistant may only allow the use of its own wake phrase in order for the user to barge in while it is speaking, whereas another voice assistant may allow the user to barge-in with any voice assistant's wake phrase, a guideline in the Multi-Agent Design Guide [2]. Another example involves the use of a voice assistant activation button, where its invocation may require that any ongoing dialog between the user and a voice assistant immediately cease and activate the button-designated voice assistant, regardless of the state of former active voice assistant (e.g., during the listening, thinking or speaking states). The device maker must consider whether these policies should apply to all voice assistants on the device and work with the voice assistant providers to achieve a consistent user experience.

Identifying Voice Assistants

Users should always know what voice assistant they are interacting with. This is especially important for voice assistant attribution and branding in multiple voice assistant experiences. Voice assistant attribution may be explicit, such as through colors or logos, or it may be conveyed through different visual and sound cues. It can also include the personality, behavioral characteristics, and the voice that is unique to each voice assistant. The wake phrase, along with the voice assistant's voice, are significant components of persona and brand.

The Multi-Agent Design Guide [2] encourages the use of unique visuals for voice assistants when they activate, listen, stop listening, and speak. The device maker must implement each of the voice assistant's expected visual and sound cues, and ensure that these are visible and can be heard for any voice assistant operating in the expected environmental settings of the multi-voice assistant device. For example, a device that is intended for outdoor use should account for the possibility of bright sunlight or nighttime use and incorporate display technology that adjusts for these very varying conditions, where the visuals utilized by each of the voice assistants are always visible. Similarly for devices that may be subject to wind noise such as in a convertible car, where the audio front end should be able to filter this noise for improved voice recognition for all voice assistants and the speaker system be capable of louder volume levels. By utilizing technologies that compensate for the intended environmental conditions, a quality user experience for each of the voice assistants can still be achieved.

Competing Audio Output

More than one of the voice assistants on a device may offer music playback or out loud reading of an e-book. When a user initiates a dialog or phone call with a voice assistant and music or an e-book is playing, its volume level may interfere with the voice assistant's ability to discern what the user said or the user's ability to hear the voice assistant's response. To minimize the effects of interference, the device needs to pause the audio playback or lower its volume (a.k.a. duck down) while a voice assistant is speaking or capturing the user's speech.

Do Not Disturb

A voice assistant may offer a Do-not-Disturb feature, where the earcon cues or voice audio need to be suppressed for this user setting. For example, the Do-not-Disturb feature in Alexa suppresses announcements, notifications, incoming calls, messages, drop-in, and Ring Doorbell chimes, but does not suppress alarms, timers, and Alexa responses to user inquiries, requests or commands. The voice assistant onboarding process for voice assistants that offer this feature should make clear which features are and are not suppressed.

Voice Assistants Coordination

In order to ensure that one voice assistant does not interrupt the other, it is necessary to design software that explicitly prevents this. For example, the customer may have asked a voice assistant to remind them of a particular event in the future. If this reminder speaks while another voice assistant is speaking, it may result in gibberish voice output, conflicting responses or behavior, and confusion. The device software may avoid this situation by providing middleware that allows for the coordination of voice assistant activation.

When multiple voice assistants are involved, additional software that coordinates turn-taking in dialogs may be required to ensure that only one voice assistant is speaking at any time such that interactions are natural and predictable. Utilizing this approach consistently across all voice assistants on the multi-voice assistant device is important for a high-quality customer experience.

The operating system (OS) platform may implement app sandboxing for robust execution and to provide security boundaries for app code. On devices that utilize these OSes, it may be necessary to utilize an inter-process communications protocol and develop a messaging scheme in order for voice assistants on the device to interface with a common middleware that coordinates dialog turn-taking with one voice assistant at any time.

Display Policies

Some voice assistants have established specifications for displaying information on a screen. These specifications may impose a need for considering alternative ways of displaying information to provide a consistent user experience across the multiple voice assistants on the device. For example, Siri CarPlay has certain requirements for other voice assistants to occupy the entire display versus a portion of the display depending on the type of information and whether certain kinds of user interactions with the display are required. By convening on common or compatible ways of displaying information in a consistent manner, a higher-quality user experience may be achieved.

Voice Assistant Enablement

The multiple voice assistant device may incorporate an OS or service bundle that enables the OS vendor's voice assistant by default, such as Google Automotive Services (GAS), which enables the Google voice assistant by default. Also, depending on the OS version and the device, it may be necessary to disable one of the voice assistants before another can be enabled. In these cases, the device maker may need to include documentation that explains the procedure for enabling voice assistants.

Voice Assistant Arbitration

The OS may offer or enforce voice assistant arbitration, in which case a specialized arbitrator may not be needed or might only be used for a subset of the voice assistants.

Available System Resources

Running multiple voice assistants may consume significant system resources and require substantial CPU, memory or electric power, and thereby increase system cost, create heat dissipation challenges and reduce battery life.

Bringing Voice Assistants Together

The Multi-Agent Design Guide [2] offers guidance on how to address many of the challenges facing devices with multiple voice assistants. One way to educate users about the capabilities of a voice assistant is during the onboarding process, when the user initially signs-in to their account for the voice assistant or enables it. The voice assistant may speak to describe the kinds of capabilities it has and offer a way to ask for help, such as "What can I say?" For devices with a display, the onboarding process may walk the user through a wizard that covers the capabilities of the voice assistant. This onboarding process may also familiarize the user with the voice assistant's unique voice and explain which button or button pattern to use to activate the voice assistant.

When a device is in a noisy setting, such as in a convertible car with a lot of wind noise, the audio front end may be designed to perform well at capturing the user's voice and reducing the background noise, but the user may not be able to hear or discern the voice assistant's distinct sound queues and voice. To address varying noise conditions, volume controls should be made available for the voice assistant's sound cues and voice. For example, the media playback and notification volume controls may also simultaneously adjust the volume level of sound cues and the voice assistant's voice, respectively. Doing so allows the user to continue to benefit from a hands-free experience in noisy environments. The voice assistant may also offer a capability for automatically adjusting its voice to be louder when loud sounds or noises are detected in the background.

