

**(12) CERTIFIED INNOVATION PATENT**  
**(19) AUSTRALIAN PATENT OFFICE**

(11) Application No. **AU 2018100133 C4**

(54) Title  
**Intelligent list reading**

(51) International Patent Classification(s)  
**G06F 3/16** (2006.01) **G06F 17/28** (2006.01)

(21) Application No: **2018100133** (22) Date of Filing: **2018.01.31**

(45) Publication Date: **2018.03.01**

(45) Publication Journal Date: **2018.03.01**

(45) Granted Journal Date: **2018.03.01**

(45) Certified Journal Date: **2018.06.14**

(45) Amended Journal Date: **2019.07.04**

(62) Divisional of:  
**2017100582**

(71) Applicant(s)  
**Apple Inc.**

(72) Inventor(s)  
**BOOKER, Susan L.;KRISHNAN, Murali;WEINBERG, Garrett L.;PIERCY, Aimee**

(74) Agent / Attorney  
**FPA Patent Attorneys Pty Ltd, ANZ Tower 161 Castlereagh Street, Sydney, NSW, 2000,  
AU**

**ABSTRACT**

Systems and processes for operating an intelligent automated assistant to perform intelligent list reading are provided. In one example process, a spoken user request associated with a plurality of data items is received. The process determines whether a degree of specificity of the spoken user request is less than a threshold level. In response to determining that a degree of specificity of the spoken user request is less than a threshold level, one or more attributes related to the spoken user request are determined. The one or more attributes are not defined in the spoken user request. Additionally, a list of data items based on the spoken user request and the one or more attributes is obtained. A spoken response comprising a subset of the list of data items is generated and the spoken response is provided.

# INTELLIGENT LIST READING

## Cross-Reference to Related Applications

[0001] This application claims priority to U.S. Provisional Serial No. 62/346,236, filed on June 6, 2016, entitled “INTELLIGENT LIST READING,” U.S. Non-Provisional Serial No. 15/272,214, filed on September 21, 2016, entitled “INTELLIGENT LIST READING,” and PCT Application Serial No. PCT/US2016/054459, filed on September 29, 2016, entitled “INTELLIGENT LIST READING,” all of which are hereby incorporated by reference in their entirety for all purposes.

## Field

[0002] This relates generally to intelligent automated assistants and, more specifically, to intelligent list reading by intelligent automated assistants.

## Background

[0003] Intelligent automated assistants (or digital assistants) can provide a beneficial interface between human users and electronic devices. Such assistants can allow users to interact with devices or systems using natural language in spoken and/or text forms. For example, a user can provide a speech input containing a user request to a digital assistant operating on an electronic device. The digital assistant can interpret the user’s intent from the speech input and operationalize the user’s intent into tasks. The tasks can then be performed by executing one or more services of the electronic device, and a relevant output responsive to the user request can be returned to the user.

[0004] Interactions with digital assistants can often be voice-based, where the user provides a spoken user request and the digital assistant replies with a spoken response that satisfies the request. However, providing intuitive and natural-sounding voice-based interactions with a digital assistant can be challenging. For example, spoken responses can often contain too little or too much information. Additionally, spoken responses can have awkward transitions and can

force the user into one or more unproductive follow-up interactions. This can impact user experience and hinder the widespread adoption of digital assistants.

**[0004A]** Reference to any prior art in the specification is not an acknowledgment or suggestion that this prior art forms part of the common general knowledge in any jurisdiction or that this prior art could reasonably be expected to be understood, regarded as relevant, and/or combined with other pieces of prior art by a skilled person in the art.

### Summary

**[0004B]** According to a first aspect of the invention there is provided a method for operating a digital assistant, the method comprising: at a portable electronic device with a processor and memory: receiving, from an audio input unit, a spoken user request associated with a plurality of data items; determining whether a degree of specificity of the spoken user request is less than a threshold level; in response to determining that a degree of specificity of the spoken user request is less than a threshold level: determining one or more attributes related to the spoken user request, the one or more attributes not defined in the spoken user request; obtaining a list of data items based on the spoken user request and the one or more attributes; generating a spoken response comprising a subset of the list of data items; and providing, to a speaker unit, the spoken response.

**[0005]** Systems and processes for operating an intelligent automated assistant to perform intelligent list reading are provided. In one example process, a spoken user request associated with a plurality of data items is received. The process determines whether a degree of specificity of the spoken user request is less than a threshold level. In response to determining that the degree of specificity of the spoken user request is less than a threshold level, one or more attributes related to the spoken user request are determined. The one or more attributes are not defined in the spoken user request. Additionally, a list of data items based on the spoken user request and the one or more attributes is obtained. A spoken response comprising a subset of the list of data items is generated and the spoken response is provided.

**[0006]** In some examples, in response to determining that the degree of specificity of the spoken user request is not less than a threshold level, a second list of data items is obtained based on the spoken user request. The process further determines whether the number of data items in

the fourth list of data items exceeds a predetermined number. In response to determining that the number of data items in the fourth list of data items exceeds a predetermined number, a second spoken response comprising a subset of the second list of data items is generated and the second spoken response is provided.

### **Brief Description of the Drawings**

[0007] FIG. 1 is a block diagram illustrating a system and environment for implementing a digital assistant, according to various examples.

[0008] FIG. 2A is a block diagram illustrating a portable multifunction device implementing the client-side portion of a digital assistant, according to various examples.

[0009] FIG. 2B is a block diagram illustrating exemplary components for event handling, according to various examples.

[0010] FIG. 3 illustrates a portable multifunction device implementing the client-side portion of a digital assistant, according to various examples.

[0011] FIG. 4 is a block diagram of an exemplary multifunction device with a display and a touch-sensitive surface, according to various examples.

[0012] FIG. 5A illustrates an exemplary user interface for a menu of applications on a portable multifunction device, according to various examples.

[0013] FIG. 5B illustrates an exemplary user interface for a multifunction device with a touch-sensitive surface that is separate from the display, according to various examples.

[0014] FIG. 6A illustrates a personal electronic device, according to various examples.

[0015] FIG. 6B is a block diagram illustrating a personal electronic device, according to various examples.

[0016] FIG. 7A is a block diagram illustrating a digital assistant system or a server portion thereof, according to various examples.

[0017] FIG. 7B illustrates the functions of the digital assistant shown in FIG. 7A, according to various examples.

[0018] FIG. 7C illustrates a portion of an ontology, according to various examples.

[0019] FIGs. 8A-D illustrate a process for operating a digital assistant to perform intelligent list reading, according to various examples.

[0020] FIGs. 9A-D illustrate intelligent list reading performed by a digital assistant implemented on a user device in response to spoken user requests from a user, according to various examples.

[0021] FIG. 10 illustrates a functional block diagram of an electronic device, according to various examples.

### **Detailed Description**

[0022] In the following description of examples, reference is made to the accompanying drawings in which are shown by way of illustration specific examples that can be practiced. It is to be understood that other examples can be used and structural changes can be made without departing from the scope of the various examples.

[0023] As discussed above, providing natural-sounding voice-based interactions with a digital assistant is challenging. In particular, user requests range from being overly broad (e.g., “Any good places to eat?”) to very specific (e.g., “What is the meaning of the word ‘plot?’”). If all user requests were addressed in the same manner, spoken responses provided by the digital assistant can be awkward and unproductive. For example, in response to the user request “Any good places to eat?” the digital assistant could provide a verbose spoken response containing a long list of restaurants located near the user’s current location. Such a response are overwhelming and unhelpful.

[0024] In accordance with some exemplary systems and processes described herein, the spoken response generated by the digital assistant is intelligently constructed based on the degree of specificity of the spoken user request. In one example process, a spoken user request

associated with a plurality of data items is received. The process determines whether the degree of specificity of the spoken user request is less than a threshold level. In response to determining that the degree of specificity of the spoken user request is less than a threshold level, one or more attributes related to the spoken user request are determined. The one or more attributes are not defined in the spoken user request. In particular, the one or more attributes serve to refine the spoken user request, which may be vague and overly broad. A list of data items based on the spoken user request and the one or more attributes is obtained. A spoken response comprising a subset of the list of data items is generated and the spoken response is provided. The subset provides a useful and targeted recommendation to the user instead of overwhelming the user with a long list of data items.

**[0025]** In some examples, in response to determining that the degree of specificity of the spoken user request is not less than a threshold level, a second list of data items is obtained based on the spoken user request. Because the degree of specificity is not less than the threshold level, the second list of data items contain a limited number of data items. The process determines whether the number of data items in the second list of data items exceeds a predetermined number. The predetermined number is determined based on various factors, such as the information density of each data item in the second list of data items, the user's familiarity with each data item in the second list of data items, and a cognitive load associated with the user. In response to determining that the number of data items in the second list of data items exceeds a predetermined number, a second spoken response comprising a subset of the second list of data items is generated and the second spoken response is provided. In this way, the second list of data items is presented to the user in smaller groups that is more easily processed and retained by the user.

**[0026]** Although the following description uses terms "first," "second," etc. to describe various elements, these elements should not be limited by the terms. These terms are only used to distinguish one element from another. For example, a first input could be termed a second input, and, similarly, a second input could be termed a first input, without departing from the scope of the various described examples. The first input and the second input are both inputs and, in some cases, are separate and different inputs.

**[0027]** The terminology used in the description of the various described examples herein is for the purpose of describing particular examples only and is not intended to be limiting. As used in the description of the various described examples and the appended claims, the singular forms “a,” “an,” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will also be understood that the term “and/or” as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will be further understood that the terms “includes,” “including,” “comprises,” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

**[0028]** The term “if” may be construed to mean “when” or “upon” or “in response to determining” or “in response to detecting,” depending on the context. Similarly, the phrase “if it is determined” or “if [a stated condition or event] is detected” may be construed to mean “upon determining” or “in response to determining” or “upon detecting [the stated condition or event]” or “in response to detecting [the stated condition or event],” depending on the context.

#### 1. System and Environment

**[0029]** FIG. 1 illustrates a block diagram of system 100 according to various examples. In some examples, system 100 implements a digital assistant. The terms “digital assistant,” “virtual assistant,” “intelligent automated assistant,” or “automatic digital assistant” refer to any information processing system that interprets natural language input in spoken and/or textual form to infer user intent, and performs actions based on the inferred user intent. For example, to act on an inferred user intent, the system performs one or more of the following: identifying a task flow with steps and parameters designed to accomplish the inferred user intent, inputting specific requirements from the inferred user intent into the task flow; executing the task flow by invoking programs, methods, services, APIs, or the like; and generating output responses to the user in an audible (e.g., speech) and/or visual form.

**[0030]** Specifically, a digital assistant is capable of accepting a user request at least partially in the form of a natural language command, request, statement, narrative, and/or inquiry.



Typically, the user request seeks either an informational answer or performance of a task by the digital assistant. A satisfactory response to the user request includes a provision of the requested informational answer, a performance of the requested task, or a combination of the two. For example, a user asks the digital assistant a question, such as “Where am I right now?” Based on the user’s current location, the digital assistant answers, “You are in Central Park near the west gate.” The user also requests the performance of a task, for example, “Please invite my friends to my girlfriend’s birthday party next week.” In response, the digital assistant can acknowledge the request by saying “Yes, right away,” and then send a suitable calendar invite on behalf of the user to each of the user’s friends listed in the user’s electronic address book. During performance of a requested task, the digital assistant sometimes interacts with the user in a continuous dialogue involving multiple exchanges of information over an extended period of time. There are numerous other ways of interacting with a digital assistant to request information or performance of various tasks. In addition to providing verbal responses and taking programmed actions, the digital assistant also provides responses in other visual or audio forms, e.g., as text, alerts, music, videos, animations, etc.

**[0031]** As shown in FIG. 1, in some examples, a digital assistant is implemented according to a client-server model. The digital assistant includes client-side portion 102 (hereafter “DA client 102”) executed on user device 104 and server-side portion 106 (hereafter “DA server 106”) executed on server system 108. DA client 102 communicates with DA server 106 through one or more networks 110. DA client 102 provides client-side functionalities such as user-facing input and output processing and communication with DA server 106. DA server 106 provides server-side functionalities for any number of DA clients 102 each residing on a respective user device 104.

**[0032]** In some examples, DA server 106 includes client-facing I/O interface 112, one or more processing modules 114, data and models 116, and I/O interface to external services 118. The client-facing I/O interface 112 facilitates the client-facing input and output processing for DA server 106. One or more processing modules 114 utilize data and models 116 to process speech input and determine the user’s intent based on natural language input. Further, one or more processing modules 114 perform task execution based on inferred user intent. In some examples, DA server 106 communicates with external services 120 through network(s) 110 for

task completion or information acquisition. I/O interface to external services 118 facilitates such communications.

**[0033]** User device 104 can be any suitable electronic device. In some examples, user device is a portable multifunctional device (e.g., device 200, described below with reference to FIG. 2A), a multifunctional device (e.g., device 400, described below with reference to FIG. 4), or a personal electronic device (e.g., device 600, described below with reference to FIG. 6A-B.) A portable multifunctional device is, for example, a mobile telephone that also contains other functions, such as PDA and/or music player functions. Specific examples of portable multifunction devices include the iPhone®, iPod Touch®, and iPad® devices from Apple Inc. of Cupertino, California. Other examples of portable multifunction devices include, without limitation, laptop or tablet computers. Further, in some examples, user device 104 is a non-portable multifunctional device. In particular, user device 104 is a desktop computer, a game console, a television, or a television set-top box. In some examples, user device 104 includes a touch-sensitive surface (e.g., touch screen displays and/or touchpads). Further, user device 104 optionally includes one or more other physical user-interface devices, such as a physical keyboard, a mouse, and/or a joystick. Various examples of electronic devices, such as multifunctional devices, are described below in greater detail.

**[0034]** Examples of communication network(s) 110 include local area networks (LAN) and wide area networks (WAN), e.g., the Internet. Communication network(s) 110 is implemented using any known network protocol, including various wired or wireless protocols, such as, for example, Ethernet, Universal Serial Bus (USB), FIREWIRE, Global System for Mobile Communications (GSM), Enhanced Data GSM Environment (EDGE), code division multiple access (CDMA), time division multiple access (TDMA), Bluetooth, Wi-Fi, voice over Internet Protocol (VoIP), Wi-MAX, or any other suitable communication protocol.

**[0035]** Server system 108 is implemented on one or more standalone data processing apparatus or a distributed network of computers. In some examples, server system 108 also employs various virtual devices and/or services of third-party service providers (e.g., third-party cloud service providers) to provide the underlying computing resources and/or infrastructure resources of server system 108.

**[0036]** In some examples, user device 104 communicates with DA server 106 via second user device 122. Second user device 122 is similar or identical to user device 104. For example, second user device 122 is similar to devices 200, 400, or 600 described below with reference to FIGs. 2A, 4, and 6A-B. User device 104 is configured to communicatively couple to second user device 122 via a direct communication connection, such as Bluetooth, NFC, BTLE, or the like, or via a wired or wireless network, such as a local Wi-Fi network. In some examples, second user device 122 is configured to act as a proxy between user device 104 and DA server 106. For example, DA client 102 of user device 104 is configured to transmit information (e.g., a user request received at user device 104) to DA server 106 via second user device 122. DA server 106 processes the information and return relevant data (e.g., data content responsive to the user request) to user device 104 via second user device 122.

**[0037]** In some examples, user device 104 is configured to communicate abbreviated requests for data to second user device 122 to reduce the amount of information transmitted from user device 104. Second user device 122 is configured to determine supplemental information to add to the abbreviated request to generate a complete request to transmit to DA server 106. This system architecture can advantageously allow user device 104 having limited communication capabilities and/or limited battery power (e.g., a watch or a similar compact electronic device) to access services provided by DA server 106 by using second user device 122, having greater communication capabilities and/or battery power (e.g., a mobile phone, laptop computer, tablet computer, or the like), as a proxy to DA server 106. While only two user devices 104 and 122 are shown in FIG. 1, it should be appreciated that system 100, in some examples, includes any number and type of user devices configured in this proxy configuration to communicate with DA server system 106.

**[0038]** Although the digital assistant shown in FIG. 1 includes both a client-side portion (e.g., DA client 102) and a server-side portion (e.g., DA server 106), in some examples, the functions of a digital assistant are implemented as a standalone application installed on a user device. In addition, the divisions of functionalities between the client and server portions of the digital assistant can vary in different implementations. For instance, in some examples, the DA client is a thin-client that provides only user-facing input and output processing functions, and delegates all other functionalities of the digital assistant to a backend server.