When multiple voice assistants are brought together on a single device, it is necessary to understand the requirements of each of the voice assistants before embarking on an integration and implementation. It may be necessary to recognize where inconsistent policies exist and whether they will result in conflicting user experiences with a negative impact on the user. If so, it may be necessary to seek waivers and adapt the device software and voice assistant behaviors accordingly. Sometimes it may only be necessary to adjust the ways in which information is presented to meet voice assistant requirements. For example, a device maker may opt to occupy the entire display for information that needs to be presented by a particular voice assistant to circumvent the underlying complexities of partial or full screen requirements. Similarly, a voice assistant may restrict its policy for allowing a user to barge-in with any voice assistant's wake phrase to just the speaking voice assistant's wake phrase to provide for a more consistent user experience.

A device maker may implement or leverage existing inter-process communication techniques on the device to address the security constraints that may be imposed by a secure voice assistant needing access to a common middleware that helps coordinate voice assistants and user interactions.

User and Voice Assistant Dialog Cues

Having more than one voice assistant on a device requires careful consideration to ensure that the user knows which voice assistant is being addressed. A user typically initiates a dialog with a voice assistant by speaking a wake phrase or by a gesture, such as pushing or tapping a button. This initiation is a cue for the requested voice assistant to engage in a dialog with the user.

By default, voice assistants are designed to only respond to specific wake words or wake phrases. For example, "Alexa" is the default wake word that is spoken to wake up the Alexa voice assistant such that it starts listening for user speech and "Hey Google" and "OK Google" are the default English wake phrases for the Google Assistant. It is important that the wake phrase of each voice assistant be distinct and not sound similar in whole or in part to any of the other wake phrases on the multi-voice assistant device. The Multi-Agent Design Guide [2] and the Multi-Agents Wake Words paper [4] discuss the importance of wake phrases being distinct to avoid misfires and subsequent invalid activations of an unintended voice assistant.

Being able to invoke a voice assistant by voice with a wake phrase is important because it allows for hands-free and potentially eyes-free use without a need for physical interaction with the device. This is particularly useful when driving a vehicle or cooking in the kitchen. It is also important that all enabled voice assistants on the multi-voice assistant device are able to be invoked by the user at any time, with an understanding that invoking a voice assistant does not necessarily imply that it will be immediately activated because a dialog between a user and another voice assistant may already be underway. Disabled voice assistants are not expected to activate when invoked.

Once the voice assistant becomes active, it responds with a cue indicating that it is attentive and ready for the user's inquiry, request, or command. The cue typically consists of a visual, such as a distinct light ring color and pattern, a sound, or a non-lexical backchannel vocalized sound like "uh huh" to indicate attentiveness. It is at this time that the voice data starts streaming it to its speech recognition capability, which may be running in a cloud service or locally on the device. These voice assistant cues and the input audio capture must be timely such that the user does not begin speaking before the voice assistant is ready. Conveying this audio capture state also lets users know when their voice is being captured and possibly recorded on the device or in the cloud.

The voice assistant may also use acoustic cues combined with visual information from a device with a camera to infer a person's body position to distinguish device-directed versus non-device-directed speech and whether they are likely to be addressing the voice assistant [5]. This approach can distinguish device-directed speech even when multiple speakers are interacting with each other and the voice assistant.

A proactive voice assistant may initiate a dialog with the user on behalf of a scheduled activity or situational awareness and user settings. In this case, the voice assistant provides distinct cues based on the nature of the circumstance. For example, a voice assistant that is able to detect a person falling or the breaking of glass may treat this as a potential high-priority or emergency situation, provide a cue with a distinct color or sound, and initiate a dialog with the user to confirm whether help is needed. Voice assistants on a multi-voice assistant device may need to give certain higher priority cues higher precedence.

The voice assistant may use cues in the user's request, inquiry or command to determine whether they are done speaking, such as when speech is absent for a duration that is longer than natural pauses in and between sentences. Also, the voice assistant may incorporate a timeout to limit the user's speech input to reasonably short requests, inquiries or commands.

Cues provide for a more natural and effective conversation and play a crucial role in facilitating smooth conversations. They are especially important in human and voice assistant dialogs because they help guide the conversation and prevent confusion, and inform the user when the voice assistant is listening and expecting the user to speak, as well as when it is in the thinking state or speaking. They also aid the user in timing their interactions such that they are not speaking before the voice assistant is ready to capture and recognize their speech.

Devices with multiple voice assistants may provide different cues and voices for each voice assistant such that the user is able to discern the particular voice assistant that is actively participating in the dialog. They also impose a need for distinct wake phrases that sound different in whole or in part, such that a voice assistant is not unintentionally activated [4]. Additionally, the interplay of voice assistant cues and dialogs is also important such that communication is effective and not diminished. For example, when a voice assistant is already

in a dialog with the user, another voice assistant may opt to play a distinct sound cue to notify the user that it has pending information for them.

Cues cover a range of situations. Having distinct wake words for each voice assistant allows a user to address a specific voice assistant without activating others, which helps prevent confusion and ensures the requested voice assistant responds. Distinct visual indicators, such as unique LED light ring colors or animations on a screen, show which voice assistant is active, listening and responding, which helps users understand which assistant they're interacting with at any time. Different audio cues may be assigned to each voice assistant such that users know when their voice is being captured or a command has been successfully processed by the intended voice assistant. They also help the user know when they may speak. Cues also enhance contextual understanding. Advanced voice assistants may understand the context of a conversation and hand off tasks or requests to another voice assistant if necessary [6]. Cues also facilitate natural turn-taking such that voice assistants avoid talking over one another. By leveraging cues, multiple voice assistants can work together to provide a seamless, effective and friendly user experience.

Dialog Metaphor

One approach to modeling turn-taking in a conversation between a person and multiple voice assistants on a device is to use a dialog metaphor, where interacting with a voice assistant is much like having a conversation between two participants. A person asks a voice assistant for something and the voice assistant responds. Dialogs with present-day voice assistants generally consist of a person speaking a single or compound sentence and the requested voice assistant giving a response or performing an action. The voice assistant may also subsequently ask the user for additional input or clarification, or be set to a follow-up mode where it continues to listen for a short period of time and responds to any additional voice commands, requests or responses. Figure 1 illustrates a voice assistant dialog lifecycle that models human and voice assistant interactions.