## 2. Electronic Devices

**[0039]** Attention is now directed toward embodiments of electronic devices for implementing the client-side portion of a digital assistant. FIG. 2A is a block diagram illustrating portable multifunction device 200 with touch-sensitive display system 212 in accordance with some embodiments. Touch-sensitive display 212 is sometimes called a “touch screen” for convenience and is sometimes known as or called a “touch-sensitive display system.” Device 200 includes memory 202 (which optionally includes one or more computer-readable storage mediums), memory controller 222, one or more processing units (CPUs) 220, peripherals interface 218, RF circuitry 208, audio circuitry 210, speaker 211, microphone 213, input/output (I/O) subsystem 206, other input control devices 216, and external port 224. Device 200 optionally includes one or more optical sensors 264. Device 200 optionally includes one or more contact intensity sensors 265 for detecting intensity of contacts on device 200 (e.g., a touch-sensitive surface such as touch-sensitive display system 212 of device 200). Device 200 optionally includes one or more tactile output generators 267 for generating tactile outputs on device 200 (e.g., generating tactile outputs on a touch-sensitive surface such as touch-sensitive display system 212 of device 200 or touchpad 455 of device 400). These components optionally communicate over one or more communication buses or signal lines 203.

**[0040]** As used in the specification and claims, the term “intensity” of a contact on a touch-sensitive surface refers to the force or pressure (force per unit area) of a contact (e.g., a finger contact) on the touch-sensitive surface, or to a substitute (proxy) for the force or pressure of a contact on the touch-sensitive surface. The intensity of a contact has a range of values that includes at least four distinct values and more typically includes hundreds of distinct values (e.g., at least 256). Intensity of a contact is, optionally, determined (or measured) using various approaches and various sensors or combinations of sensors. For example, one or more force sensors underneath or adjacent to the touch-sensitive surface are, optionally, used to measure force at various points on the touch-sensitive surface. In some implementations, force measurements from multiple force sensors are combined (e.g., a weighted average) to determine an estimated force of a contact. Similarly, a pressure-sensitive tip of a stylus is, optionally, used to determine a pressure of the stylus on the touch-sensitive surface. Alternatively, the size of the contact area detected on the touch-sensitive surface and/or changes thereto, the capacitance of

the touch-sensitive surface proximate to the contact and/or changes thereto, and/or the resistance of the touch-sensitive surface proximate to the contact and/or changes thereto are, optionally, used as a substitute for the force or pressure of the contact on the touch-sensitive surface. In some implementations, the substitute measurements for contact force or pressure are used directly to determine whether an intensity threshold has been exceeded (e.g., the intensity threshold is described in units corresponding to the substitute measurements). In some implementations, the substitute measurements for contact force or pressure are converted to an estimated force or pressure, and the estimated force or pressure is used to determine whether an intensity threshold has been exceeded (e.g., the intensity threshold is a pressure threshold measured in units of pressure). Using the intensity of a contact as an attribute of a user input allows for user access to additional device functionality that may otherwise not be accessible by the user on a reduced-size device with limited real estate for displaying affordances (e.g., on a touch-sensitive display) and/or receiving user input (e.g., via a touch-sensitive display, a touch-sensitive surface, or a physical/mechanical control such as a knob or a button).

**[0041]** As used in the specification and claims, the term “tactile output” refers to physical displacement of a device relative to a previous position of the device, physical displacement of a component (e.g., a touch-sensitive surface) of a device relative to another component (e.g., housing) of the device, or displacement of the component relative to a center of mass of the device that will be detected by a user with the user’s sense of touch. For example, in situations where the device or the component of the device is in contact with a surface of a user that is sensitive to touch (e.g., a finger, palm, or other part of a user’s hand), the tactile output generated by the physical displacement will be interpreted by the user as a tactile sensation corresponding to a perceived change in physical characteristics of the device or the component of the device. For example, movement of a touch-sensitive surface (e.g., a touch-sensitive display or trackpad) is, optionally, interpreted by the user as a “down click” or “up click” of a physical actuator button. In some cases, a user will feel a tactile sensation such as an “down click” or “up click” even when there is no movement of a physical actuator button associated with the touch-sensitive surface that is physically pressed (e.g., displaced) by the user’s movements. As another example, movement of the touch-sensitive surface is, optionally, interpreted or sensed by the user as “roughness” of the touch-sensitive surface, even when there is no change in smoothness of the touch-sensitive surface. While such interpretations of touch by a user will be subject to

the individualized sensory perceptions of the user, there are many sensory perceptions of touch that are common to a large majority of users. Thus, when a tactile output is described as corresponding to a particular sensory perception of a user (e.g., an “up click,” a “down click,” “roughness”), unless otherwise stated, the generated tactile output corresponds to physical displacement of the device or a component thereof that will generate the described sensory perception for a typical (or average) user.

**[0042]** It should be appreciated that device 200 is only one example of a portable multifunction device, and that device 200 optionally has more or fewer components than shown, optionally combines two or more components, or optionally has a different configuration or arrangement of the components. The various components shown in FIG. 2A are implemented in hardware, software, or a combination of both hardware and software, including one or more signal processing and/or application-specific integrated circuits.

**[0043]** Memory 202 includes one or more computer-readable storage mediums. The computer-readable storage mediums are, for example, tangible and non-transitory. Memory 202 includes high-speed random access memory and also includes non-volatile memory, such as one or more magnetic disk storage devices, flash memory devices, or other non-volatile solid-state memory devices. Memory controller 222 controls access to memory 202 by other components of device 200.

**[0044]** In some examples, a non-transitory computer-readable storage medium of memory 202 is used to store instructions (e.g., for performing aspects of processes described below) for use by or in connection with an instruction execution system, apparatus, or device, such as a computer-based system, processor-containing system, or other system that can fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions. In other examples, the instructions (e.g., for performing aspects of the processes described below) are stored on a non-transitory computer-readable storage medium (not shown) of the server system 108 or are divided between the non-transitory computer-readable storage medium of memory 202 and the non-transitory computer-readable storage medium of server system 108.

**[0045]** Peripherals interface 218 is used to couple input and output peripherals of the device to CPU 220 and memory 202. The one or more processors 220 run or execute various software programs and/or sets of instructions stored in memory 202 to perform various functions for device 200 and to process data. In some embodiments, peripherals interface 218, CPU 220, and memory controller 222 are implemented on a single chip, such as chip 204. In some other embodiments, they are implemented on separate chips.

**[0046]** RF (radio frequency) circuitry 208 receives and sends RF signals, also called electromagnetic signals. RF circuitry 208 converts electrical signals to/from electromagnetic signals and communicates with communications networks and other communications devices via the electromagnetic signals. RF circuitry 208 optionally includes well-known circuitry for performing these functions, including but not limited to an antenna system, an RF transceiver, one or more amplifiers, a tuner, one or more oscillators, a digital signal processor, a CODEC chipset, a subscriber identity module (SIM) card, memory, and so forth. RF circuitry 208 optionally communicates with networks, such as the Internet, also referred to as the World Wide Web (WWW), an intranet and/or a wireless network, such as a cellular telephone network, a wireless local area network (LAN) and/or a metropolitan area network (MAN), and other devices by wireless communication. The RF circuitry 208 optionally includes well-known circuitry for detecting near field communication (NFC) fields, such as by a short-range communication radio. The wireless communication optionally uses any of a plurality of communications standards, protocols, and technologies, including but not limited to Global System for Mobile Communications (GSM), Enhanced Data GSM Environment (EDGE), high-speed downlink packet access (HSDPA), high-speed uplink packet access (HSUPA), Evolution, Data-Only (EV-DO), HSPA, HSPA+, Dual-Cell HSPA (DC-HSPDA), long term evolution (LTE), near field communication (NFC), wideband code division multiple access (W-CDMA), code division multiple access (CDMA), time division multiple access (TDMA), Bluetooth, Bluetooth Low Energy (BTLE), Wireless Fidelity (Wi-Fi) (e.g., IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, IEEE 802.11n, and/or IEEE 802.11ac), voice over Internet Protocol (VoIP), Wi-MAX, a protocol for e mail (e.g., Internet message access protocol (IMAP) and/or post office protocol (POP)), instant messaging (e.g., extensible messaging and presence protocol (XMPP), Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE), Instant Messaging and Presence Service (IMPS)), and/or Short Message Service (SMS), or any other

suitable communication protocol, including communication protocols not yet developed as of the filing date of this document.

**[0047]** Audio circuitry 210, speaker 211, and microphone 213 provide an audio interface between a user and device 200. Audio circuitry 210 receives audio data from peripherals interface 218, converts the audio data to an electrical signal, and transmits the electrical signal to speaker 211. Speaker 211 converts the electrical signal to human-audible sound waves. Audio circuitry 210 also receives electrical signals converted by microphone 213 from sound waves. Audio circuitry 210 converts the electrical signal to audio data and transmits the audio data to peripherals interface 218 for processing. Audio data are retrieved from and/or transmitted to memory 202 and/or RF circuitry 208 by peripherals interface 218. In some embodiments, audio circuitry 210 also includes a headset jack (e.g., 312, FIG. 3). The headset jack provides an interface between audio circuitry 210 and removable audio input/output peripherals, such as output-only headphones or a headset with both output (e.g., a headphone for one or both ears) and input (e.g., a microphone).

**[0048]** I/O subsystem 206 couples input/output peripherals on device 200, such as touch screen 212 and other input control devices 216, to peripherals interface 218. I/O subsystem 206 optionally includes display controller 256, optical sensor controller 258, intensity sensor controller 259, haptic feedback controller 261, and one or more input controllers 260 for other input or control devices. The one or more input controllers 260 receive/send electrical signals from/to other input control devices 216. The other input control devices 216 optionally include physical buttons (e.g., push buttons, rocker buttons, etc.), dials, slider switches, joysticks, click wheels, and so forth. In some alternate embodiments, input controller(s) 260 are, optionally, coupled to any (or none) of the following: a keyboard, an infrared port, a USB port, and a pointer device such as a mouse. The one or more buttons (e.g., 308, FIG. 3) optionally include an up/down button for volume control of speaker 211 and/or microphone 213. The one or more buttons optionally include a push button (e.g., 306, FIG. 3).

**[0049]** A quick press of the push button disengages a lock of touch screen 212 or begin a process that uses gestures on the touch screen to unlock the device, as described in U.S. Patent Application 11/322,549, "Unlocking a Device by Performing Gestures on an Unlock Image,"



filed December 23, 2005, U.S. Pat. No. 7,657,849, which is hereby incorporated by reference in its entirety. A longer press of the push button (e.g., 306) turns power to device 200 on or off. The user is able to customize a functionality of one or more of the buttons. Touch screen 212 is used to implement virtual or soft buttons and one or more soft keyboards.

**[0050]** Touch-sensitive display 212 provides an input interface and an output interface between the device and a user. Display controller 256 receives and/or sends electrical signals from/to touch screen 212. Touch screen 212 displays visual output to the user. The visual output includes graphics, text, icons, video, and any combination thereof (collectively termed “graphics”). In some embodiments, some or all of the visual output correspond to user-interface objects.

**[0051]** Touch screen 212 has a touch-sensitive surface, sensor, or set of sensors that accepts input from the user based on haptic and/or tactile contact. Touch screen 212 and display controller 256 (along with any associated modules and/or sets of instructions in memory 202) detect contact (and any movement or breaking of the contact) on touch screen 212 and convert the detected contact into interaction with user-interface objects (e.g., one or more soft keys, icons, web pages, or images) that are displayed on touch screen 212. In an exemplary embodiment, a point of contact between touch screen 212 and the user corresponds to a finger of the user.

**[0052]** Touch screen 212 uses LCD (liquid crystal display) technology, LPD (light emitting polymer display) technology, or LED (light emitting diode) technology, although other display technologies may be used in other embodiments. Touch screen 212 and display controller 256 detect contact and any movement or breaking thereof using any of a plurality of touch sensing technologies now known or later developed, including but not limited to capacitive, resistive, infrared, and surface acoustic wave technologies, as well as other proximity sensor arrays or other elements for determining one or more points of contact with touch screen 212. In an exemplary embodiment, projected mutual capacitance sensing technology is used, such as that found in the iPhone® and iPod Touch® from Apple Inc. of Cupertino, California.

**[0053]** A touch-sensitive display in some embodiments of touch screen 212 is analogous to the multi-touch sensitive touchpads described in the following U.S. Patents: 6,323,846

(Westerman et al.), 6,570,557 (Westerman et al.), and/or 6,677,932 (Westerman), and/or U.S. Patent Publication 2002/0015024A1, each of which is hereby incorporated by reference in its entirety. However, touch screen 212 displays visual output from device 200, whereas touch-sensitive touchpads do not provide visual output.

**[0054]** A touch-sensitive display in some embodiments of touch screen 212 is as described in the following applications: (1) U.S. Patent Application No. 11/381,313, “Multipoint Touch Surface Controller,” filed May 2, 2006; (2) U.S. Patent Application No. 10/840,862, “Multipoint Touchscreen,” filed May 6, 2004; (3) U.S. Patent Application No. 10/903,964, “Gestures For Touch Sensitive Input Devices,” filed July 30, 2004; (4) U.S. Patent Application No. 11/048,264, “Gestures For Touch Sensitive Input Devices,” filed January 31, 2005; (5) U.S. Patent Application No. 11/038,590, “Mode-Based Graphical User Interfaces For Touch Sensitive Input Devices,” filed January 18, 2005; (6) U.S. Patent Application No. 11/228,758, “Virtual Input Device Placement On A Touch Screen User Interface,” filed September 16, 2005; (7) U.S. Patent Application No. 11/228,700, “Operation Of A Computer With A Touch Screen Interface,” filed September 16, 2005; (8) U.S. Patent Application No. 11/228,737, “Activating Virtual Keys Of A Touch-Screen Virtual Keyboard,” filed September 16, 2005; and (9) U.S. Patent Application No. 11/367,749, “Multi-Functional Hand-Held Device,” filed March 3, 2006. All of these applications are incorporated by reference herein in their entirety.

**[0055]** Touch screen 212 has, for example, a video resolution in excess of 100 dpi. In some embodiments, the touch screen has a video resolution of approximately 160 dpi. The user makes contact with touch screen 212 using any suitable object or appendage, such as a stylus, a finger, and so forth. In some embodiments, the user interface is designed to work primarily with finger-based contacts and gestures, which can be less precise than stylus-based input due to the larger area of contact of a finger on the touch screen. In some embodiments, the device translates the rough finger-based input into a precise pointer/cursor position or command for performing the actions desired by the user.

**[0056]** In some embodiments, in addition to the touch screen, device 200 includes a touchpad (not shown) for activating or deactivating particular functions. In some embodiments, the touchpad is a touch-sensitive area of the device that, unlike the touch screen, does not display

visual output. The touchpad is a touch-sensitive surface that is separate from touch screen 212 or an extension of the touch-sensitive surface formed by the touch screen.

**[0057]** Device 200 also includes power system 262 for powering the various components. Power system 262 includes a power management system, one or more power sources (e.g., battery, alternating current (AC)), a recharging system, a power failure detection circuit, a power converter or inverter, a power status indicator (e.g., a light-emitting diode (LED)) and any other components associated with the generation, management and distribution of power in portable devices.

**[0058]** Device 200 also includes one or more optical sensors 264. FIG. 2A shows an optical sensor coupled to optical sensor controller 258 in I/O subsystem 206. Optical sensor 264 includes charge-coupled device (CCD) or complementary metal-oxide semiconductor (CMOS) phototransistors. Optical sensor 264 receives light from the environment, projected through one or more lenses, and converts the light to data representing an image. In conjunction with imaging module 243 (also called a camera module), optical sensor 264 captures still images or video. In some embodiments, an optical sensor is located on the back of device 200, opposite touch screen display 212 on the front of the device so that the touch screen display is used as a viewfinder for still and/or video image acquisition. In some embodiments, an optical sensor is located on the front of the device so that the user's image is obtained for video conferencing while the user views the other video conference participants on the touch screen display. In some embodiments, the position of optical sensor 264 can be changed by the user (e.g., by rotating the lens and the sensor in the device housing) so that a single optical sensor 264 is used along with the touch screen display for both video conferencing and still and/or video image acquisition.

**[0059]** Device 200 optionally also includes one or more contact intensity sensors 265. FIG. 2A shows a contact intensity sensor coupled to intensity sensor controller 259 in I/O subsystem 206. Contact intensity sensor 265 optionally includes one or more piezoresistive strain gauges, capacitive force sensors, electric force sensors, piezoelectric force sensors, optical force sensors, capacitive touch-sensitive surfaces, or other intensity sensors (e.g., sensors used to measure the force (or pressure) of a contact on a touch-sensitive surface). Contact intensity sensor 265

receives contact intensity information (e.g., pressure information or a proxy for pressure information) from the environment. In some embodiments, at least one contact intensity sensor is collocated with, or proximate to, a touch-sensitive surface (e.g., touch-sensitive display system 212). In some embodiments, at least one contact intensity sensor is located on the back of device 200, opposite touch screen display 212, which is located on the front of device 200.

[0060] Device 200 also includes one or more proximity sensors 266. FIG. 2A shows proximity sensor 266 coupled to peripherals interface 218. Alternately, proximity sensor 266 is coupled to input controller 260 in I/O subsystem 206. Proximity sensor 266 is performed as described in U.S. Patent Application Nos. 11/241,839, “Proximity Detector In Handheld Device”; 11/240,788, “Proximity Detector In Handheld Device”; 11/620,702, “Using Ambient Light Sensor To Augment Proximity Sensor Output”; 11/586,862, “Automated Response To And Sensing Of User Activity In Portable Devices”; and 11/638,251, “Methods And Systems For Automatic Configuration Of Peripherals,” which are hereby incorporated by reference in their entirety. In some embodiments, the proximity sensor turns off and disables touch screen 212 when the multifunction device is placed near the user’s ear (e.g., when the user is making a phone call).