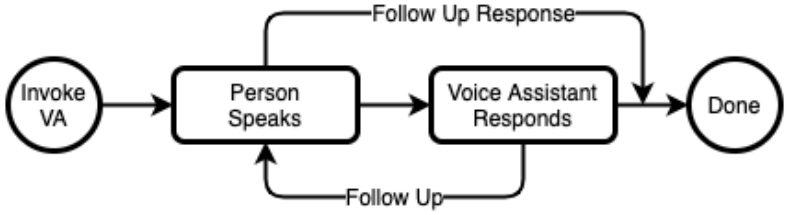

Figure 1 - Voice Assistant Dialog Lifecycle

Voice Assistant Dialog States

Voice assistants have different operational states while in a dialog. Table 1 lists various states a voice assistant may be in before and during a dialog with the user. Prior to engaging in a dialog, a voice assistant is normally in an *idle* state, where it passively waits until it is invoked. Once a

voice assistant is invoked, it typically enters into a *listening* state to capture input voice audio. Subsequent automatic speech recognition (ASR) and natural language understanding (NLU) determine the user's intent and further processing then generates a response or performs an action. The ASR, NLU and response generation steps are treated as a *thinking* state by some voice assistants. Finally, the response is spoken using voice synthesis during the *speaking* state. Some voice assistants combine the *listening* and *thinking* states into a single *recognizing speech* state. Also, in multi-turn dialogs, a voice assistant may be in the *expecting* state while awaiting a response to a question. The listening, thinking, and expecting states may also timeout when the expected behavior doesn't occur within a certain period of time.

State	**Description**
	The voice assistant is …
Idle	waiting to be invoked
Listening	actively listening and capturing voice input
Thinking	processing voice input and determining an appropriate response
Speaking	responding in a synthesized voice
Expecting	expecting a response from the user

Table 1 - Voice Assistant Dialog States

On products with visual cues, each of the above states should be distinguishable. Some voice assistants absolutely require it. Of special importance to the user is knowing when their voice is being captured and sent to Cloud services for further processing. Audio cues may be provided to indicate when a voice assistant has started or stopped listening, or has performed an action, where a voice response need not accompany it. For example, a voice assistant may play a chime or sound when a light is turned on after the user's command and forego providing an accompanying voice response. Whereas, audio cues are generally not used while a voice assistant is speaking. Different sound cues may be played for different scenarios on multi-voice assistant devices.

Figure 2 shows the dialog state transitions in single turn and multi-turn dialogs, as well for proactive voice assistant-initiated dialogs with a user. The starting state is Idle, where voice assistants wake phrase detectors passively listen for a wake phrase. Once a wake phrase is spoken and detected or a voice assistant is invoked using a voice command button, and a dialog with the user is not already underway, the voice assistant may enter its listening state and start capturing speech.

Speech capture continues until the voice assistant detects an absence of speech or the person releases the voice command button. The voice assistant then enters a thinking state where it processes the user's speech and determines what the response or action will be.

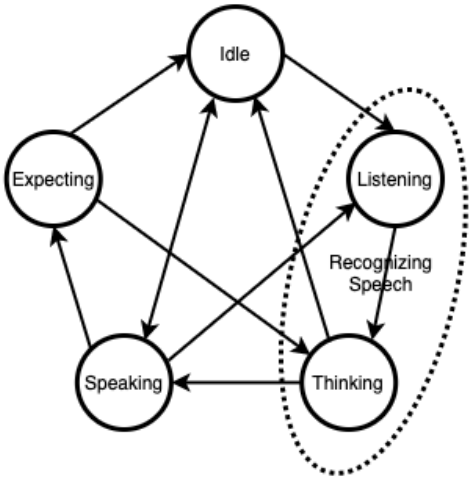

Figure 2 - Dialog State Transitions

Depending on the outcome of the speech processing, the voice assistant may proceed to the speaking state and speak the response, conduct an action such as for a voice command where a sound is played to indicate that it performed the requested command, or take no action such as when the user says “never mind” or “cancel.” It then returns to the Idle state.

The voice assistant may, in its speaking state, ask the user a question or request more information as part of a multi-turn dialog, after which it proceeds to its expecting state (or its listening state if it doesn’t have one) to listen for any further user voice input that is then processed. Whereas, if the voice assistant supports a follow-on mode, it listens for a brief period after its speaking state, where the user is not required to repeat the wake phrase for any subsequent request, inquiry or command. Once the request, inquiry, or command is completed, the voice assistant then returns to the Idle state.

A proactive voice assistant that initiates a dialog with the user proceeds from the Idle state directly to the speaking state, after which it may proceed to the listening, expecting (if applicable), or idle states depending on the intent of the dialog.

The listening, thinking and speaking state transitions, relevant to dialog interactions with a user, should be accompanied with a multi-assistant distinct visual and sound cue such that the user is always aware of when they may speak or await a response or action from the voice assistant. This combination of dialog state transitions and cues provides for a more seamless and natural interaction with voice assistants. When multiple voice assistants are involved, a capability for coordinating cues and dialog transitions is especially important to avoid confusing the user.

Natural Turn-Taking

Turn-taking allows for natural and predictable interactions and also avoids incomprehensible speech that may occur when multiple voice assistants speak at the same time. A basic turn-taking dialog between a user and a voice assistant is illustrated in Figure 3, where events

progress in a left-to-right, top-down sequence. The dialog begins with the user invoking a voice assistant by speaking its wake phrase or pressing its voice command button, and the voice assistant then entering its listening state where it is actively capturing audio. The user is notified that the voice assistant has entered its listening state through a distinct visual, such as a unique color and pattern that is rendered on the device's attention system, and a unique sound cue emitted through the device's speaker. The user then speaks their request, inquiry or command (i.e., the utterance) while the voice assistant processes it. Upon realizing the user has finished speaking their utterance, the voice assistant notifies the user that it is no longer capturing audio and renders a unique visual indicating that it is processing the audio to generate a response (a.k.a. the thinking state) and plays a distinct sound. Once the response is generated, a unique speaking visual is rendered, a distinct sound cue may be played, and the voice assistant's audio response is played through the device's speaker system. This flow can be thought of as a baseline dialog experience between a user and a voice assistant.

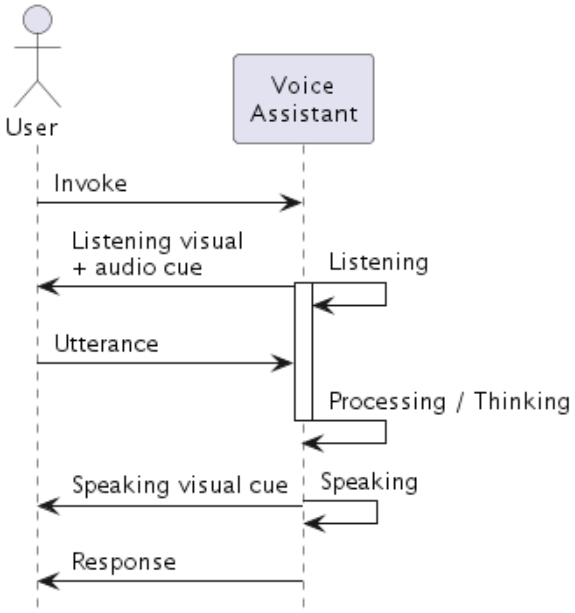

Figure 3 - Turn-Taking Dialog

A voice assistant may require additional clarification to a user request, inquiry or command or may offer an alternative. In this case, instead of providing a direct response, the voice assistant asks the user a question (i.e., it makes an inquiry), to which the user is expected to respond within a certain time period.

A voice assistant may support a follow-up or continued conversation mode, where the microphone reopens briefly after the voice assistant's response, to capture any subsequent utterances from the user. In this follow-up mode, the user does not need to repeat the wake phrase, nor does the voice command button need to be pressed again. The user simply speaks another utterance within a relatively short period of time.

Figure 4 illustrates a Turn-Taking Dialog that incorporates the flow for voice assistant inquiries and Follow-up Mode. Although it shows the user setting Follow-up beforehand, a voice assistant may choose to always be in this mode.

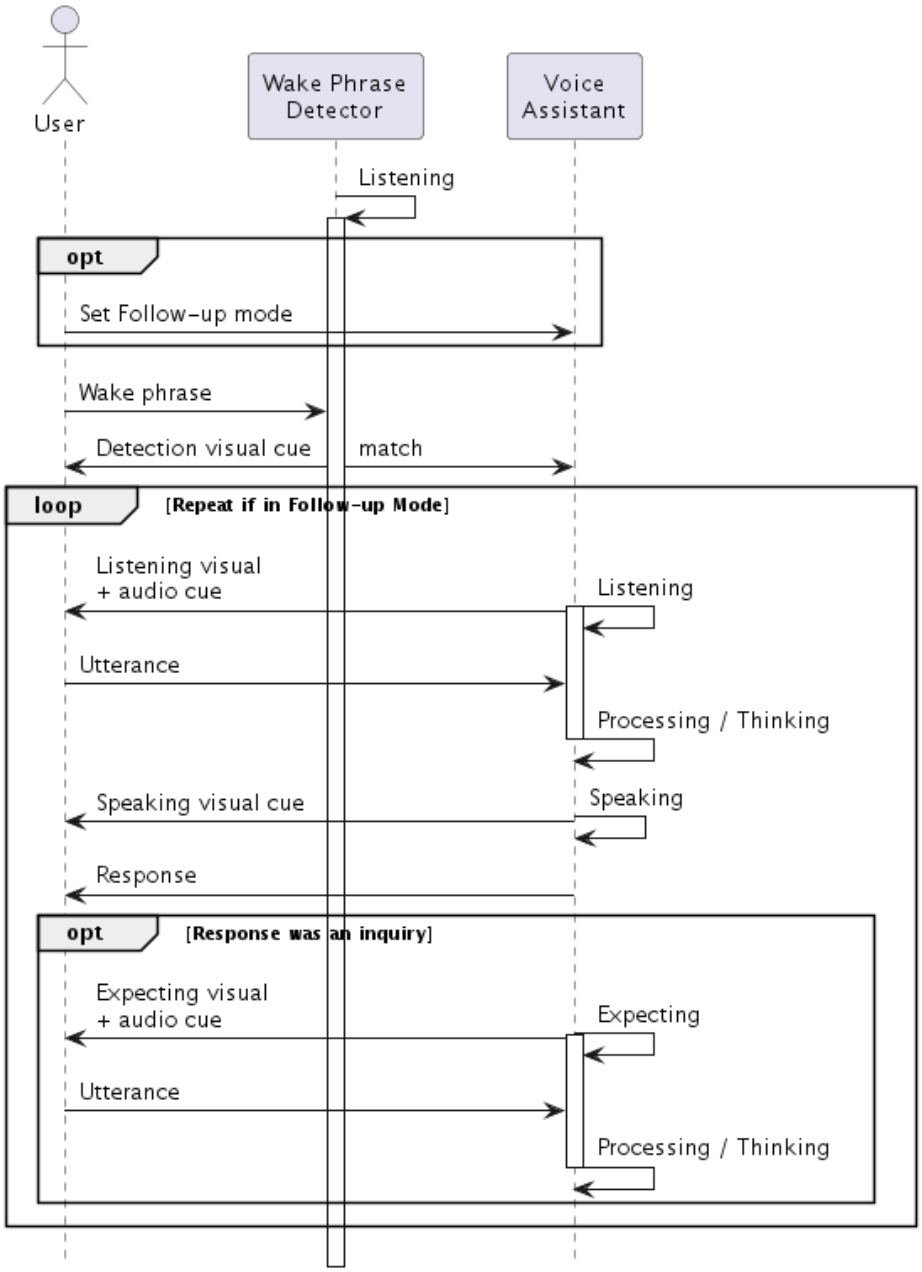

Figure 4 - Turn-Taking Dialog with Inquiry and Follow-up Mode

Follow-up mode is essentially a conversational mode that re-enters the listening state with each iteration of a loop that continues until the user no longer follows up with an utterance, where the voice assistant times out, or explicitly ends the conversation (e.g., via a “stop” command). Whereas, when the voice assistant asks for clarification or offers an alternative with an inquiry,

the user is expected to respond with an utterance, after which the dialog may or may not be complete, depending on whether the alternative or response entails further conversation.