[0061] Device 200 optionally also includes one or more tactile output generators 267. FIG. 2A shows a tactile output generator coupled to haptic feedback controller 261 in I/O subsystem 206. Tactile output generator 267 optionally includes one or more electroacoustic devices such as speakers or other audio components and/or electromechanical devices that convert energy into linear motion such as a motor, solenoid, electroactive polymer, piezoelectric actuator, electrostatic actuator, or other tactile output generating component (e.g., a component that converts electrical signals into tactile outputs on the device). Contact intensity sensor 265 receives tactile feedback generation instructions from haptic feedback module 233 and generates tactile outputs on device 200 that are capable of being sensed by a user of device 200. In some embodiments, at least one tactile output generator is collocated with, or proximate to, a touch-sensitive surface (e.g., touch-sensitive display system 212) and, optionally, generates a tactile output by moving the touch-sensitive surface vertically (e.g., in/out of a surface of device 200) or laterally (e.g., back and forth in the same plane as a surface of device 200). In some

embodiments, at least one tactile output generator sensor is located on the back of device 200, opposite touch screen display 212, which is located on the front of device 200.

**[0062]** Device 200 also includes one or more accelerometers 268. FIG. 2A shows accelerometer 268 coupled to peripherals interface 218. Alternately, accelerometer 268 is coupled to an input controller 260 in I/O subsystem 206. Accelerometer 268 performs, for example, as described in U.S. Patent Publication No. 20050190059, “Acceleration-based Theft Detection System for Portable Electronic Devices,” and U.S. Patent Publication No. 20060017692, “Methods And Apparatuses For Operating A Portable Device Based On An Accelerometer,” both of which are incorporated by reference herein in their entirety. In some embodiments, information is displayed on the touch screen display in a portrait view or a landscape view based on an analysis of data received from the one or more accelerometers. Device 200 optionally includes, in addition to accelerometer(s) 268, a magnetometer (not shown) and a GPS (or GLONASS or other global navigation system) receiver (not shown) for obtaining information concerning the location and orientation (e.g., portrait or landscape) of device 200.

**[0063]** In some embodiments, the software components stored in memory 202 include operating system 226, communication module (or set of instructions) 228, contact/motion module (or set of instructions) 230, graphics module (or set of instructions) 232, text input module (or set of instructions) 234, Global Positioning System (GPS) module (or set of instructions) 235, Digital Assistant Client Module 229, and applications (or sets of instructions) 236. Further, memory 202 stores data and models, such as user data and models 231. Furthermore, in some embodiments, memory 202 (FIG. 2A) or 470 (FIG. 4) stores device/global internal state 257, as shown in FIGS. 2A and 4. Device/global internal state 257 includes one or more of: active application state, indicating which applications, if any, are currently active; display state, indicating what applications, views or other information occupy various regions of touch screen display 212; sensor state, including information obtained from the device’s various sensors and input control devices 216; and location information concerning the device’s location and/or attitude.

**[0064]** Operating system 226 (e.g., Darwin, RTXC, LINUX, UNIX, OS X, iOS, WINDOWS, or an embedded operating system such as VxWorks) includes various software

components and/or drivers for controlling and managing general system tasks (e.g., memory management, storage device control, power management, etc.) and facilitates communication between various hardware and software components.

**[0065]** Communication module 228 facilitates communication with other devices over one or more external ports 224 and also includes various software components for handling data received by RF circuitry 208 and/or external port 224. External port 224 (e.g., Universal Serial Bus (USB), FIREWIRE, etc.) is adapted for coupling directly to other devices or indirectly over a network (e.g., the Internet, wireless LAN, etc.). In some embodiments, the external port is a multi-pin (e.g., 30-pin) connector that is the same as, or similar to and/or compatible with, the 30-pin connector used on iPod® (trademark of Apple Inc.) devices.

**[0066]** Contact/motion module 230 optionally detects contact with touch screen 212 (in conjunction with display controller 256) and other touch-sensitive devices (e.g., a touchpad or physical click wheel). Contact/motion module 230 includes various software components for performing various operations related to detection of contact, such as determining if contact has occurred (e.g., detecting a finger-down event), determining an intensity of the contact (e.g., the force or pressure of the contact or a substitute for the force or pressure of the contact), determining if there is movement of the contact and tracking the movement across the touch-sensitive surface (e.g., detecting one or more finger-dragging events), and determining if the contact has ceased (e.g., detecting a finger-up event or a break in contact). Contact/motion module 230 receives contact data from the touch-sensitive surface. Determining movement of the point of contact, which is represented by a series of contact data, optionally includes determining speed (magnitude), velocity (magnitude and direction), and/or an acceleration (a change in magnitude and/or direction) of the point of contact. These operations are, optionally, applied to single contacts (e.g., one finger contacts) or to multiple simultaneous contacts (e.g., “multitouch”/multiple finger contacts). In some embodiments, contact/motion module 230 and display controller 256 detect contact on a touchpad.

**[0067]** In some embodiments, contact/motion module 230 uses a set of one or more intensity thresholds to determine whether an operation has been performed by a user (e.g., to determine whether a user has “clicked” on an icon). In some embodiments, at least a subset of the intensity

thresholds are determined in accordance with software parameters (e.g., the intensity thresholds are not determined by the activation thresholds of particular physical actuators and can be adjusted without changing the physical hardware of device 200). For example, a mouse “click” threshold of a trackpad or touch screen display can be set to any of a large range of predefined threshold values without changing the trackpad or touch screen display hardware. Additionally, in some implementations, a user of the device is provided with software settings for adjusting one or more of the set of intensity thresholds (e.g., by adjusting individual intensity thresholds and/or by adjusting a plurality of intensity thresholds at once with a system-level click “intensity” parameter).

**[0068]** Contact/motion module 230 optionally detects a gesture input by a user. Different gestures on the touch-sensitive surface have different contact patterns (e.g., different motions, timings, and/or intensities of detected contacts). Thus, a gesture is, optionally, detected by detecting a particular contact pattern. For example, detecting a finger tap gesture includes detecting a finger-down event followed by detecting a finger-up (liftoff) event at the same position (or substantially the same position) as the finger-down event (e.g., at the position of an icon). As another example, detecting a finger swipe gesture on the touch-sensitive surface includes detecting a finger-down event followed by detecting one or more finger-dragging events, and subsequently followed by detecting a finger-up (liftoff) event.

**[0069]** Graphics module 232 includes various known software components for rendering and displaying graphics on touch screen 212 or other display, including components for changing the visual impact (e.g., brightness, transparency, saturation, contrast, or other visual property) of graphics that are displayed. As used herein, the term “graphics” includes any object that can be displayed to a user, including, without limitation, text, web pages, icons (such as user-interface objects including soft keys), digital images, videos, animations, and the like.

**[0070]** In some embodiments, graphics module 232 stores data representing graphics to be used. Each graphic is, optionally, assigned a corresponding code. Graphics module 232 receives, from applications etc., one or more codes specifying graphics to be displayed along with, if necessary, coordinate data and other graphic property data, and then generates screen image data to output to display controller 256.

**[0071]** Haptic feedback module 233 includes various software components for generating instructions used by tactile output generator(s) 267 to produce tactile outputs at one or more locations on device 200 in response to user interactions with device 200.

**[0072]** Text input module 234, which is, in some examples, a component of graphics module 232, provides soft keyboards for entering text in various applications (e.g., contacts 237, email 240, IM 241, browser 247, and any other application that needs text input).

**[0073]** GPS module 235 determines the location of the device and provides this information for use in various applications (e.g., to telephone 238 for use in location-based dialing; to camera 243 as picture/video metadata; and to applications that provide location-based services such as weather widgets, local yellow page widgets, and map/navigation widgets).

**[0074]** Digital assistant client module 229 includes various client-side digital assistant instructions to provide the client-side functionalities of the digital assistant. For example, digital assistant client module 229 is capable of accepting voice input (e.g., speech input), text input, touch input, and/or gestural input through various user interfaces (e.g., microphone 213, accelerometer(s) 268, touch-sensitive display system 212, optical sensor(s) 229, other input control devices 216, etc.) of portable multifunction device 200. Digital assistant client module 229 is also capable of providing output in audio (e.g., speech output), visual, and/or tactile forms through various output interfaces (e.g., speaker 211, touch-sensitive display system 212, tactile output generator(s) 267, etc.) of portable multifunction device 200. For example, output is provided as voice, sound, alerts, text messages, menus, graphics, videos, animations, vibrations, and/or combinations of two or more of the above. During operation, digital assistant client module 229 communicates with DA server 106 using RF circuitry 208.

**[0075]** User data and models 231 include various data associated with the user (e.g., user-specific vocabulary data, user preference data, user-specified name pronunciations, data from the user's electronic address book, to-do lists, shopping lists, etc.) to provide the client-side functionalities of the digital assistant. Further, user data and models 231 include various models (e.g., speech recognition models, statistical language models, natural language processing models, ontology, task flow models, service models, etc.) for processing user input and determining user intent.



**[0076]** In some examples, digital assistant client module 229 utilizes the various sensors, subsystems, and peripheral devices of portable multifunction device 200 to gather additional information from the surrounding environment of the portable multifunction device 200 to establish a context associated with a user, the current user interaction, and/or the current user input. In some examples, digital assistant client module 229 provides the contextual information or a subset thereof with the user input to DA server 106 to help infer the user's intent. In some examples, the digital assistant also uses the contextual information to determine how to prepare and deliver outputs to the user. Contextual information is referred to as context data.

**[0077]** In some examples, the contextual information that accompanies the user input includes sensor information, e.g., lighting, ambient noise, ambient temperature, images or videos of the surrounding environment, etc. In some examples, the contextual information can also include the physical state of the device, e.g., device orientation, device location, device temperature, power level, speed, acceleration, motion patterns, cellular signals strength, etc. In some examples, information related to the software state of DA server 106, e.g., running processes, installed programs, past and present network activities, background services, error logs, resources usage, etc., and of portable multifunction device 200 is provided to DA server 106 as contextual information associated with a user input.

**[0078]** In some examples, the digital assistant client module 229 selectively provides information (e.g., user data 231) stored on the portable multifunction device 200 in response to requests from DA server 106. In some examples, digital assistant client module 229 also elicits additional input from the user via a natural language dialogue or other user interfaces upon request by DA server 106. Digital assistant client module 229 passes the additional input to DA server 106 to help DA server 106 in intent deduction and/or fulfillment of the user's intent expressed in the user request.

**[0079]** A more detailed description of a digital assistant is described below with reference to FIGs. 7A-C. It should be recognized that digital assistant client module 229 can include any number of the sub-modules of digital assistant module 726 described below.

**[0080]** Applications 236 include the following modules (or sets of instructions), or a subset or superset thereof:

- Contacts module 237 (sometimes called an address book or contact list);
- Telephone module 238;
- Video conference module 239;
- E-mail client module 240;
- Instant messaging (IM) module 241;
- Workout support module 242;
- Camera module 243 for still and/or video images;
- Image management module 244;
- Video player module;
- Music player module;
- Browser module 247;
- Calendar module 248;
- Widget modules 249, which includes, in some examples, one or more of: weather widget 249-1, stocks widget 249-2, calculator widget 249-3, alarm clock widget 249-4, dictionary widget 249-5, and other widgets obtained by the user, as well as user-created widgets 249-6;
- Widget creator module 250 for making user-created widgets 249-6;
- Search module 251;
- Video and music player module 252, which merges video player module and music player module;
- Notes module 253;

- Map module 254; and/or
- Online video module 255.

**[0081]** Examples of other applications 236 that are stored in memory 202 include other word processing applications, other image editing applications, drawing applications, presentation applications, JAVA-enabled applications, encryption, digital rights management, voice recognition, and voice replication.

**[0082]** In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, contacts module 237 are used to manage an address book or contact list (e.g., stored in application internal state 292 of contacts module 237 in memory 202 or memory 470), including: adding name(s) to the address book; deleting name(s) from the address book; associating telephone number(s), e-mail address(es), physical address(es) or other information with a name; associating an image with a name; categorizing and sorting names; providing telephone numbers or e-mail addresses to initiate and/or facilitate communications by telephone 238, video conference module 239, e-mail 240, or IM 241; and so forth.

**[0083]** In conjunction with RF circuitry 208, audio circuitry 210, speaker 211, microphone 213, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, telephone module 238 are used to enter a sequence of characters corresponding to a telephone number, access one or more telephone numbers in contacts module 237, modify a telephone number that has been entered, dial a respective telephone number, conduct a conversation, and disconnect or hang up when the conversation is completed. As noted above, the wireless communication uses any of a plurality of communications standards, protocols, and technologies.

**[0084]** In conjunction with RF circuitry 208, audio circuitry 210, speaker 211, microphone 213, touch screen 212, display controller 256, optical sensor 264, optical sensor controller 258, contact/motion module 230, graphics module 232, text input module 234, contacts module 237, and telephone module 238, video conference module 239 includes executable instructions to

initiate, conduct, and terminate a video conference between a user and one or more other participants in accordance with user instructions.

**[0085]** In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, e-mail client module 240 includes executable instructions to create, send, receive, and manage e-mail in response to user instructions. In conjunction with image management module 244, e-mail client module 240 makes it very easy to create and send e-mails with still or video images taken with camera module 243.

**[0086]** In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, the instant messaging module 241 includes executable instructions to enter a sequence of characters corresponding to an instant message, to modify previously entered characters, to transmit a respective instant message (for example, using a Short Message Service (SMS) or Multimedia Message Service (MMS) protocol for telephony-based instant messages or using XMPP, SIMPLE, or IMPS for Internet-based instant messages), to receive instant messages, and to view received instant messages. In some embodiments, transmitted and/or received instant messages include graphics, photos, audio files, video files and/or other attachments as are supported in an MMS and/or an Enhanced Messaging Service (EMS). As used herein, “instant messaging” refers to both telephony-based messages (e.g., messages sent using SMS or MMS) and Internet-based messages (e.g., messages sent using XMPP, SIMPLE, or IMPS).

**[0087]** In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, GPS module 235, map module 254, and music player module, workout support module 242 includes executable instructions to create workouts (e.g., with time, distance, and/or calorie burning goals); communicate with workout sensors (sports devices); receive workout sensor data; calibrate sensors used to monitor a workout; select and play music for a workout; and display, store, and transmit workout data.

**[0088]** In conjunction with touch screen 212, display controller 256, optical sensor(s) 264, optical sensor controller 258, contact/motion module 230, graphics module 232, and image

management module 244, camera module 243 includes executable instructions to capture still images or video (including a video stream) and store them into memory 202, modify characteristics of a still image or video, or delete a still image or video from memory 202.

**[0089]** In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, and camera module 243, image management module 244 includes executable instructions to arrange, modify (e.g., edit), or otherwise manipulate, label, delete, present (e.g., in a digital slide show or album), and store still and/or video images.

**[0090]** In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, browser module 247 includes executable instructions to browse the Internet in accordance with user instructions, including searching, linking to, receiving, and displaying web pages or portions thereof, as well as attachments and other files linked to web pages.

**[0091]** In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, e-mail client module 240, and browser module 247, calendar module 248 includes executable instructions to create, display, modify, and store calendars and data associated with calendars (e.g., calendar entries, to-do lists, etc.) in accordance with user instructions.

**[0092]** In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, and browser module 247, widget modules 249 are mini-applications that can be downloaded and used by a user (e.g., weather widget 249-1, stocks widget 249-2, calculator widget 249-3, alarm clock widget 249-4, and dictionary widget 249-5) or created by the user (e.g., user-created widget 249-6). In some embodiments, a widget includes an HTML (Hypertext Markup Language) file, a CSS (Cascading Style Sheets) file, and a JavaScript file. In some embodiments, a widget includes an XML (Extensible Markup Language) file and a JavaScript file (e.g., Yahoo! Widgets).

**[0093]** In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, and browser module

247, the widget creator module 250 are used by a user to create widgets (e.g., turning a user-specified portion of a web page into a widget).

**[0094]** In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, search module 251 includes executable instructions to search for text, music, sound, image, video, and/or other files in memory 202 that match one or more search criteria (e.g., one or more user-specified search terms) in accordance with user instructions.

**[0095]** In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, audio circuitry 210, speaker 211, RF circuitry 208, and browser module 247, video and music player module 252 includes executable instructions that allow the user to download and play back recorded music and other sound files stored in one or more file formats, such as MP3 or AAC files, and executable instructions to display, present, or otherwise play back videos (e.g., on touch screen 212 or on an external, connected display via external port 224). In some embodiments, device 200 optionally includes the functionality of an MP3 player, such as an iPod (trademark of Apple Inc.).