When multiple voice assistants are involved, a mechanism must be available to enforce the negotiated rules required for interruptions, barge-ins, multi-voice assistant cooperation, and so on. However, certain visual and sound cues may continue to be used by voice assistants that are not actively engaged in a dialog to notify the user that a message is pending. In this case, the user may ask the corresponding voice assistant for details on the outstanding message after the current dialog completes. Notifications emphasize the need for having voice assistant-distinct visuals and cues such that the user knows which voice assistant to ask.

Arbitrating Multiple Voice Assistants

Allowing multiple voice assistants to simultaneously respond at the same time on the same device causes confusion and frustration for the user of the system. When a dialog between a person and a voice assistant is underway, designs that minimize the likelihood that another voice assistant inadvertently enters into the dialog or captures audio can minimize customer frustration or confusion. For example, a user may be asking a voice assistant about another voice assistant and refer to its wake phrase. That reference to a wake phrase should not result in the corresponding voice assistant entering into or taking over the existing dialog. Similarly, proactive voice assistants generally should not speak over a user or while another voice assistant is speaking. Speaking over another person or interrupting a conversation is generally considered rude and may result in the person speaking losing their train of thought. Also, when voice assistants speak at the same time it results in garbled speech that is difficult for the user to understand and a poor user experience. Speech etiquette imposes rules of live conversation and correspondence that people are accustomed to and expect when engaging with voice assistants.

A mechanism that guards active dialogs between a user and a voice assistant from vocal intrusions by other voice assistants on the multi-voice assistant device affords a quality user experience. One approach is to implement a voice assistant arbitrator that each voice assistant first inquires with and requests to enter into a dialog with the user. A voice assistant makes this request because it may not have the necessary cues a person may have in a human-to-human conversation to know whether to proceed without interrupting. When a voice assistant's request to enter into a dialog with the user is granted, it proceeds by entering into its listening state to capture input audio for subsequent processing. Whereas, when the request is denied, the user's invocation is suppressed and the voice assistant is unaware that it was invoked. When all voice assistants on the device abide to this mechanism for dialog arbitration, only one voice assistant actively converses in a dialog with the user at any time, thereby avoiding inadvertent vocal interruptions by other voice assistants. Note that voice assistants may elect to play a sound to notify the user that a voice response or proactive speech is pending.

In addition to the need for coordinating conversations between a user and voice assistants, voice assistants may have data rules that govern where and how voice input is routed for

processing on the device or in a cloud. An added benefit of a voice assistant arbitrator is that it helps safeguard both voice input data and a voice assistant's synthetic speech output because this restrains voice assistants from entering a listening or speaking state (i.e., get activated), respectively, upon having gotten approval from the arbitrator to enter into a dialog with the user.

This technique can work well when each voice assistant on the device expects to not be interrupted by other voice assistants or to not be disengaged from a dialog with the user when active. Figure 5 illustrates the request an application program makes with the arbitrator when a voice assistant's wake phrase detection occurs or a proactive voice assistant initiates the dialog with the user. The application program boxes are shown in different colors to convey that they may not be one and the same program. The topmost application program notifies the arbitrator of the invocation type, wake phrase in this case, as a voice assistant's behavior may vary for different invocation types, such as for button initiation.

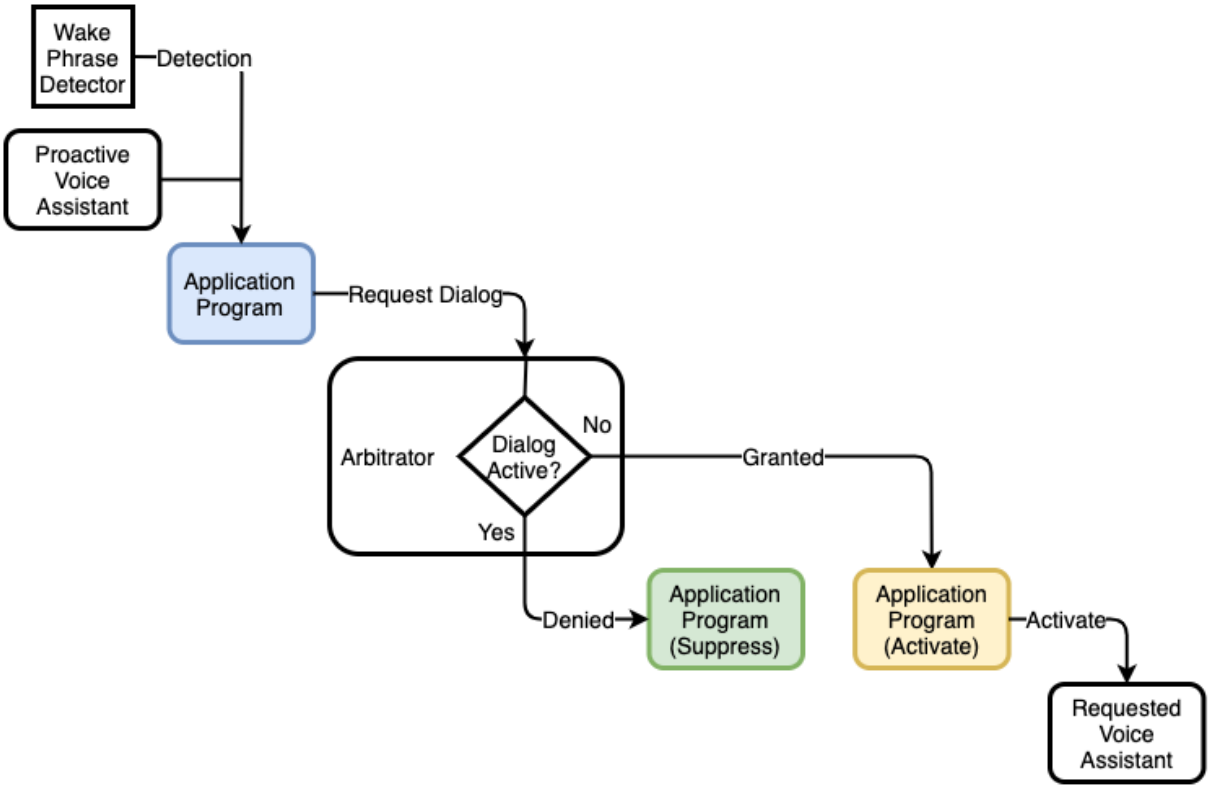

Figure 5 - Request Dialog Flow for a Wake Word or Phrase

These single-turn, dialog-driven interactions provide a somewhat natural and predictable means for communicating with multiple voice assistants. An implication of applying the dialog metaphor is that only one dialog may occur at any point in time on the multi-voice assistant device. The two participants in the dialog are a person and one of the voice assistants. While a voice assistant may indicate through a visual or low-volume sound cue that it has information

for a user of the device, it generally may not verbally interrupt an ongoing dialog. An exception could be a proactive voice assistant that reports or handles life-threatening situations such as for severe weather warnings or a 911 virtual voice assistant.

Another benefit of using a dialog metaphor is it that it provides a means for avoiding invalid activations by another voice assistant when its wake phrase is spoken in the user's request, inquiry or command. For example, a person may ask one voice assistant about another voice assistant and refer to its wake phrase. Normally, speaking that wake phrase would result in the corresponding voice assistant becoming active and the possible unintended termination of the already active voice assistant. However, because the other voice assistant must first request a dialog from the arbitrator, it will be denied entry into the ongoing dialog with the user. Hence, even though the other voice assistant's wake phrase detector triggered, the application program will know to suppress the wake phrase detection and not activate its voice assistant. This is especially beneficial because it results in the suppression of any other voice assistant's wake phrase triggers that may be spoken by other users in the vicinity of the device.

Other means for invoking voice assistants must also be considered. A voice assistant may require that an invocation of its designated button or button pattern (e.g., double press) immediately activate its voice assistant and cease an ongoing dialog's voice assistant. This design decision may be based on the immediacy implied by an inconvenient physical action that may be required by the user to invoke a voice assistant, relative to the simplicity of speaking its wake phrase. For example, the driver of a vehicle may need to inconveniently reach over to the center console in a vehicle to tap on a voice assistant's icon to activate it. Their motivation for doing so may be based on a need for immediate activation of a particular voice assistant, regardless of any state an active voice assistant in a dialog may be in. For example, a user may require immediate confirmation from a navigating voice assistant that an upcoming offramp on a freeway is the correct one. Figure *6* illustrates the role of an arbitrator when the user invokes a voice assistant by touch (e.g., a button-press or screen tap) or a high-priority proactive voice assistant initiates a dialog with the user.

Another aspect requiring consideration is user barge-in. User barge-in is when a user speaks a wake phrase to interrupt a speaking voice assistant. It is especially useful when the voice assistant's verbal response is lengthy or may no longer be of interest. For example, a user may barge-in to terminate a lengthy voice assistant response for a 10-day weather forecast. Some voice assistants may be more restrictive and only allow a user to barge-in with its corresponding wake phrase. Whereas other voice assistants may be flexible in allowing the user to barge-in with any voice assistant's wake phrase. A decision may then need to be made as to whether all voice assistants on the device should comply with a common barge-in policy to provide a more consistent user experience. Otherwise, the user is burdened with having to remember the policy for each voice assistant. For example, if one of the voice assistants limits the user to barge-in with its wake phrase, a common policy could be established and implemented where this applies to all of the voice assistants on the device in order to provide a consistent and predictable customer experience.