**[0096]** In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, notes module 253 includes executable instructions to create and manage notes, to-do lists, and the like in accordance with user instructions.

**[0097]** In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, GPS module 235, and browser module 247, map module 254 are used to receive, display, modify, and store maps and data associated with maps (e.g., driving directions, data on stores and other points of interest at or near a particular location, and other location-based data) in accordance with user instructions.

**[0098]** In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, audio circuitry 210, speaker 211, RF circuitry 208, text input module 234, e-mail client module 240, and browser module 247, online video module 255 includes instructions that allow the user to access, browse, receive (e.g., by streaming and/or download),

play back (e.g., on the touch screen or on an external, connected display via external port 224), send an e-mail with a link to a particular online video, and otherwise manage online videos in one or more file formats, such as H.264. In some embodiments, instant messaging module 241, rather than e-mail client module 240, is used to send a link to a particular online video.

Additional description of the online video application can be found in U.S. Provisional Patent Application No. 60/936,562, “Portable Multifunction Device, Method, and Graphical User Interface for Playing Online Videos,” filed June 20, 2007, and U.S. Patent Application No. 11/968,067, “Portable Multifunction Device, Method, and Graphical User Interface for Playing Online Videos,” filed December 31, 2007, the contents of which are hereby incorporated by reference in their entirety.

**[0099]** Each of the above-identified modules and applications corresponds to a set of executable instructions for performing one or more functions described above and the methods described in this application (e.g., the computer-implemented methods and other information processing methods described herein). These modules (e.g., sets of instructions) need not be implemented as separate software programs, procedures, or modules, and thus various subsets of these modules can be combined or otherwise rearranged in various embodiments. For example, video player module can be combined with music player module into a single module (e.g., video and music player module 252, FIG. 2A). In some embodiments, memory 202 stores a subset of the modules and data structures identified above. Furthermore, memory 202 stores additional modules and data structures not described above.

**[00100]** In some embodiments, device 200 is a device where operation of a predefined set of functions on the device is performed exclusively through a touch screen and/or a touchpad. By using a touch screen and/or a touchpad as the primary input control device for operation of device 200, the number of physical input control devices (such as push buttons, dials, and the like) on device 200 is reduced.

**[0100]** The predefined set of functions that are performed exclusively through a touch screen and/or a touchpad optionally include navigation between user interfaces. In some embodiments, the touchpad, when touched by the user, navigates device 200 to a main, home, or root menu from any user interface that is displayed on device 200. In such embodiments, a “menu button”

is implemented using a touchpad. In some other embodiments, the menu button is a physical push button or other physical input control device instead of a touchpad.

**[0101]** FIG. 2B is a block diagram illustrating exemplary components for event handling in accordance with some embodiments. In some embodiments, memory 202 (FIG. 2A) or 470 (FIG. 4) includes event sorter 270 (e.g., in operating system 226) and a respective application 236-1 (e.g., any of the aforementioned applications 237-251, 255, 480-490).

**[0102]** Event sorter 270 receives event information and determines the application 236-1 and application view 291 of application 236-1 to which to deliver the event information. Event sorter 270 includes event monitor 271 and event dispatcher module 274. In some embodiments, application 236-1 includes application internal state 292, which indicates the current application view(s) displayed on touch-sensitive display 212 when the application is active or executing. In some embodiments, device/global internal state 257 is used by event sorter 270 to determine which application(s) is (are) currently active, and application internal state 292 is used by event sorter 270 to determine application views 291 to which to deliver event information.

**[0103]** In some embodiments, application internal state 292 includes additional information, such as one or more of: resume information to be used when application 236-1 resumes execution, user interface state information that indicates information being displayed or that is ready for display by application 236-1, a state queue for enabling the user to go back to a prior state or view of application 236-1, and a redo/undo queue of previous actions taken by the user.

**[0104]** Event monitor 271 receives event information from peripherals interface 218. Event information includes information about a sub-event (e.g., a user touch on touch-sensitive display 212, as part of a multi-touch gesture). Peripherals interface 218 transmits information it receives from I/O subsystem 206 or a sensor, such as proximity sensor 266, accelerometer(s) 268, and/or microphone 213 (through audio circuitry 210). Information that peripherals interface 218 receives from I/O subsystem 206 includes information from touch-sensitive display 212 or a touch-sensitive surface.

**[0105]** In some embodiments, event monitor 271 sends requests to the peripherals interface 218 at predetermined intervals. In response, peripherals interface 218 transmits event



information. In other embodiments, peripherals interface 218 transmits event information only when there is a significant event (e.g., receiving an input above a predetermined noise threshold and/or for more than a predetermined duration).

**[0106]** In some embodiments, event sorter 270 also includes a hit view determination module 272 and/or an active event recognizer determination module 273.

**[0107]** Hit view determination module 272 provides software procedures for determining where a sub-event has taken place within one or more views when touch-sensitive display 212 displays more than one view. Views are made up of controls and other elements that a user can see on the display.

**[0108]** Another aspect of the user interface associated with an application is a set of views, sometimes herein called application views or user interface windows, in which information is displayed and touch-based gestures occur. The application views (of a respective application) in which a touch is detected correspond to programmatic levels within a programmatic or view hierarchy of the application. For example, the lowest level view in which a touch is detected is called the hit view, and the set of events that are recognized as proper inputs is determined based, at least in part, on the hit view of the initial touch that begins a touch-based gesture.

**[0109]** Hit view determination module 272 receives information related to sub events of a touch-based gesture. When an application has multiple views organized in a hierarchy, hit view determination module 272 identifies a hit view as the lowest view in the hierarchy which should handle the sub-event. In most circumstances, the hit view is the lowest level view in which an initiating sub-event occurs (e.g., the first sub-event in the sequence of sub-events that form an event or potential event). Once the hit view is identified by the hit view determination module 272, the hit view typically receives all sub-events related to the same touch or input source for which it was identified as the hit view.

**[0110]** Active event recognizer determination module 273 determines which view or views within a view hierarchy should receive a particular sequence of sub-events. In some embodiments, active event recognizer determination module 273 determines that only the hit view should receive a particular sequence of sub-events. In other embodiments, active event

recognizer determination module 273 determines that all views that include the physical location of a sub-event are actively involved views, and therefore determines that all actively involved views should receive a particular sequence of sub-events. In other embodiments, even if touch sub-events were entirely confined to the area associated with one particular view, views higher in the hierarchy would still remain as actively involved views.

**[0111]** Event dispatcher module 274 dispatches the event information to an event recognizer (e.g., event recognizer 280). In embodiments including active event recognizer determination module 273, event dispatcher module 274 delivers the event information to an event recognizer determined by active event recognizer determination module 273. In some embodiments, event dispatcher module 274 stores in an event queue the event information, which is retrieved by a respective event receiver 282.

**[0112]** In some embodiments, operating system 226 includes event sorter 270. Alternatively, application 236-1 includes event sorter 270. In yet other embodiments, event sorter 270 is a stand-alone module, or a part of another module stored in memory 202, such as contact/motion module 230.

**[0113]** In some embodiments, application 236-1 includes a plurality of event handlers 290 and one or more application views 291, each of which includes instructions for handling touch events that occur within a respective view of the application's user interface. Each application view 291 of the application 236-1 includes one or more event recognizers 280. Typically, a respective application view 291 includes a plurality of event recognizers 280. In other embodiments, one or more of event recognizers 280 are part of a separate module, such as a user interface kit (not shown) or a higher level object from which application 236-1 inherits methods and other properties. In some embodiments, a respective event handler 290 includes one or more of: data updater 276, object updater 277, GUI updater 278, and/or event data 279 received from event sorter 270. Event handler 290 utilizes or calls data updater 276, object updater 277, or GUI updater 278 to update the application internal state 292. Alternatively, one or more of the application views 291 include one or more respective event handlers 290. Also, in some embodiments, one or more of data updater 276, object updater 277, and GUI updater 278 are included in a respective application view 291.

**[0114]** A respective event recognizer 280 receives event information (e.g., event data 279) from event sorter 270 and identifies an event from the event information. Event recognizer 280 includes event receiver 282 and event comparator 284. In some embodiments, event recognizer 280 also includes at least a subset of: metadata 283, and event delivery instructions 288 (which include sub-event delivery instructions).

**[0115]** Event receiver 282 receives event information from event sorter 270. The event information includes information about a sub-event, for example, a touch or a touch movement. Depending on the sub-event, the event information also includes additional information, such as location of the sub-event. When the sub-event concerns motion of a touch, the event information also includes speed and direction of the sub-event. In some embodiments, events include rotation of the device from one orientation to another (e.g., from a portrait orientation to a landscape orientation, or vice versa), and the event information includes corresponding information about the current orientation (also called device attitude) of the device.

**[0116]** Event comparator 284 compares the event information to predefined event or sub-event definitions and, based on the comparison, determines an event or sub event, or determines or updates the state of an event or sub-event. In some embodiments, event comparator 284 includes event definitions 286. Event definitions 286 contain definitions of events (e.g., predefined sequences of sub-events), for example, event 1 (287-1), event 2 (287-2), and others. In some embodiments, sub-events in an event (287) include, for example, touch begin, touch end, touch movement, touch cancellation, and multiple touching. In one example, the definition for event 1 (287-1) is a double tap on a displayed object. The double tap, for example, comprises a first touch (touch begin) on the displayed object for a predetermined phase, a first liftoff (touch end) for a predetermined phase, a second touch (touch begin) on the displayed object for a predetermined phase, and a second liftoff (touch end) for a predetermined phase. In another example, the definition for event 2 (287-2) is a dragging on a displayed object. The dragging, for example, comprises a touch (or contact) on the displayed object for a predetermined phase, a movement of the touch across touch-sensitive display 212, and liftoff of the touch (touch end). In some embodiments, the event also includes information for one or more associated event handlers 290.

**[0117]** In some embodiments, event definition 287 includes a definition of an event for a respective user-interface object. In some embodiments, event comparator 284 performs a hit test to determine which user-interface object is associated with a sub-event. For example, in an application view in which three user-interface objects are displayed on touch-sensitive display 212, when a touch is detected on touch-sensitive display 212, event comparator 284 performs a hit test to determine which of the three user-interface objects is associated with the touch (sub-event). If each displayed object is associated with a respective event handler 290, the event comparator uses the result of the hit test to determine which event handler 290 should be activated. For example, event comparator 284 selects an event handler associated with the sub-event and the object triggering the hit test.

**[0118]** In some embodiments, the definition for a respective event (287) also includes delayed actions that delay delivery of the event information until after it has been determined whether the sequence of sub-events does or does not correspond to the event recognizer's event type.

**[0119]** When a respective event recognizer 280 determines that the series of sub-events do not match any of the events in event definitions 286, the respective event recognizer 280 enters an event impossible, event failed, or event ended state, after which it disregards subsequent sub-events of the touch-based gesture. In this situation, other event recognizers, if any, that remain active for the hit view continue to track and process sub-events of an ongoing touch-based gesture.

**[0120]** In some embodiments, a respective event recognizer 280 includes metadata 283 with configurable properties, flags, and/or lists that indicate how the event delivery system should perform sub-event delivery to actively involved event recognizers. In some embodiments, metadata 283 includes configurable properties, flags, and/or lists that indicate how event recognizers interact, or are enabled to interact, with one another. In some embodiments, metadata 283 includes configurable properties, flags, and/or lists that indicate whether sub-events are delivered to varying levels in the view or programmatic hierarchy.

**[0121]** In some embodiments, a respective event recognizer 280 activates event handler 290 associated with an event when one or more particular sub-events of an event are recognized. In

some embodiments, a respective event recognizer 280 delivers event information associated with the event to event handler 290. Activating an event handler 290 is distinct from sending (and deferred sending) sub-events to a respective hit view. In some embodiments, event recognizer 280 throws a flag associated with the recognized event, and event handler 290 associated with the flag catches the flag and performs a predefined process.

**[0122]** In some embodiments, event delivery instructions 288 include sub-event delivery instructions that deliver event information about a sub-event without activating an event handler. Instead, the sub-event delivery instructions deliver event information to event handlers associated with the series of sub-events or to actively involved views. Event handlers associated with the series of sub-events or with actively involved views receive the event information and perform a predetermined process.

**[0123]** In some embodiments, data updater 276 creates and updates data used in application 236-1. For example, data updater 276 updates the telephone number used in contacts module 237, or stores a video file used in video player module. In some embodiments, object updater 277 creates and updates objects used in application 236-1. For example, object updater 277 creates a new user-interface object or updates the position of a user-interface object. GUI updater 278 updates the GUI. For example, GUI updater 278 prepares display information and sends it to graphics module 232 for display on a touch-sensitive display.

**[0124]** In some embodiments, event handler(s) 290 includes or has access to data updater 276, object updater 277, and GUI updater 278. In some embodiments, data updater 276, object updater 277, and GUI updater 278 are included in a single module of a respective application 236-1 or application view 291. In other embodiments, they are included in two or more software modules.

**[0125]** It shall be understood that the foregoing discussion regarding event handling of user touches on touch-sensitive displays also applies to other forms of user inputs to operate multifunction devices 200 with input devices, not all of which are initiated on touch screens. For example, mouse movement and mouse button presses, optionally coordinated with single or multiple keyboard presses or holds; contact movements such as taps, drags, scrolls, etc. on touchpads; pen stylus inputs; movement of the device; oral instructions; detected eye

movements; biometric inputs; and/or any combination thereof are optionally utilized as inputs corresponding to sub-events which define an event to be recognized.

**[0126]** FIG. 3 illustrates a portable multifunction device 200 having a touch screen 212 in accordance with some embodiments. The touch screen optionally displays one or more graphics within user interface (UI) 300. In this embodiment, as well as others described below, a user is enabled to select one or more of the graphics by making a gesture on the graphics, for example, with one or more fingers 302 (not drawn to scale in the figure) or one or more styluses 303 (not drawn to scale in the figure). In some embodiments, selection of one or more graphics occurs when the user breaks contact with the one or more graphics. In some embodiments, the gesture optionally includes one or more taps, one or more swipes (from left to right, right to left, upward and/or downward), and/or a rolling of a finger (from right to left, left to right, upward and/or downward) that has made contact with device 200. In some implementations or circumstances, inadvertent contact with a graphic does not select the graphic. For example, a swipe gesture that sweeps over an application icon optionally does not select the corresponding application when the gesture corresponding to selection is a tap.

**[0127]** Device 200 also includes one or more physical buttons, such as “home” or menu button 304. As described previously, menu button 304 is used to navigate to any application 236 in a set of applications that is executed on device 200. Alternatively, in some embodiments, the menu button is implemented as a soft key in a GUI displayed on touch screen 212.

**[0128]** In one embodiment, device 200 includes touch screen 212, menu button 304, push button 306 for powering the device on/off and locking the device, volume adjustment button(s) 308, subscriber identity module (SIM) card slot 310, headset jack 312, and docking/charging external port 224. Push button 306 is, optionally, used to turn the power on/off on the device by depressing the button and holding the button in the depressed state for a predefined time interval; to lock the device by depressing the button and releasing the button before the predefined time interval has elapsed; and/or to unlock the device or initiate an unlock process. In an alternative embodiment, device 200 also accepts verbal input for activation or deactivation of some functions through microphone 213. Device 200 also, optionally, includes one or more contact

intensity sensors 265 for detecting intensity of contacts on touch screen 212 and/or one or more tactile output generators 267 for generating tactile outputs for a user of device 200.

**[0129]** FIG. 4 is a block diagram of an exemplary multifunction device with a display and a touch-sensitive surface in accordance with some embodiments. Device 400 need not be portable. In some embodiments, device 400 is a laptop computer, a desktop computer, a tablet computer, a multimedia player device, a navigation device, an educational device (such as a child's learning toy), a gaming system, or a control device (e.g., a home or industrial controller). Device 400 typically includes one or more processing units (CPUs) 410, one or more network or other communications interfaces 460, memory 470, and one or more communication buses 420 for interconnecting these components. Communication buses 420 optionally include circuitry (sometimes called a chipset) that interconnects and controls communications between system components. Device 400 includes input/output (I/O) interface 430 comprising display 440, which is typically a touch screen display. I/O interface 430 also optionally includes a keyboard and/or mouse (or other pointing device) 450 and touchpad 455, tactile output generator 457 for generating tactile outputs on device 400 (e.g., similar to tactile output generator(s) 267 described above with reference to FIG. 2A), sensors 459 (e.g., optical, acceleration, proximity, touch-sensitive, and/or contact intensity sensors similar to contact intensity sensor(s) 265 described above with reference to FIG. 2A). Memory 470 includes high-speed random access memory, such as DRAM, SRAM, DDR RAM, or other random access solid state memory devices; and optionally includes non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. Memory 470 optionally includes one or more storage devices remotely located from CPU(s) 410. In some embodiments, memory 470 stores programs, modules, and data structures analogous to the programs, modules, and data structures stored in memory 202 of portable multifunction device 200 (FIG. 2A), or a subset thereof. Furthermore, memory 470 optionally stores additional programs, modules, and data structures not present in memory 202 of portable multifunction device 200. For example, memory 470 of device 400 optionally stores drawing module 480, presentation module 482, word processing module 484, website creation module 486, disk authoring module 488, and/or spreadsheet module 490, while memory 202 of portable multifunction device 200 (FIG. 2A) optionally does not store these modules.