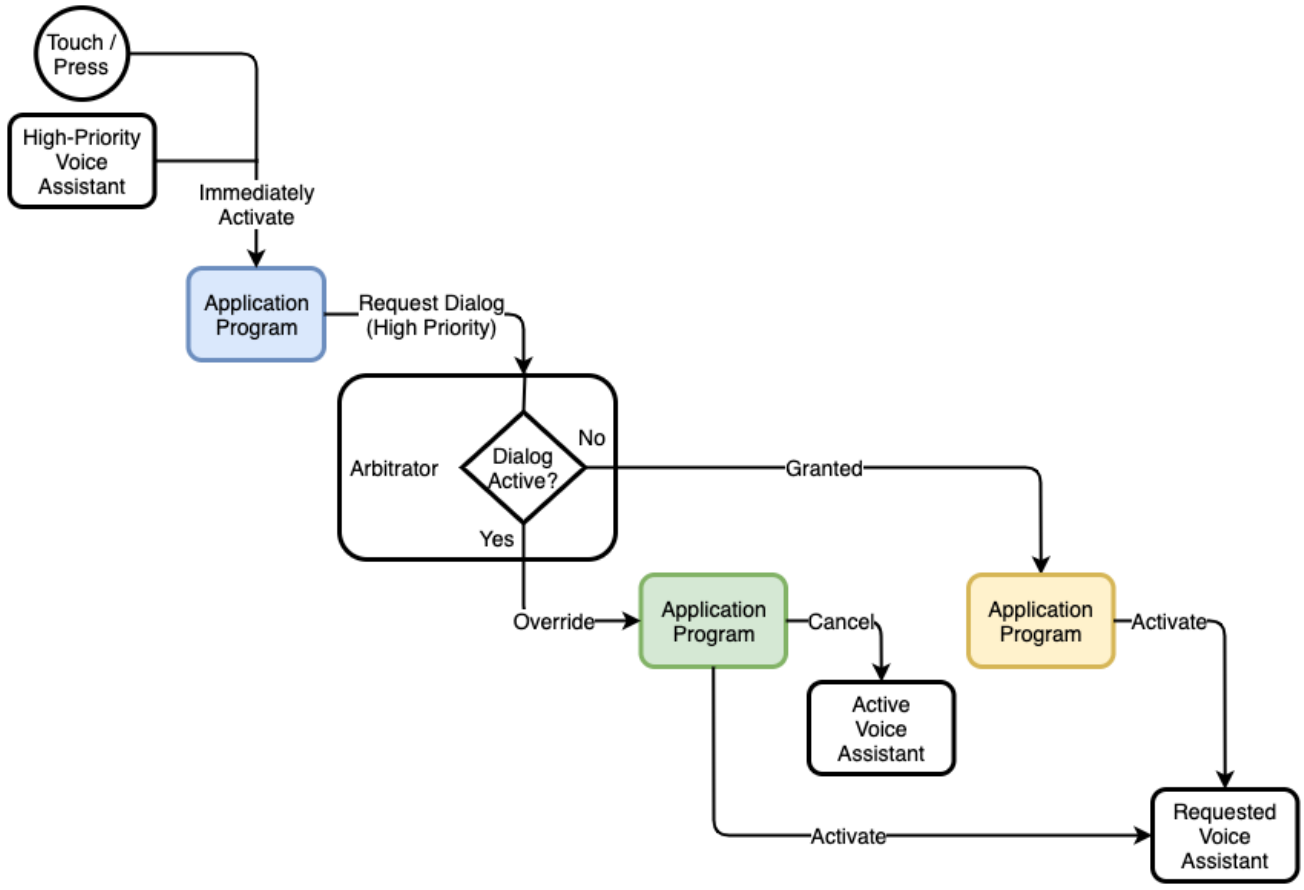

Figure 6 - *Immediate Voice Assistant Activation*

To implement a common barge-in policy, it may be necessary for the application program to notify the arbitrator whenever a voice assistant is in its speaking state such that a particular barge-in policy may apply. For example, while a voice assistant is speaking, it may only allow for a user to barge-in with its wake phrase. In this case, upon receiving a notification that the voice assistant is speaking, the arbitrator would suppress any wake phrase-based dialog requests that it receives from other voice assistants. Then, upon receiving a notification that the speaking voice assistant has returned to its idle state, any further voice assistant dialog requests would be honored so long as a dialog with the user is not already underway. Conversely, if the voice assistant making a dialog request allows for user barge-in with any voice assistant's wake phrase, the arbitrator would grant entry into a dialog with the user and notify the application program to terminate the speaking voice assistant, provided that a common policy limiting user barge-in to the speaking voice assistant's wake phrase has not been put in place.

The arbitrator may be designed to only require notification of the invocation type (e.g., wake phrase or button press) and whether an active voice assistant is transitioning into or out of its speaking state. Or it may be designed to require every voice assistant on the device to first

register its rules for user barge-in and button invocations and then be notified of every voice assistant's state transition to determine whether to grant or deny a dialog with the user. For example, the application program may register one voice assistant with the arbitrator as only allowing for user barge-in with its wake phrase and another as allowing for any wake phrase. It may also register one voice assistant as not requiring immediate termination of an active voice assistant and activation of the newly requested voice assistant when its button is invoked. Then, whenever a user barge-in or button invocation occurs, the arbitrator may analyze the voice assistants rule set to determine the appropriate action and whether to grant or deny a dialog.

Voice assistants may have different rules of engagement for various parts of a dialog interaction. These rules of engagement may also differ for the different types of voice input modalities such as a wake phrase or a button press and may also depend on the various interaction states of voice assistants. Table 2 lists examples of rules that may be passed to a voice assistant arbitrator during application software initialization, similar to the rules that get passed to the voice assistant arbitrator in the Alexa Auto SDK. The set of rules in this example convey that Alexa allows for a user to barge-in with any voice assistant's wake phrase and that a button-based invocation is to immediately cease the currently active voice assistant and activate Alexa. Whereas, for a hypothetical Brandon voice assistant, only its wake phrase may be used to barge-in and a button-based invocation is to be accepted only when it is in the idle state. To provide a more consistent and predictable customer experience, the arbitrator may then universally restrict all user barge-ins for all voice assistants to the speaking voice assistant's wake phrase.

Voice Assistant	Barge-In Rule	Button Rule
Alexa	Any wake phrase	Any state
Brandon	My wake phrase	Idle state

Table 2 – Arbitrator Rules

Additional rules may be specified for other scenarios. For example, a proactive voice assistant that services high-priority events may require precedence in certain states. In this case, it may specify rules to limit user barge-in to its own wake phrase while it is speaking and immediately end a dialog with another voice assistant when its button is pressed or when it, instead of the user, proactively barges in.

A benefit of having a central voice assistant arbitrator is that it alleviates the need for voice assistants to know about one another or having to coordinate directly with each other. Also, a central arbitrator may reside in middleware that is impartial to all of the voice assistants on the device and not governed by or biased towards any particular voice assistant. The use of a centralized arbitrator that accepts rules for state-related behavior depends on each voice assistant providing the rules when the voice assistant is enabled. It also requires the voice assistant's application program request, from the arbitrator, entry into a dialog with the user

whenever the voice assistant is invoked and also indicate the type of invocation. Upon having been granted entry into a dialog with the user, only then does the application program signal the voice assistant to activate. Following this request-first arbitration technique provides for a harmonious multi-voice assistant user experience.

The arbitration behavior should remain consistent, regardless of the number of voice assistants that have been onboarded or enabled on the device to ensure a consistent user experience. Hence, barge-in and button behavior rules for the voice assistants that utilize the arbitrator should be pre-configured on the device before onboarding or enablement. Otherwise, it may result in a change in the barge-in or button behavior of a voice assistant when a previously dormant voice assistant is onboarded or enabled, which will result in a confusing user experience.

A voice assistant arbitrator has two primary roles:

1. To provide a first-come, first served granting mechanism that suppresses subsequent requests by other voice assistants, however close in time, until the granted voice assistant indicates it is no longer active, and
2. To prevent other voice assistants from speaking over an ongoing dialog between the user and a voice assistant.

Generally, two or more invocations of different voice assistants will not occur very close in time. When they do, it should always be the case that the first request the arbitrator receives for a dialog matches the user's intended activation of a voice assistant. For example, if the user was to say "Alexa, Hey Siri …" and the arbitrator receives the request for a dialog from the Alexa application program first, then activation of Alexa is as intended. "Hey Siri" in this case should not and does not activate Siri because the arbitrator denies the request on behalf of a dialog already being active (that for Alexa). Whereas, if "Hey Siri" was supported by a different wake phrase detector and unexpectedly, the arbitrator receives a request from Siri's application program for a dialog first, then Siri will activate and Alexa will be blocked from being activated. However, this is a highly unlikely scenario because it would require that the two voice assistants each have a different wake phrase detector and that Siri's application program and its wake phrase detector either had operating system precedence over Alexa's or was able to process a detection and make the dialog request much faster. In this case, the device maker would need to adjust the preset priorities of the two application programs or wake phrase detectors to be equitable, or introduce software to account for multiple wake phrase detections that occur within very small time periods to address the performance disparity in the detectors. Using a common wake phrase detector can help ensure uniformity in responsiveness to activation triggers.

Operating System-Based Arbitration

A multiple voice assistant device may embed operating system (OS) middleware that implicitly enforces voice assistant arbitration. For example, applications and their respective voice assistants may run in their own sandbox such as on Android, where they may execute with a

protected file system and do not have visibility into other application programs on the device. In this case, the OS and its middleware services may control whether an app (i.e., application program), and hence its voice assistant, becomes active by bringing it to the foreground based on events occurring on the device, such as a user tapping on the application's voice assistant activation icon or by speaking its wake phrase. An active voice assistant can be interrupted by an event, such as the user starting another app, or arbitration mechanism in the OS that invokes another voice assistant, where the active voice assistant's app is then backgrounded and the invoked voice assistant's app is brought to the foreground. The foregrounding and backgrounding of applications, and hence voice assistants, may be on behalf of the user or extraneous events such as an incoming phone call. In these situations, the OS middleware approves or denies app requests for foregrounding and the related services determine which app gets access to system resources such as the microphones. The related policies are also customizable by the platform integrator.

Applications may have a special role such as for an accessibility voice assistant, which may need to be treated with a higher priority. Similarly, platform suppliers and third-party providers may provide voice assistant applications that have special requirements to prioritize one request over another depending on the system state. Software developers may implement inter-process communication or utilize messaging capabilities, such as BroadcastReceiver on Android, to exchange messages across applications through a central arbitrator, middleware or service. In these systems, voice assistant applications may register with a centralized middleware service. This registration process helps the middleware service learn each registered application's role in the system.

These two mechanisms may be implemented as part of the operating system running on the device or by an external s/w component. OS-level implementations are still a relatively new thing. However, with the recent releases of popular mobile OSes, we already see enhancements allowing the installation of multiple applications that register themselves in the system as voice assistants and where the user can select one of them as default VA. Arbitration components from third-party software vendors can provide a means for deeper integration with specific types of devices, such as IoT or automotive systems, and vendor-specific technologies like Android Auto and CarPlay, as well as with an external voice assistant running on a connected smartphone. For the latter part, such components may also facilitate implementing vendor-specific requirements and guidelines for integrating their voice assistants in multi-VA environments. On the other hand, the usage of such 3rd-party components means that they should either come pre-installed on the device or be installed with the applications that depend on them. It should also be noted that the arbitration will only be possible among the voice assistants integrated with the same 3rd-party component.

Summary

Voice assistants empower users with a hands-free experience for getting information and performing tasks by simply using their voice. They may also be proactive and initiate dialogs with the user on behalf of prior user requests or settings. Having multiple voice assistants on a single device further empowers users with a broader range of capabilities, flexibility and freedom to choose their preferred voice assistant for certain features and capabilities, and with redundancy to make services more readily available. It also benefits companies with an opportunity for having their own branded voice assistant that offers a unique set of capabilities and extends the capabilities of other voice assistants on the device [6].

Dialog management, cues, and conveying voice assistant states when interacting with multiple voice assistants on a single device is important for providing a consistent, seamless, and easy-to-use and follow customer experience. The more common voice and button activation input modalities of voice assistants were discussed. A voice assistant arbitrator and dialog-request model that enforces natural turn-taking in dialogs with the user was also covered, where different input modalities may specify distinct user barge-in and activation policies. Lastly, an operating system may offer a built-in voice assistant arbitration mechanism that alleviates some of the concerns for coordinating dialogs with voice assistants.

Conclusion

Voice assistants adopt a turn-taking dialog technique when interacting with a person because multi-modal cues are much more limited on devices than are utilized by people when they speak with one another. Modeling turn-taking in dialogs between a user and one of several voice assistants on a device may be achieved through the use of a voice assistant arbitrator that limits interactions to one voice assistant at a time and universal policies for barge in and other means of invocation to help provide a more predictable and consistent experience for the user.

Glossary

Also refer to the Glossary in the Multi-Agent Design Guide [2].

Active Voice Assistant
The voice assistant that is currently in a listening, thinking, or expecting state. Some voice assistants also include the speaking state or may characterize the listening, thinking, and expecting states as recognizing speech.

Wake Phrase
One or more words that must be spoken to “wake up” a specific voice assistant such that it activates and starts listening for speech from the user. E.G., “Hey Google.” When only one word is needed, it is usually referred to as a wake word, wake-up word, hot word, or keyword. E.G., “Alexa.”

Wake Phrase Detector
A software component that listens for a wake phrase and provides an indication (i.e., triggers) when a match occurs. It typically utilizes a trained neural network model to determine whether a wake phrase was detected.

Authors

The authors of this document are:

- Robert Mars, Principal Solutions Architect, Alexa Voice Services, Amazon
- Vladimir Beloborodov, Senior Staff Engineer, Advanced Engineering, Panasonic Automotive Systems
- Senthilnathan Subramanian, Staff Software Engineer, Advanced Engineering, Panasonic Automotive Systems

Additional Resources

1. Voice Interoperability Initiative: https://developer.amazon.com/en-US/alexa/voice-interoperability
2. Multi-Agent Design Guide: https://build.amazonalexadev.com/rs/365-EFI-026/images/VII_Multi_Agent_Design_Guide.pdf
3. VII Architecture Best Practices – Foundational Concepts: https://m.media-amazon.com/images/G/01/vii/VII_Architecture_Best_Practices_Foundational_Concepts_Whitepaper.pdf
4. Multi-Agent Wake Words: https://m.media-amazon.com/images/G/01/vii/VII_Architecture_Series_Whitepapers_-_Multi-Agents_Wake_Word.pdf
5. New Alexa features — Natural turn-taking: https://www.amazon.science/blog/change-to-alexa-wake-word-process-adds-natural-turn-taking
6. For the first time, Amazon enables companies to access Alexa's advanced AI to build their own intelligent assistants with Alexa Custom Assistant; Fiat Chrysler Automobiles is the first Automotive OEM to implement in vehicles: https://developer.amazon.com/en-US/blogs/alexa/alexa-auto/2021/01/Amazon-Announces-Alexa-Custom-Assistant
7. How to use multiple voice assistants on your Galaxy phone | Samsung US: https://www.youtube.com/watch?v=LNjyOcjpy7g