**[0130]** Each of the above-identified elements in FIG. 4 is, in some examples, stored in one or more of the previously mentioned memory devices. Each of the above-identified modules corresponds to a set of instructions for performing a function described above. The above-identified modules or programs (e.g., sets of instructions) need not be implemented as separate software programs, procedures, or modules, and thus various subsets of these modules are combined or otherwise rearranged in various embodiments. In some embodiments, memory 470 stores a subset of the modules and data structures identified above. Furthermore, memory 470 stores additional modules and data structures not described above.

**[0131]** Attention is now directed towards embodiments of user interfaces that can be implemented on, for example, portable multifunction device 200.

**[0132]** FIG. 5A illustrates an exemplary user interface for a menu of applications on portable multifunction device 200 in accordance with some embodiments. Similar user interfaces are implemented on device 400. In some embodiments, user interface 500 includes the following elements, or a subset or superset thereof:

**[0133]** Signal strength indicator(s) 502 for wireless communication(s), such as cellular and Wi-Fi signals;

- Time 504;
- Bluetooth indicator 505;
- Battery status indicator 506;
- Tray 508 with icons for frequently used applications, such as:
  - Icon 516 for telephone module 238, labeled “Phone,” which optionally includes an indicator 514 of the number of missed calls or voicemail messages;
  - Icon 518 for e-mail client module 240, labeled “Mail,” which optionally includes an indicator 510 of the number of unread e-mails;
  - Icon 520 for browser module 247, labeled “Browser;” and



- Icon 522 for video and music player module 252, also referred to as iPod (trademark of Apple Inc.) module 252, labeled “iPod;” and
- Icons for other applications, such as:
  - Icon 524 for IM module 241, labeled “Messages;”
  - Icon 526 for calendar module 248, labeled “Calendar;”
  - Icon 528 for image management module 244, labeled “Photos;”
  - Icon 530 for camera module 243, labeled “Camera;”
  - Icon 532 for online video module 255, labeled “Online Video;”
  - Icon 534 for stocks widget 249-2, labeled “Stocks;”
  - Icon 536 for map module 254, labeled “Maps;”
  - Icon 538 for weather widget 249-1, labeled “Weather;”
  - Icon 540 for alarm clock widget 249-4, labeled “Clock;”
  - Icon 542 for workout support module 242, labeled “Workout Support;”
  - Icon 544 for notes module 253, labeled “Notes;” and
  - Icon 546 for a settings application or module, labeled “Settings;” which provides access to settings for device 200 and its various applications 236.

**[0134]** It should be noted that the icon labels illustrated in FIG. 5A are merely exemplary. For example, icon 522 for video and music player module 252 is optionally labeled “Music” or “Music Player.” Other labels are, optionally, used for various application icons. In some embodiments, a label for a respective application icon includes a name of an application corresponding to the respective application icon. In some embodiments, a label for a particular application icon is distinct from a name of an application corresponding to the particular application icon.

**[0135]** FIG. 5B illustrates an exemplary user interface on a device (e.g., device 400, FIG. 4) with a touch-sensitive surface 551 (e.g., a tablet or touchpad 455, FIG. 4) that is separate from the display 550 (e.g., touch screen display 212). Device 400 also, optionally, includes one or more contact intensity sensors (e.g., one or more of sensors 457) for detecting intensity of contacts on touch-sensitive surface 551 and/or one or more tactile output generators 459 for generating tactile outputs for a user of device 400.

**[0136]** Although some of the examples which follow will be given with reference to inputs on touch screen display 212 (where the touch-sensitive surface and the display are combined), in some embodiments, the device detects inputs on a touch-sensitive surface that is separate from the display, as shown in FIG. 5B. In some embodiments, the touch-sensitive surface (e.g., 551 in FIG. 5B) has a primary axis (e.g., 552 in FIG. 5B) that corresponds to a primary axis (e.g., 553 in FIG. 5B) on the display (e.g., 550). In accordance with these embodiments, the device detects contacts (e.g., 560 and 562 in FIG. 5B) with the touch-sensitive surface 551 at locations that correspond to respective locations on the display (e.g., in FIG. 5B, 560 corresponds to 568 and 562 corresponds to 570). In this way, user inputs (e.g., contacts 560 and 562, and movements thereof) detected by the device on the touch-sensitive surface (e.g., 551 in FIG. 5B) are used by the device to manipulate the user interface on the display (e.g., 550 in FIG. 5B) of the multifunction device when the touch-sensitive surface is separate from the display. It should be understood that similar methods are, optionally, used for other user interfaces described herein.

**[0137]** Additionally, while the following examples are given primarily with reference to finger inputs (e.g., finger contacts, finger tap gestures, finger swipe gestures), it should be understood that, in some embodiments, one or more of the finger inputs are replaced with input from another input device (e.g., a mouse-based input or stylus input). For example, a swipe gesture is, optionally, replaced with a mouse click (e.g., instead of a contact) followed by movement of the cursor along the path of the swipe (e.g., instead of movement of the contact). As another example, a tap gesture is, optionally, replaced with a mouse click while the cursor is located over the location of the tap gesture (e.g., instead of detection of the contact followed by ceasing to detect the contact). Similarly, when multiple user inputs are simultaneously detected, it should be understood that multiple computer mice are, optionally, used simultaneously, or a mouse and finger contacts are, optionally, used simultaneously.

**[0138]** FIG. 6A illustrates exemplary personal electronic device 600. Device 600 includes body 602. In some embodiments, device 600 includes some or all of the features described with respect to devices 200 and 400 (e.g., FIGS. 2A-4B). In some embodiments, device 600 has touch-sensitive display screen 604, hereafter touch screen 604. Alternatively, or in addition to touch screen 604, device 600 has a display and a touch-sensitive surface. As with devices 200 and 400, in some embodiments, touch screen 604 (or the touch-sensitive surface) has one or more intensity sensors for detecting intensity of contacts (e.g., touches) being applied. The one or more intensity sensors of touch screen 604 (or the touch-sensitive surface) provide output data that represents the intensity of touches. The user interface of device 600 responds to touches based on their intensity, meaning that touches of different intensities can invoke different user interface operations on device 600.

**[0139]** Techniques for detecting and processing touch intensity are found, for example, in related applications: International Patent Application Serial No. PCT/US2013/040061, titled “Device, Method, and Graphical User Interface for Displaying User Interface Objects Corresponding to an Application,” filed May 8, 2013, and International Patent Application Serial No. PCT/US2013/069483, titled “Device, Method, and Graphical User Interface for Transitioning Between Touch Input to Display Output Relationships,” filed November 11, 2013, each of which is hereby incorporated by reference in their entirety.

**[0140]** In some embodiments, device 600 has one or more input mechanisms 606 and 608. Input mechanisms 606 and 608, if included, are physical. Examples of physical input mechanisms include push buttons and rotatable mechanisms. In some embodiments, device 600 has one or more attachment mechanisms. Such attachment mechanisms, if included, can permit attachment of device 600 with, for example, hats, eyewear, earrings, necklaces, shirts, jackets, bracelets, watch straps, chains, trousers, belts, shoes, purses, backpacks, and so forth. These attachment mechanisms permit device 600 to be worn by a user.

**[0141]** FIG. 6B depicts exemplary personal electronic device 600. In some embodiments, device 600 includes some or all of the components described with respect to FIGS. 2A, 2B, and 4. Device 600 has bus 612 that operatively couples I/O section 614 with one or more computer processors 616 and memory 618. I/O section 614 is connected to display 604, which can have

touch-sensitive component 622 and, optionally, touch-intensity sensitive component 624. In addition, I/O section 614 is connected with communication unit 630 for receiving application and operating system data, using Wi-Fi, Bluetooth, near field communication (NFC), cellular, and/or other wireless communication techniques. Device 600 includes input mechanisms 606 and/or 608. Input mechanism 606 is a rotatable input device or a depressible and rotatable input device, for example. Input mechanism 608 is a button, in some examples.

**[0142]** Input mechanism 608 is a microphone, in some examples. Personal electronic device 600 includes, for example, various sensors, such as GPS sensor 632, accelerometer 634, directional sensor 640 (e.g., compass), gyroscope 636, motion sensor 638, and/or a combination thereof, all of which are operatively connected to I/O section 614.

**[0143]** Memory 618 of personal electronic device 600 is a non-transitory computer-readable storage medium, for storing computer-executable instructions, which, when executed by one or more computer processors 616, for example, cause the computer processors to perform the techniques and processes described below. The computer-executable instructions, for example, are also stored and/or transported within any non-transitory computer-readable storage medium for use by or in connection with an instruction execution system, apparatus, or device, such as a computer-based system, processor-containing system, or other system that can fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions. Personal electronic device 600 is not limited to the components and configuration of FIG. 6B, but can include other or additional components in multiple configurations.

**[0144]** As used here, the term “affordance” refers to a user-interactive graphical user interface object that is, for example, displayed on the display screen of devices 200, 400, and/or 600 (FIGS. 2, 4, and 6). For example, an image (e.g., icon), a button, and text (e.g., hyperlink) each constitutes an affordance.

**[0145]** As used herein, the term “focus selector” refers to an input element that indicates a current part of a user interface with which a user is interacting. In some implementations that include a cursor or other location marker, the cursor acts as a “focus selector” so that when an input (e.g., a press input) is detected on a touch-sensitive surface (e.g., touchpad 455 in FIG. 4 or touch-sensitive surface 551 in FIG. 5B) while the cursor is over a particular user interface

element (e.g., a button, window, slider or other user interface element), the particular user interface element is adjusted in accordance with the detected input. In some implementations that include a touch screen display (e.g., touch-sensitive display system 212 in FIG. 2A or touch screen 212 in FIG. 5A) that enables direct interaction with user interface elements on the touch screen display, a detected contact on the touch screen acts as a “focus selector” so that when an input (e.g., a press input by the contact) is detected on the touch screen display at a location of a particular user interface element (e.g., a button, window, slider, or other user interface element), the particular user interface element is adjusted in accordance with the detected input. In some implementations, focus is moved from one region of a user interface to another region of the user interface without corresponding movement of a cursor or movement of a contact on a touch screen display (e.g., by using a tab key or arrow keys to move focus from one button to another button); in these implementations, the focus selector moves in accordance with movement of focus between different regions of the user interface. Without regard to the specific form taken by the focus selector, the focus selector is generally the user interface element (or contact on a touch screen display) that is controlled by the user so as to communicate the user’s intended interaction with the user interface (e.g., by indicating, to the device, the element of the user interface with which the user is intending to interact). For example, the location of a focus selector (e.g., a cursor, a contact, or a selection box) over a respective button while a press input is detected on the touch-sensitive surface (e.g., a touchpad or touch screen) will indicate that the user is intending to activate the respective button (as opposed to other user interface elements shown on a display of the device).

**[0146]** As used in the specification and claims, the term “characteristic intensity” of a contact refers to a characteristic of the contact based on one or more intensities of the contact. In some embodiments, the characteristic intensity is based on multiple intensity samples. The characteristic intensity is, optionally, based on a predefined number of intensity samples, or a set of intensity samples collected during a predetermined time period (e.g., 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10 seconds) relative to a predefined event (e.g., after detecting the contact, prior to detecting liftoff of the contact, before or after detecting a start of movement of the contact, prior to detecting an end of the contact, before or after detecting an increase in intensity of the contact, and/or before or after detecting a decrease in intensity of the contact). A characteristic intensity of a contact is, optionally based on one or more of: a maximum value of the intensities of the

contact, a mean value of the intensities of the contact, an average value of the intensities of the contact, a top 10 percentile value of the intensities of the contact, a value at the half maximum of the intensities of the contact, a value at the 90 percent maximum of the intensities of the contact, or the like. In some embodiments, the duration of the contact is used in determining the characteristic intensity (e.g., when the characteristic intensity is an average of the intensity of the contact over time). In some embodiments, the characteristic intensity is compared to a set of one or more intensity thresholds to determine whether an operation has been performed by a user. For example, the set of one or more intensity thresholds includes a first intensity threshold and a second intensity threshold. In this example, a contact with a characteristic intensity that does not exceed the first threshold results in a first operation, a contact with a characteristic intensity that exceeds the first intensity threshold and does not exceed the second intensity threshold results in a second operation, and a contact with a characteristic intensity that exceeds the second threshold results in a third operation. In some embodiments, a comparison between the characteristic intensity and one or more thresholds is used to determine whether or not to perform one or more operations (e.g., whether to perform a respective operation or forgo performing the respective operation) rather than being used to determine whether to perform a first operation or a second operation.

**[0147]** In some embodiments, a portion of a gesture is identified for purposes of determining a characteristic intensity. For example, a touch-sensitive surface receives a continuous swipe contact transitioning from a start location and reaching an end location, at which point the intensity of the contact increases. In this example, the characteristic intensity of the contact at the end location is based on only a portion of the continuous swipe contact, and not the entire swipe contact (e.g., only the portion of the swipe contact at the end location). In some embodiments, a smoothing algorithm is applied to the intensities of the swipe contact prior to determining the characteristic intensity of the contact. For example, the smoothing algorithm optionally includes one or more of: an unweighted sliding-average smoothing algorithm, a triangular smoothing algorithm, a median filter smoothing algorithm, and/or an exponential smoothing algorithm. In some circumstances, these smoothing algorithms eliminate narrow spikes or dips in the intensities of the swipe contact for purposes of determining a characteristic intensity.

**[0148]** The intensity of a contact on the touch-sensitive surface is characterized relative to one or more intensity thresholds, such as a contact-detection intensity threshold, a light press intensity threshold, a deep press intensity threshold, and/or one or more other intensity thresholds. In some embodiments, the light press intensity threshold corresponds to an intensity at which the device will perform operations typically associated with clicking a button of a physical mouse or a trackpad. In some embodiments, the deep press intensity threshold corresponds to an intensity at which the device will perform operations that are different from operations typically associated with clicking a button of a physical mouse or a trackpad. In some embodiments, when a contact is detected with a characteristic intensity below the light press intensity threshold (e.g., and above a nominal contact-detection intensity threshold below which the contact is no longer detected), the device will move a focus selector in accordance with movement of the contact on the touch-sensitive surface without performing an operation associated with the light press intensity threshold or the deep press intensity threshold. Generally, unless otherwise stated, these intensity thresholds are consistent between different sets of user interface figures.

**[0149]** An increase of characteristic intensity of the contact from an intensity below the light press intensity threshold to an intensity between the light press intensity threshold and the deep press intensity threshold is sometimes referred to as a “light press” input. An increase of characteristic intensity of the contact from an intensity below the deep press intensity threshold to an intensity above the deep press intensity threshold is sometimes referred to as a “deep press” input. An increase of characteristic intensity of the contact from an intensity below the contact-detection intensity threshold to an intensity between the contact-detection intensity threshold and the light press intensity threshold is sometimes referred to as detecting the contact on the touch-surface. A decrease of characteristic intensity of the contact from an intensity above the contact-detection intensity threshold to an intensity below the contact-detection intensity threshold is sometimes referred to as detecting liftoff of the contact from the touch-surface. In some embodiments, the contact-detection intensity threshold is zero. In some embodiments, the contact-detection intensity threshold is greater than zero.

**[0150]** In some embodiments described herein, one or more operations are performed in response to detecting a gesture that includes a respective press input or in response to detecting

the respective press input performed with a respective contact (or a plurality of contacts), where the respective press input is detected based at least in part on detecting an increase in intensity of the contact (or plurality of contacts) above a press-input intensity threshold. In some embodiments, the respective operation is performed in response to detecting the increase in intensity of the respective contact above the press-input intensity threshold (e.g., a “down stroke” of the respective press input). In some embodiments, the press input includes an increase in intensity of the respective contact above the press-input intensity threshold and a subsequent decrease in intensity of the contact below the press-input intensity threshold, and the respective operation is performed in response to detecting the subsequent decrease in intensity of the respective contact below the press-input threshold (e.g., an “up stroke” of the respective press input).

**[0151]** In some embodiments, the device employs intensity hysteresis to avoid accidental inputs sometimes termed “jitter,” where the device defines or selects a hysteresis intensity threshold with a predefined relationship to the press-input intensity threshold (e.g., the hysteresis intensity threshold is X intensity units lower than the press-input intensity threshold or the hysteresis intensity threshold is 75%, 90%, or some reasonable proportion of the press-input intensity threshold). Thus, in some embodiments, the press input includes an increase in intensity of the respective contact above the press-input intensity threshold and a subsequent decrease in intensity of the contact below the hysteresis intensity threshold that corresponds to the press-input intensity threshold, and the respective operation is performed in response to detecting the subsequent decrease in intensity of the respective contact below the hysteresis intensity threshold (e.g., an “up stroke” of the respective press input). Similarly, in some embodiments, the press input is detected only when the device detects an increase in intensity of the contact from an intensity at or below the hysteresis intensity threshold to an intensity at or above the press-input intensity threshold and, optionally, a subsequent decrease in intensity of the contact to an intensity at or below the hysteresis intensity, and the respective operation is performed in response to detecting the press input (e.g., the increase in intensity of the contact or the decrease in intensity of the contact, depending on the circumstances).

**[0152]** For ease of explanation, the descriptions of operations performed in response to a press input associated with a press-input intensity threshold or in response to a gesture including



the press input are, optionally, triggered in response to detecting either: an increase in intensity of a contact above the press-input intensity threshold, an increase in intensity of a contact from an intensity below the hysteresis intensity threshold to an intensity above the press-input intensity threshold, a decrease in intensity of the contact below the press-input intensity threshold, and/or a decrease in intensity of the contact below the hysteresis intensity threshold corresponding to the press-input intensity threshold. Additionally, in examples where an operation is described as being performed in response to detecting a decrease in intensity of a contact below the press-input intensity threshold, the operation is, optionally, performed in response to detecting a decrease in intensity of the contact below a hysteresis intensity threshold corresponding to, and lower than, the press-input intensity threshold.

### 3. Digital Assistant System

**[0153]** FIG. 7A illustrates a block diagram of digital assistant system 700 in accordance with various examples. In some examples, digital assistant system 700 is implemented on a standalone computer system. In some examples, digital assistant system 700 is distributed across multiple computers. In some examples, some of the modules and functions of the digital assistant are divided into a server portion and a client portion, where the client portion resides on one or more user devices (e.g., devices 104, 122, 200, 400, or 600) and communicates with the server portion (e.g., server system 108) through one or more networks, e.g., as shown in FIG. 1. In some examples, digital assistant system 700 is an implementation of server system 108 (and/or DA server 106) shown in FIG. 1. It should be noted that digital assistant system 700 is only one example of a digital assistant system, and that digital assistant system 700 can have more or fewer components than shown, can combine two or more components, or can have a different configuration or arrangement of the components. The various components shown in FIG. 7A are implemented in hardware, software instructions for execution by one or more processors, firmware, including one or more signal processing and/or application specific integrated circuits, or a combination thereof.

**[0154]** Digital assistant system 700 includes memory 702, one or more processors 704, input/output (I/O) interface 706, and network communications interface 708. These components can communicate with one another over one or more communication buses or signal lines 710.

**[0155]** In some examples, memory 702 includes a non-transitory computer-readable medium, such as high-speed random access memory and/or a non-volatile computer-readable storage medium (e.g., one or more magnetic disk storage devices, flash memory devices, or other non-volatile solid-state memory devices).

**[0156]** In some examples, I/O interface 706 couples input/output devices 716 of digital assistant system 700, such as displays, keyboards, touch screens, and microphones, to user interface module 722. I/O interface 706, in conjunction with user interface module 722, receives user inputs (e.g., voice input, keyboard inputs, touch inputs, etc.) and processes them accordingly. In some examples, e.g., when the digital assistant is implemented on a standalone user device, digital assistant system 700 includes any of the components and I/O communication interfaces described with respect to devices 200, 400, or 600 in FIGs. 2A, 4, 6A-B, respectively. In some examples, digital assistant system 700 represents the server portion of a digital assistant implementation, and can interact with the user through a client-side portion residing on a user device (e.g., devices 104, 200, 400, or 600).

**[0157]** In some examples, the network communications interface 708 includes wired communication port(s) 712 and/or wireless transmission and reception circuitry 714. The wired communication port(s) receives and send communication signals via one or more wired interfaces, e.g., Ethernet, Universal Serial Bus (USB), FIREWIRE, etc. The wireless circuitry 714 receives and sends RF signals and/or optical signals from/to communications networks and other communications devices. The wireless communications use any of a plurality of communications standards, protocols, and technologies, such as GSM, EDGE, CDMA, TDMA, Bluetooth, Wi-Fi, VoIP, Wi-MAX, or any other suitable communication protocol. Network communications interface 708 enables communication between digital assistant system 700 with networks, such as the Internet, an intranet, and/or a wireless network, such as a cellular telephone network, a wireless local area network (LAN), and/or a metropolitan area network (MAN), and other devices.

**[0158]** In some examples, memory 702, or the computer-readable storage media of memory 702, stores programs, modules, instructions, and data structures including all or a subset of: operating system 718, communications module 720, user interface module 722, one or more applications 724, and digital assistant module 726. In particular, memory 702, or the computer-readable storage media of memory 702, stores instructions for performing the processes described below. One or more processors 704 execute these programs, modules, and instructions, and reads/writes from/to the data structures.

**[0159]** Operating system 718 (e.g., Darwin, RTXC, LINUX, UNIX, iOS, OS X, WINDOWS, or an embedded operating system such as VxWorks) includes various software components and/or drivers for controlling and managing general system tasks (e.g., memory management, storage device control, power management, etc.) and facilitates communications between various hardware, firmware, and software components.

**[0160]** Communications module 720 facilitates communications between digital assistant system 700 with other devices over network communications interface 708. For example, communications module 720 communicates with RF circuitry 208 of electronic devices such as devices 200, 400, and 600 shown in FIG. 2A, 4, 6A-B, respectively. Communications module 720 also includes various components for handling data received by wireless circuitry 714 and/or wired communications port 712.

**[0161]** User interface module 722 receives commands and/or inputs from a user via I/O interface 706 (e.g., from a keyboard, touch screen, pointing device, controller, and/or microphone), and generate user interface objects on a display. User interface module 722 also prepares and delivers outputs (e.g., speech, sound, animation, text, icons, vibrations, haptic feedback, light, etc.) to the user via the I/O interface 706 (e.g., through displays, audio channels, speakers, touch-pads, etc.).

**[0162]** Applications 724 include programs and/or modules that are configured to be executed by one or more processors 704. For example, if the digital assistant system is implemented on a standalone user device, applications 724 include user applications, such as games, a calendar application, a navigation application, or an email application. If digital assistant system 700 is

implemented on a server, applications 724 include resource management applications, diagnostic applications, or scheduling applications, for example.

**[0163]** Memory 702 also stores digital assistant module 726 (or the server portion of a digital assistant). In some examples, digital assistant module 726 includes the following sub-modules, or a subset or superset thereof: input/output processing module 728, speech-to-text (STT) processing module 730, natural language processing module 732, dialogue flow processing module 734, task flow processing module 736, service processing module 738, and speech synthesis module 740. Each of these modules has access to one or more of the following systems or data and models of the digital assistant module 726, or a subset or superset thereof: ontology 760, vocabulary index 744, user data 748, task flow models 754, service models 756, and ASR systems.

**[0164]** In some examples, using the processing modules, data, and models implemented in digital assistant module 726, the digital assistant can perform at least some of the following: converting speech input into text; identifying a user's intent expressed in a natural language input received from the user; actively eliciting and obtaining information needed to fully infer the user's intent (e.g., by disambiguating words, games, intentions, etc.); determining the task flow for fulfilling the inferred intent; and executing the task flow to fulfill the inferred intent.

**[0165]** In some examples, as shown in FIG. 7B, I/O processing module 728 interacts with the user through I/O devices 716 in FIG. 7A or with a user device (e.g., devices 104, 200, 400, or 600) through network communications interface 708 in FIG. 7A to obtain user input (e.g., a speech input) and to provide responses (e.g., as speech outputs) to the user input. I/O processing module 728 optionally obtains contextual information associated with the user input from the user device, along with or shortly after the receipt of the user input. The contextual information includes user-specific data, vocabulary, and/or preferences relevant to the user input. In some examples, the contextual information also includes software and hardware states of the user device at the time the user request is received, and/or information related to the surrounding environment of the user at the time that the user request was received. In some examples, I/O processing module 728 also sends follow-up questions to, and receive answers from, the user regarding the user request. When a user request is received by I/O processing module 728 and

the user request includes speech input, I/O processing module 728 forwards the speech input to STT processing module 730 (or speech recognizer) for speech-to-text conversions.

**[0166]** STT processing module 730 includes one or more ASR systems. The one or more ASR systems can process the speech input that is received through I/O processing module 728 to produce a recognition result. Each ASR system includes a front-end speech pre-processor. The front-end speech pre-processor extracts representative features from the speech input. For example, the front-end speech pre-processor performs a Fourier transform on the speech input to extract spectral features that characterize the speech input as a sequence of representative multi-dimensional vectors. Further, each ASR system includes one or more speech recognition models (e.g., acoustic models and/or language models) and implements one or more speech recognition engines. Examples of speech recognition models include Hidden Markov Models, Gaussian-Mixture Models, Deep Neural Network Models, n-gram language models, and other statistical models. Examples of speech recognition engines include the dynamic time warping based engines and weighted finite-state transducers (WFST) based engines. The one or more speech recognition models and the one or more speech recognition engines are used to process the extracted representative features of the front-end speech pre-processor to produce intermediate recognitions results (e.g., phonemes, phonemic strings, and sub-words), and ultimately, text recognition results (e.g., words, word strings, or sequence of tokens). In some examples, the speech input is processed at least partially by a third-party service or on the user's device (e.g., device 104, 200, 400, or 600) to produce the recognition result. Once STT processing module 730 produces recognition results containing a text string (e.g., words, or sequence of words, or sequence of tokens), the recognition result is passed to natural language processing module 732 for intent deduction.

**[0167]** More details on the speech-to-text processing are described in U.S. Utility Application Serial No. 13/236,942 for "Consolidating Speech Recognition Results," filed on September 20, 2011, the entire disclosure of which is incorporated herein by reference.

**[0168]** In some examples, STT processing module 730 includes and/or accesses a vocabulary of recognizable words via phonetic alphabet conversion module 731. Each vocabulary word is associated with one or more candidate pronunciations of the word represented in a speech

recognition phonetic alphabet. In particular, the vocabulary of recognizable words includes a word that is associated with a plurality of candidate pronunciations. For example, the vocabulary includes the word “tomato” that is associated with the candidate pronunciations of /tə'meɪroʊ/ and /tə'matoʊ/. Further, vocabulary words are associated with custom candidate pronunciations that are based on previous speech inputs from the user. Such custom candidate pronunciations are stored in STT processing module 730 and are associated with a particular user via the user's profile on the device. In some examples, the candidate pronunciations for words are determined based on the spelling of the word and one or more linguistic and/or phonetic rules. In some examples, the candidate pronunciations are manually generated, e.g., based on known canonical pronunciations.

**[0169]** In some examples, the candidate pronunciations are ranked based on the commonness of the candidate pronunciation. For example, the candidate pronunciation /tə'meɪroʊ/ is ranked higher than /tə'matoʊ/, because the former is a more commonly used pronunciation (e.g., among all users, for users in a particular geographical region, or for any other appropriate subset of users). In some examples, candidate pronunciations are ranked based on whether the candidate pronunciation is a custom candidate pronunciation associated with the user. For example, custom candidate pronunciations are ranked higher than canonical candidate pronunciations. This can be useful for recognizing proper nouns having a unique pronunciation that deviates from canonical pronunciation. In some examples, candidate pronunciations are associated with one or more speech characteristics, such as geographic origin, nationality, or ethnicity. For example, the candidate pronunciation /tə'meɪroʊ/ is associated with the United States, whereas the candidate pronunciation /tə'matoʊ/ is associated with Great Britain. Further, the rank of the candidate pronunciation is based on one or more characteristics (e.g., geographic origin, nationality, ethnicity, etc.) of the user stored in the user's profile on the device. For example, it can be determined from the user's profile that the user is associated with the United States. Based on the user being associated with the United States, the candidate pronunciation /tə'meɪroʊ/ (associated with the United States) is ranked higher than the candidate pronunciation /tə'matoʊ/ (associated with Great Britain). In some examples, one of the ranked candidate pronunciations is selected as a predicted pronunciation (e.g., the most likely pronunciation).

**[0170]** When a speech input is received, STT processing module 730 is used to determine the phonemes corresponding to the speech input (e.g., using an acoustic model), and then attempt to determine words that match the phonemes (e.g., using a language model). For example, if STT processing module 730 first identifies the sequence of phonemes /tə'meɪrəʊ/ corresponding to a portion of the speech input, it can then determine, based on vocabulary index 744, that this sequence corresponds to the word “tomato.”

**[0171]** In some examples, STT processing module 730 uses approximate matching techniques to determine words in an utterance. Thus, for example, the STT processing module 730 determines that the sequence of phonemes /tə'meɪrəʊ/ corresponds to the word “tomato,” even if that particular sequence of phonemes is not one of the candidate sequence of phonemes for that word.

**[0172]** Natural language processing module 732 (“natural language processor”) of the digital assistant takes the sequence of words or tokens (“token sequence”) generated by STT processing module 730, and attempt to associate the token sequence with one or more “actionable intents” recognized by the digital assistant. An “actionable intent” represents a task that can be performed by the digital assistant, and can have an associated task flow implemented in task flow models 754. The associated task flow is a series of programmed actions and steps that the digital assistant takes in order to perform the task. The scope of a digital assistant’s capabilities is dependent on the number and variety of task flows that have been implemented and stored in task flow models 754, or in other words, on the number and variety of “actionable intents” that the digital assistant recognizes. The effectiveness of the digital assistant, however, also depends on the assistant’s ability to infer the correct “actionable intent(s)” from the user request expressed in natural language.

**[0173]** In some examples, in addition to the sequence of words or tokens obtained from STT processing module 730, natural language processing module 732 also receives contextual information associated with the user request, e.g., from I/O processing module 728. The natural language processing module 732 optionally uses the contextual information to clarify, supplement, and/or further define the information contained in the token sequence received from STT processing module 730. The contextual information includes, for example, user

preferences, hardware, and/or software states of the user device, sensor information collected before, during, or shortly after the user request, prior interactions (e.g., dialogue) between the digital assistant and the user, and the like. As described herein, contextual information is, in some examples, dynamic, and changes with time, location, content of the dialogue, and other factors.

**[0174]** In some examples, the natural language processing is based on, e.g., ontology 760. Ontology 760 is a hierarchical structure containing many nodes, each node representing either an “actionable intent” or a “property” relevant to one or more of the “actionable intents” or other “properties.” As noted above, an “actionable intent” represents a task that the digital assistant is capable of performing, i.e., it is “actionable” or can be acted on. A “property” represents a parameter associated with an actionable intent or a sub-aspect of another property. A linkage between an actionable intent node and a property node in ontology 760 defines how a parameter represented by the property node pertains to the task represented by the actionable intent node.

**[0175]** In some examples, ontology 760 is made up of actionable intent nodes and property nodes. Within ontology 760, each actionable intent node is linked to one or more property nodes either directly or through one or more intermediate property nodes. Similarly, each property node is linked to one or more actionable intent nodes either directly or through one or more intermediate property nodes. For example, as shown in FIG. 7C, ontology 760 includes a “restaurant reservation” node (i.e., an actionable intent node). Property nodes “restaurant,” “date/time” (for the reservation), and “party size” are each directly linked to the actionable intent node (i.e., the “restaurant reservation” node).

**[0176]** In addition, property nodes “cuisine,” “price range,” “phone number,” and “location” are sub-nodes of the property node “restaurant,” and are each linked to the “restaurant reservation” node (i.e., the actionable intent node) through the intermediate property node “restaurant.” For another example, as shown in FIG. 7C, ontology 760 also includes a “set reminder” node (i.e., another actionable intent node). Property nodes “date/time” (for setting the reminder) and “subject” (for the reminder) are each linked to the “set reminder” node. Since the property “date/time” is relevant to both the task of making a restaurant reservation and the task



of setting a reminder, the property node “date/time” is linked to both the “restaurant reservation” node and the “set reminder” node in ontology 760.

**[0177]** An actionable intent node, along with its linked concept nodes, is described as a “domain.” In the present discussion, each domain is associated with a respective actionable intent, and refers to the group of nodes (and the relationships there between) associated with the particular actionable intent. For example, ontology 760 shown in FIG. 7C includes an example of restaurant reservation domain 762 and an example of reminder domain 764 within ontology 760. The restaurant reservation domain includes the actionable intent node “restaurant reservation,” property nodes “restaurant,” “date/time,” and “party size,” and sub-property nodes “cuisine,” “price range,” “phone number,” and “location.” Reminder domain 764 includes the actionable intent node “set reminder,” and property nodes “subject” and “date/time.” In some examples, ontology 760 is made up of many domains. Each domain shares one or more property nodes with one or more other domains. For example, the “date/time” property node is associated with many different domains (e.g., a scheduling domain, a travel reservation domain, a movie ticket domain, etc.), in addition to restaurant reservation domain 762 and reminder domain 764.

**[0178]** While FIG. 7C illustrates two example domains within ontology 760, other domains include, for example, “find a movie,” “initiate a phone call,” “find directions,” “schedule a meeting,” “send a message,” and “provide an answer to a question,” “read a list,” “providing navigation instructions,” “provide instructions for a task” and so on. A “send a message” domain is associated with a “send a message” actionable intent node, and further includes property nodes such as “recipient(s),” “message type,” and “message body.” The property node “recipient” is further defined, for example, by the sub-property nodes such as “recipient name” and “message address.”

**[0179]** In some examples, ontology 760 includes all the domains (and hence actionable intents) that the digital assistant is capable of understanding and acting upon. In some examples, ontology 760 is modified, such as by adding or removing entire domains or nodes, or by modifying relationships between the nodes within the ontology 760.

**[0180]** In some examples, nodes associated with multiple related actionable intents are clustered under a “super domain” in ontology 760. For example, a “travel” super-domain

includes a cluster of property nodes and actionable intent nodes related to travel. The actionable intent nodes related to travel includes “airline reservation,” “hotel reservation,” “car rental,” “get directions,” “find points of interest,” and so on. The actionable intent nodes under the same super domain (e.g., the “travel” super domain) have many property nodes in common. For example, the actionable intent nodes for “airline reservation,” “hotel reservation,” “car rental,” “get directions,” and “find points of interest” share one or more of the property nodes “start location,” “destination,” “departure date/time,” “arrival date/time,” and “party size.”

**[0181]** In some examples, each node in ontology 760 is associated with a set of words and/or phrases that are relevant to the property or actionable intent represented by the node. The respective set of words and/or phrases associated with each node are the so-called “vocabulary” associated with the node. The respective set of words and/or phrases associated with each node are stored in vocabulary index 744 in association with the property or actionable intent represented by the node. For example, returning to FIG. 7B, the vocabulary associated with the node for the property of “restaurant” includes words such as “food,” “drinks,” “cuisine,” “hungry,” “eat,” “pizza,” “fast food,” “meal,” and so on. For another example, the vocabulary associated with the node for the actionable intent of “initiate a phone call” includes words and phrases such as “call,” “phone,” “dial,” “ring,” “call this number,” “make a call to,” and so on. The vocabulary index 744 optionally includes words and phrases in different languages.

**[0182]** Natural language processing module 732 receives the token sequence (e.g., a text string) from STT processing module 730, and determines what nodes are implicated by the words in the token sequence. In some examples, if a word or phrase in the token sequence is found to be associated with one or more nodes in ontology 760 (via vocabulary index 744), the word or phrase “triggers” or “activates” those nodes. Based on the quantity and/or relative importance of the activated nodes, natural language processing module 732 selects one of the actionable intents as the task that the user intended the digital assistant to perform. In some examples, the domain that has the most “triggered” nodes is selected. In some examples, the domain having the highest confidence value (e.g., based on the relative importance of its various triggered nodes) is selected. In some examples, the domain is selected based on a combination of the number and the importance of the triggered nodes. In some examples, additional factors

are considered in selecting the node as well, such as whether the digital assistant has previously correctly interpreted a similar request from a user.

**[0183]** User data 748 includes user-specific information, such as user-specific vocabulary, user preferences, user address, user's default and secondary languages, user's contact list, and other short-term or long-term information for each user. In some examples, natural language processing module 732 uses the user-specific information to supplement the information contained in the user input to further define the user intent. For example, for a user request "invite my friends to my birthday party," natural language processing module 732 is able to access user data 748 to determine who the "friends" are and when and where the "birthday party" would be held, rather than requiring the user to provide such information explicitly in his/her request.

**[0184]** Other details of searching an ontology based on a token string is described in U.S. Utility Application Serial No. 12/341,743 for "Method and Apparatus for Searching Using An Active Ontology," filed December 22, 2008, the entire disclosure of which is incorporated herein by reference.

**[0185]** In some examples, once natural language processing module 732 identifies an actionable intent (or domain) based on the user request, natural language processing module 732 generates a structured query to represent the identified actionable intent. In some examples, the structured query includes parameters for one or more nodes within the domain for the actionable intent, and at least some of the parameters are populated with the specific information and requirements specified in the user request. For example, the user says "Make me a dinner reservation at a sushi place at 7." In this case, natural language processing module 732 is able to correctly identify the actionable intent to be "restaurant reservation" based on the user input. According to the ontology, a structured query for a "restaurant reservation" domain includes parameters such as {Cuisine}, {Time}, {Date}, {Party Size}, and the like. In some examples, based on the speech input and the text derived from the speech input using STT processing module 730, natural language processing module 732 generates a partial structured query for the restaurant reservation domain, where the partial structured query includes the parameters {Cuisine = "Sushi"} and {Time = "7pm"}. However, in this example, the user's utterance

contains insufficient information to complete the structured query associated with the domain. Therefore, other necessary parameters such as {Party Size} and {Date} is not specified in the structured query based on the information currently available. In some examples, natural language processing module 732 populates some parameters of the structured query with received contextual information. For example, in some examples, if the user requested a sushi restaurant “near me,” natural language processing module 732 populates a {location} parameter in the structured query with GPS coordinates from the user device.

**[0186]** In some examples, natural language processing module 732 passes the generated structured query (including any completed parameters) to task flow processing module 736 (“task flow processor”). Task flow processing module 736 is configured to receive the structured query from natural language processing module 732, complete the structured query, if necessary, and perform the actions required to “complete” the user’s ultimate request. In some examples, the various procedures necessary to complete these tasks are provided in task flow models 754. In some examples, task flow models 754 include procedures for obtaining additional information from the user and task flows for performing actions associated with the actionable intent.

**[0187]** As described above, in order to complete a structured query, task flow processing module 736 needs to initiate additional dialogue with the user in order to obtain additional information, and/or disambiguate potentially ambiguous utterances. When such interactions are necessary, task flow processing module 736 invokes dialogue flow processing module 734 to engage in a dialogue with the user. In some examples, dialogue flow processing module 734 determines how (and/or when) to ask the user for the additional information and receives and processes the user responses. The questions are provided to and answers are received from the users through I/O processing module 728. In some examples, dialogue flow processing module 734 presents dialogue output to the user via audio and/or visual output, and receives input from the user via spoken or physical (e.g., clicking) responses. Continuing with the example above, when task flow processing module 736 invokes dialogue flow processing module 734 to determine the “party size” and “date” information for the structured query associated with the domain “restaurant reservation,” dialogue flow processing module 734 generates questions such as “For how many people?” and “On which day?” to pass to the user. Once answers are received

from the user, dialogue flow processing module 734 then populates the structured query with the missing information, or pass the information to task flow processing module 736 to complete the missing information from the structured query.

**[0188]** Once task flow processing module 736 has completed the structured query for an actionable intent, task flow processing module 736 proceeds to perform the ultimate task associated with the actionable intent. Accordingly, task flow processing module 736 executes the steps and instructions in the task flow model according to the specific parameters contained in the structured query. For example, the task flow model for the actionable intent of “restaurant reservation” includes steps and instructions for contacting a restaurant and actually requesting a reservation for a particular party size at a particular time. For example, using a structured query such as: {restaurant reservation, restaurant = ABC Café, date = 3/12/2012, time = 7pm, party size = 5}, task flow processing module 736 performs the steps of: (1) logging onto a server of the ABC Café or a restaurant reservation system such as OPENTABLE®, (2) entering the date, time, and party size information in a form on the website, (3) submitting the form, and (4) making a calendar entry for the reservation in the user’s calendar.

**[0189]** In some examples, task flow processing module 736 employs the assistance of service processing module 738 (“service processing module”) to complete a task requested in the user input or to provide an informational answer requested in the user input. For example, service processing module 738 acts on behalf of task flow processing module 736 to make a phone call, set a calendar entry, invoke a map search, invoke or interact with other user applications installed on the user device, and invoke or interact with third-party services (e.g., a restaurant reservation portal, a social networking website, a banking portal, etc.). In some examples, the protocols and application programming interfaces (API) required by each service are specified by a respective service model among service models 756. Service processing module 738 accesses the appropriate service model for a service and generate requests for the service in accordance with the protocols and APIs required by the service according to the service model.

**[0190]** For example, if a restaurant has enabled an online reservation service, the restaurant submits a service model specifying the necessary parameters for making a reservation and the APIs for communicating the values of the necessary parameter to the online reservation service.

When requested by task flow processing module 736, service processing module 738 establishes a network connection with the online reservation service using the web address stored in the service model, and send the necessary parameters of the reservation (e.g., time, date, party size) to the online reservation interface in a format according to the API of the online reservation service.

**[0191]** In some examples, natural language processing module 732, dialogue flow processing module 734, and task flow processing module 736 are used collectively and iteratively to infer and define the user's intent, obtain information to further clarify and refine the user intent, and finally generate a response (i.e., an output to the user, or the completion of a task) to fulfill the user's intent. The generated response is a dialogue response to the speech input that at least partially fulfills the user's intent. Further, in some examples, the generated response is output as a speech output. In these examples, the generated response is sent to speech synthesis module 740 (e.g., speech synthesizer) where it can be processed to synthesize the dialogue response in speech form. In yet other examples, the generated response is data content relevant to satisfying a user request in the speech input.

**[0192]** Speech synthesis module 740 is configured to synthesize speech outputs for presentation to the user. Speech synthesis module 740 synthesizes speech outputs based on text provided by the digital assistant. For example, the generated dialogue response is in the form of a text string. Speech synthesis module 740 converts the text string to an audible speech output. Speech synthesis module 740 uses any appropriate speech synthesis technique in order to generate speech outputs from text, including, but not limited, to concatenative synthesis, unit selection synthesis, diphone synthesis, domain-specific synthesis, formant synthesis, articulatory synthesis, hidden Markov model (HMM) based synthesis, and sinewave synthesis. In some examples, speech synthesis module 740 is configured to synthesize individual words based on phonemic strings corresponding to the words. For example, a phonemic string is associated with a word in the generated dialogue response. The phonemic string is stored in metadata associated with the word. Speech synthesis model 740 is configured to directly process the phonemic string in the metadata to synthesize the word in speech form.

**[0193]** In some examples, instead of (or in addition to) using speech synthesis module 740, speech synthesis is performed on a remote device (e.g., the server system 108), and the synthesized speech is sent to the user device for output to the user. For example, this can occur in some implementations where outputs for a digital assistant are generated at a server system. And because server systems generally have more processing power or resources than a user device, it is possible to obtain higher quality speech outputs than would be practical with client-side synthesis.

**[0194]** Additional details on digital assistants can be found in the U.S. Utility Application No. 12/987,982, entitled “Intelligent Automated Assistant,” filed January 10, 2011, and U.S. Utility Application No. 13/251,088, entitled “Generating and Processing Task Items That Represent Tasks to Perform,” filed September 30, 2011, the entire disclosures of which are incorporated herein by reference.

#### 4. Processes for Intelligent List Reading

**[0195]** FIGs. 8A-D illustrate process 800 for operating a digital assistant to perform intelligent list reading, according to various examples. FIGs. 9A-D illustrate intelligent list reading performed by digital assistant 905 implemented on user device 903 in response to spoken interaction from user 901, according to various examples. Process 800 is performed, for example, using one or more electronic devices implementing a digital assistant. In some examples, the process is performed at a client-server system (e.g., system 100) implementing a digital assistant. In some examples, the process is performed at a user device (e.g., device 104, 200, 400, or 600). In process 800, some blocks are, optionally, combined, the order of some blocks are, optionally, changed, and some blocks are, optionally, omitted. Further, it should be recognized that in some examples, only a subset of the features described below with reference to FIGs. 8A-D is performed in process 800.

**[0196]** At block 802, a spoken user request is received (e.g., at I/O processing module 728 and via microphone 213). The spoken user request is associated with a user and is addressed to the digital assistant implemented on the user device. Additionally, the spoken user request is associated with a plurality of data items. For example, the spoken user request is a request to

obtain and provide a list of restaurants (“What’s good to eat?”) or a list of song titles (“Recommend me some music.”).

**[0197]** At block 804, a determination is made (e.g., by natural language processing module 732) as to whether a degree of specificity of the spoken user request is less than a threshold level. The determination is based on the domain corresponding to the spoken user request, the size of the metadata associated with data items that satisfy the spoken user request, the degree of familiarity associated with data items that satisfy the spoken user request, or the number of parameters defined in the spoken user request. Each of these factors is discussed in greater detail below. Based on the determination, the requested information is presented in a manner that optimizes user experience. In particular, a spoken user request having a degree of specificity that is less than a threshold level indicates a vague user request that corresponds to a user intent of wanting to explore content. In other words, an inference is made from the vague user request that the user likely does not have any specific data items in mind, but rather wishes to explore available content and obtain a recommendation. In contrast, a spoken user request having a degree of specificity that is greater than a threshold level indicates a more specific user request that corresponds to a user intent of wanting to obtain specific data items. The digital assistant thus tailors the response according to the appropriate inferred user intent.

**[0198]** Block 804 includes determining a user intent (e.g., actionable intent) corresponding to the spoken user request. The user intent is determined in a manner described above with respect to FIGs. 7A-C. In particular, as discussed above, determining the user intent includes determining a domain corresponding to the spoken user request. Whether the spoken user request has a degree of specificity less than a threshold value depends on the specific domain corresponding to the spoken user request. In particular, the threshold value is established based on the specific domain corresponding to the spoken user request. For example, certain domains, such as the “music,” “restaurant,” or “business search” domains, encompass subject matter that is more predisposed to content exploration. The threshold value for these domains is thus higher where the digital assistant is more likely to infer from the spoken user request that the user wishes to explore available content and obtain a recommendation. Accordingly, in these examples, the degree of specificity of the spoken user request is more likely to be less than the threshold level. Other domains, such as the “sport schedules,” “movies playing,” “movie info,”



“alarm list,” “spelling,” “rhyming,” or “dictionary definition” domains, encompass subject matter that is more specific and better defined. The threshold value for these domains is thus lower where the digital assistant is more likely to infer from the spoken user request that the user wishes to obtain specific data items. Accordingly, in these examples, the degree of specificity of the spoken user request is more likely not to be less than the threshold level.

**[0199]** Additionally, block 804 includes determining the size of the metadata associated with data items that satisfy the spoken user request. In particular, the threshold value is established based on the determined size of the metadata associated with data items that satisfy the spoken user request. The size of the metadata is based on the number of parameters or attributes associated with the data items. For example, data items that satisfy the spoken user request “What’s good to eat?” are associated with a large amount of metadata. Specifically, each data item representing a specific restaurant includes metadata defining various parameters, such as the name of the restaurant, cuisine, price range, phone number, location (e.g., address), user rating, hours of operation, whether reservations are accepted, and the like. The threshold value for these types of spoken user requests is thus higher where the digital assistant is more likely to infer from the spoken user request that the user wishes to explore available content and obtain a recommendation. Accordingly, in these examples, the degree of specificity of the spoken user request is more likely to be less than the threshold level. Conversely, data items that satisfy the spoken user request “What does ‘plot’ mean?” are associated with a small amount of metadata. Specifically, each data item representing a dictionary definition of the word “plot” includes metadata defining a limited number of parameters such as the source of the dictionary definition. The threshold value for these types of spoken user requests is thus lower where the digital assistant is more likely to infer from the spoken user request that the user wishes to obtain specific data items. Accordingly, in these examples, the degree of specificity of the spoken user request is more likely not to be less than the threshold level.

**[0200]** Block 804 further includes determining a degree of familiarity associated with data items that satisfy the spoken user request. In particular, the threshold value is established based on the determined degree of familiarity associated with data items that satisfy the spoken user request. The degree of familiarity represents how familiar the user is with the data items being requested. For example, data items associated with the user are determined to have a higher

degree of familiarity. Such data items include, for example, music items from the user's personal music library, contact information from the user's contacts, or applications installed on the user device. Thus, for spoken user requests such as "Tell me all contacts for 'John,'" "Which Eagles songs do I have?" or "Tell me the alarms I have set," the threshold value is lower where the digital assistant is more likely to infer that the user wishes to obtain specific data items. Accordingly, in these examples, the degree of specificity of the spoken user request is more likely to be less than the threshold level. In other examples, the degree of familiarity is based on how recent or how frequent the user accesses or interacts with the requested data items. For example, if the user frequently browses through many Katy Perry music albums or recently listened to many Taylor Swift songs, then the degree of familiarity associated with these data items would be higher. Thus, for spoken user requests such as "What are some Katy Perry albums?" or "List me some Taylor Swift songs," the threshold value is lower where the digital assistant is more likely to infer that the user wishes to obtain specific data items. Accordingly, in these examples, the degree of specificity of the spoken user request is more likely not to be less than the threshold level.

**[0201]** In some examples, block 804 includes determining the number of parameters defined in the spoken user request. In particular, the degree of specificity is based on the determined number of parameters. The parameters represent the properties in the determined domain corresponding to spoken user request. A greater number of parameters defined in the spoken user request are associated with a higher degree of specificity, whereas a lower number of parameters defined in the spoken user request are associated with a lower degree of specificity. For example, the spoken user request "Recommend me some music" corresponds to the "music" domain. However, no specific parameters of the music domain are defined in the spoken user request. Based on the absence of any parameters defined in the spoken user request, the degree of specificity for this spoken user request is determined to be low. Accordingly, in this example, the degree of specificity of the spoken user request is more likely to be less than the threshold level. In another example, the spoken user request is "Recommend me some good R&B songs by Beyonce." In this example, the spoken user request defines several parameters associated with the music domain, including {genre}=R&B, {artist}=Beyonce, and {rating}=good. Based on having many defined parameters, the degree of specificity for the spoken user request is

determined to be higher. Accordingly, in this example, the degree of specificity of the spoken user request is more likely not to be less than the threshold level.

**[0202]** In some examples, block 804 includes determining the number of possible data items that satisfy the spoken user request. In these examples, the degree of specificity is based on the determined number of possible data items. In particular, a greater number of possible data items are associated with a lower degree of specificity, whereas a smaller number of possible data items are associated with a higher degree of specificity. For example, the spoken user request “What are some Michael Jackson songs?” is associated with a larger number of possible data items and thus have a lower degree of specificity. Accordingly, in this example, the degree of specificity of the spoken user request is more likely to be less than the threshold level. In comparison, the spoken user request “Tell me some Orianthi songs” is associated with a smaller number of possible data items and thus have a higher degree of specificity. Accordingly, in this example, the degree of specificity of the spoken user request is more likely not to be less than the threshold level.

**[0203]** In response to determining that a degree of specificity of the spoken user request is less than a threshold level, one or more of blocks 806-838 are performed. Specifically, in response to determining that a degree of specificity of the spoken user request is less than a threshold level, one or more of blocks 806-838 are performed automatically without additional input from the user. In general, blocks 806-838 involve helping the user explore a diverse range of content by recommending data items in a more focused manner.

**[0204]** At block 806, one or more attributes (e.g., parameters) related to the user request are determined (e.g., by natural language processing module 732, task flow processing module 736, and/or service processing module 738). In particular, the one or more attributes determined at block 806 are not defined in the spoken user request of block 802. Rather, the digital assistant intelligently determines the one or more attributes to help narrow down the user request and recommend the most relevant data items to the user.

**[0205]** In some examples, the one or more attributes are determined based on the frequency with which an attribute is specified in a plurality of previous user requests from a plurality of users. For example, the previous user requests of many users are analyzed to determine the most

frequently requested attributes with respect to a given domain. The most frequently requested attributes are then included in the one or more attributes. For example, if it is determined that “Katy Perry” is frequently specified in previous user requests related to the “music” domain, then the one or more attributes are determined to include the attribute “Katy Perry.”

**[0206]** In some examples, the one or more attributes are determined based on the most relevant attributes with respect to the given domain. For example, songs that are more recently released are more relevant than songs that have been released many years back. Thus, for user requests related to the “music” domain, the one or more attributes are determined to include a release date that is less than a predetermined number of years old. In another example, for user requests related to the “restaurant” domain, it is determined that restaurants located closer to the user’s current location can be more relevant than restaurants located much further away. Thus, in this example, the one or more attributes are determined to include a distance that is less than a predetermined number of miles from the user’s current location.

**[0207]** In some examples, the one or more attributes are determined based on a user profile associated with the spoken user request. The user profile indicates various characteristics associated with the user and is generated based on user data (e.g., user’s location, user music files, user emails/messages, etc.) and/or user input (e.g., user text input, user browsing history, user search history, etc.). The one or more attributes of block 806 are thus based on the characteristics indicated in the user profile. For example, based on the user profile indicating a Korean ethnicity, the one or more attributes include Korean cuisine for spoken user requests related to the “restaurant” domain. In another example, based on the user profile indicating a tendency to browse pop songs, the one or more attributes include the “pop” genre for spoken user requests related to the “music” domain.

**[0208]** In some examples, the one or more attributes are determined by randomly selecting an attribute from a category of attributes. This is desirable for helping the user to explore new or fresh data items that may pique the user’s interest. For example, based on the spoken user request “What’s good to eat,” a random selection from a set of cuisines (e.g., French, Chinese, Italian, French, Mexican, Seafood, New American, etc.) is made. The digital assistant thus provides (e.g., at block 812), for example, a recommended Seafood restaurant in one instance of

the spoken user request and a recommended Mexican restaurant in another separate instance of the same spoken user request.

**[0209]** At block 808, a list of data items is obtained based on the spoken user request and the one or more attributes. In particular, a structured query based on the determined user intent of the spoken user request is generated (e.g., by natural language processing module 732). The structured query includes the attributes defined in the spoken user request and the one or more attributes determined at block 806. A search is then performed (e.g., by task flow processing module 736 and/or service processing module 738) in accordance with the structured query to obtain a list of data items that satisfy the spoken user request. For example, based on the one or more attributes determined at block 806 and one or more attributes defined in the spoken user request of block 802, one or more information sources are searched to obtain the list of data items.

**[0210]** At block 810, a spoken response is generated (e.g., by task flow processing module 736, dialogue processing module 734, and/or speech synthesis module 740). The spoken response comprises a subset of the list of data items. In some examples, the subset of the list of data items has at most a predetermined number of data items. For example, the subset of the list of data items has no more than 1 or 2 data items. Restricting the number of data items presented to the user improves user experience. In particular, it enables the most relevant data item(s) to be recommended to the user and prevent the user from becoming overwhelmed by too many options.

**[0211]** In some examples, generating the spoken response includes generating a spoken preamble that provides some context to the subset of the list of data items to be presented. In particular, the spoken preamble describes an attribute of the one or more attributes determined at block 806. Further, the generated spoken response includes a description that specifies one or more additional attributes for each data item of the subset of the list of data items. In particular, each of the one or more additional attributes is not defined in the spoken user request of block 802 and is different from any of the one or more attributes of block 806. The description serves to provide additional details of each data item, which helps the user determine whether the recommended data item is acceptable or not.

**[0212]** In some examples, the spoken response is initially generated as a text response (e.g., with dialogue flow processing module 734) at block 810 and then converted to speech (e.g., with speech synthesis module 740) at block 812.

**[0213]** At block 812, the spoken response of block 810 is provided (e.g., using speech synthesis module 740, I/O processing module 728, and/or speaker 211). Providing the spoken response includes providing the spoken preamble followed by the subset of the list of data items in spoken form. In some examples, the spoken preamble is provided prior to providing the subset of the list of data items. Additionally, providing the spoken response includes providing a spoken prompt, as discussed in greater detail below. The spoken response is provided at the user device in the form of synthesized speech. Alternatively, the spoken response is provided in the form of audio data that is played by the user device.

**[0214]** Blocks 802 through 812 of process 800 are further described with reference to the examples illustrated in FIGs. 9A-B. In FIGs. 9A-B, user device 903 is similar or identical to devices 104, 200, 400, or 600 and includes any of the components of digital assistant system 700. Digital assistant 905 is implemented at least partially on user device 903 and at least partially on a server (e.g., DA server 106). With reference to FIG. 9A, user 901 provides spoken user request 902 “Hey Siri, what’s good to eat?” to digital assistant 905 of user device 903. Digital assistant 905 receives (block 802) spoken user request 902 and determines (block 804) whether a degree of specificity of spoken user request 902 is less than a threshold level. In this example, spoken user request 902 is vague since it does not define any specific parameters other than “good.” In addition, the request is broad since it is associated with a large number of data items. As discussed above in block 804, such characteristics correspond to a low degree of specificity of spoken user request 902. Furthermore, spoken user request 902 corresponds to the “restaurant” domain. In particular, spoken user request 902 is a request for restaurant data items, where each data item is associated with a significant amount of metadata. As discussed above in block 804, such additional characteristics correspond to a high threshold value associated with the degree of specificity. Based on the low degree of specificity of spoken user request 902 and the high threshold value associated with the degree of specificity, the degree of specificity of spoken user request 902 is determined (block 804) to be less than the threshold level. In response to this determination, digital assistant 905 determines (block 806) the attribute “Japanese cuisine” to

refine the request. Notably, “Japanese cuisine” is not an attribute defined in spoken user request 902. A search for Japanese restaurants is performed to obtain (block 808) a list of Japanese restaurants. The search is based on the determined attribute “Japanese cuisine” and the “restaurant” domain corresponding to spoken user request 902. In some examples, based on additional relevant parameters such as the distance from the user’s current location and popularity (e.g., user ratings), the search is further refined to obtain the list of Japanese restaurants. Spoken response 904 is then generated (block 810) and provided (block 812) to the user. As shown, spoken response 904 includes the Japanese restaurant “Gochi Japanese Fusion Tapas,” which is one (i.e., a subset) of the Japanese restaurants in the obtained list of Japanese restaurants. “Gochi Japanese Fusion Tapas” is selected from the list of Japanese restaurants based on factors such as popularity, relevance, distance for the user’s current location, or price. For example, it is the most popular restaurant within half a mile of the user’s current location. As shown, spoken response 904 includes the preamble “If you like Japanese food,” which indicates that the cuisine associated with the recommended restaurant is “Japanese.” Additionally, spoken response 904 includes the additional description “moderately priced” and “four-star user rating” about the recommended restaurant “Gochi Japanese Fusion Tapas,” which is helpful to the user for determining whether or not the recommendation is acceptable.

**[0215]** Turning now to the example shown in FIG. 9B, user 901 provides spoken user request 922 “Hey Siri, recommend me some music?” to digital assistant 905 of user device 903. Digital assistant 905 receives (block 802) spoken user request 922 and determines (block 804) whether a degree of specificity of spoken user request 922 is less than a threshold level. Similar to spoken user request 902 of FIG. 9A, spoken user request 922 is vague since it does not define any specific parameters related to “music.” Additionally, spoken user request 922 is broad since it is associated with a large number of data items. As discussed above in block 804, such characteristics correspond to a low degree of specificity of spoken user request 922. Moreover, spoken user request 922 corresponds to the “music” domain. In particular, spoken user request 922 is a request for music data items, where each data item is associated with a significant amount of metadata. As discussed above in block 804, such additional characteristics correspond to a high threshold level associated with the degree of specificity. Based on these factors, a threshold level associated with the degree of specificity is determined to be high. Based on the low degree of specificity of spoken user request 922 and the high threshold value

associated with the degree of specificity, the degree of specificity of spoken user request 922 is determined (block 804) to be less than the threshold level. In response to this determination, digital assistant 905 determines the attribute of genre = “alternative” music (block 806) to refine the request. Notably, the genre of “alternative” music is not an attribute defined in spoken user request 922. A search for alternative music is performed to obtain (block 808) a list of songs having the genre of “alternative” music. Spoken response 924 is then generated (block 810) and provided (block 812) to the user. As shown, spoken response 924 includes the alternative song “Under the Blacklight,” which is one (i.e., a subset) of the alternative songs in the obtained list of alternative songs. “Under the Blacklight” is selected from the list of alternative songs based on factors such as popularity, relevance, or recent release date. Spoken response 924 also includes the preamble “If you’re in the mood for Alternative,” which indicates that the genre associated with the recommended song is “alternative.” In particular, the preamble helps contextualize the data item “Under the Blacklight” that follows. Additionally, spoken response 924 includes the additional description “Rilo Kiley” to indicate the artist associated with the recommended song “Under the Blacklight. Specifically, as opposed to only naming song title in spoken response 924, providing additional information (e.g., the artist or the release date) about the recommend song makes the recommendation more meaningful to the user.

**[0216]** In some examples, the spoken response of blocks 810 and 812 optionally includes a spoken prompt. In particular, the spoken prompt indicates to the user that additional data items are available. For example, as shown in FIG. 9B, spoken response 924 includes the spoken prompt “Does that work?” Other examples of similar spoken prompts include “There’s more,” “Does that sound good?” or “There are other options.” The user is thus provided with the option to respond. For example, the microphone of the user device is turned on to receive audio input while or after providing the spoken response at block 812. The audio input is then analyzed to determine whether it contains a follow-up spoken user request. If it is determined that the audio input contains a follow-up spoken user request, a suitable response is generated and provided, as described in greater detail below.

**[0217]** In some examples, the spoken prompt is intelligently provided. In particular, the spoken prompt is initially provided for a predetermined number of times to a user to inform the user of the option to request for additional data items. The spoken prompt then ceases to be



provided after the predetermined number of times under the assumption that the user already understands the option to request for additional data items. For example, block 810 includes determining whether each of a predetermined number of previous spoken responses includes a spoken prompt indicating that additional data items are available. In response to determining that each of a predetermined number of previous spoken responses does not include a spoken prompt indicating that additional data items are available, the spoken prompt is generated and provided in the spoken response. Conversely, in response to determining that each of a predetermined number of previous spoken responses includes a spoken prompt indicating that additional data items are available, process 800 forgoes including the spoken prompt in the spoken response.

**[0218]** Turning now to block 814 of process 800 shown in FIG. 8B, speech input is received from the user (e.g., at I/O processing module 728 and via microphone 213). In some examples, the microphone of the user device is automatically (e.g., without further user input) turned on to receive speech input while or after providing the spoken response at block 812. In some examples, the microphone is turned on at all times. The speech input of block 814 is then received while the microphone is turned on. In other examples, the microphone of the user device is not automatically turned on while or after providing the spoken response at block 812. Instead, the microphone is turned on while or after the spoken response of block 812 is provided in response to receiving a user input. The speech input is thus received after the microphone is turned on in response to the user input. The speech input is responsive to the spoken response provided at block 812. For example, the speech input indicates an acceptance or rejection of the recommendation provided in the spoken response of block 812.

**[0219]** In some examples, the speech input is received while the spoken response is provided at block 812. In particular, the user interrupts the digital assistant during the spoken response and provides the spoken response. Upon detecting the speech input while the spoken response is being provided, the digital assistant ceases to provide the rest of the spoken response. Further, in response to receiving the speech input, one or more of blocks 816 to 838 are performed.

**[0220]** At block 816, a determination is made as to whether the speech input corresponds to a rejection of the subset of the list of data items. The determination is made using natural language

processing (e.g., with natural language processing module 732). For example, with reference to FIG. 9A, speech input 906 “Nah, something else” is received from the user in response to spoken response 904. The words and phrases in speech input 906 are parsed and analyzed to determine whether they correspond to a user intent of rejecting the subset of the list of data items provided in spoken response 904. In particular, it is determined that the phrases “nah” and “something else” both correspond to a negative response and based on this determination, speech input 906 is determined to correspond to a rejection of the subset of the list of data items. Other examples of speech input corresponding to a rejection of the subset of the list of data items include “No,” “Nope,” “Anything but that,” “Don’t like...,” “Try again,” “Any others?” or the like. In response to determining that the speech input of block 814 corresponds to a rejection of the subset of the list of data items, one or more of blocks 818 to 834 are performed. Specifically, in response to determining that the speech input of block 814 corresponds to a rejection of the subset of the list of data items, one or more of blocks 818 to 834 are performed automatically without additional input from the user.

**[0221]** At block 818, one or more second attributes related to the spoken user request of block 802 are determined (e.g., by natural language processing module 732, task flow processing module 736, and/or service processing module 738). The one or more second attributes are different from the one or more attributes of block 806. Additionally, the one or more second attributes may not be defined in the spoken user request of block 802. Block 818 is similar to block 806, except that different attributes related to the spoken user request are determined. For example, referring back to FIG. 9A, the digital assistant determines one or more second attributes related to restaurants to refine spoken user request 902 and provide a different recommendation to the user. In this example, the digital assistant determines the second attribute “Chinese cuisine,” which is different from the attribute “Japanese cuisine” determined at block 806. Moreover, “Chinese cuisine” is not defined in spoken user request 902.

**[0222]** At block 820, a second list of data items is obtained (e.g., using natural language processing module 732, task flow processing module 736, and/or service processing module 738) based on the user request and the one or more second attributes. Block 820 is similar to block 808, except that the second list of data items is based on the one or more second attributes of block 818 rather than the one or more attributes of block 806. For example, based on the one or

more second attributes determined at block 818 and/or one or more attributes defined in the spoken user request of block 802, one or more information sources are searched to obtain the second list of data items. In the example shown in FIG. 9A, a search for Chinese restaurants is performed to obtain a list of Chinese restaurants. The search is based on the second attribute “Chinese cuisine” and the “restaurant” domain corresponding to spoken user request 902. In some examples, based on additional relevant parameters such as the distance from the user’s current location and popularity (e.g., user ratings), the search is further refined to obtain the list of Chinese restaurants.

**[0223]** At block 822, a second spoken response is generated (e.g., by task flow processing module 736, dialogue processing module 734, and/or speech synthesis module 740). The second spoken response comprises a subset of the second list of data items. Block 822 is similar to block 810, except that the second spoken response comprises a subset of the second list of data items rather than a subset of the first list of data items.

**[0224]** At block 824, the second spoken response of block 822 is provided (e.g., using speech synthesis module 740, I/O processing module 728, and/or speaker 211). Block 824 is similar to block 812.

**[0225]** For example, in FIG. 9A, second spoken response 908 is generated (block 822) and provided (block 824) to the user in response to speech input 906. As shown, second spoken response 908 includes the Chinese restaurant “Mandarin Gourmet,” which is one (i.e., a subset) of the Chinese restaurants in the obtained list of Chinese restaurants (e.g., second list of data items obtained at block 820). “Mandarin Gourmet” is selected from the list of Chinese restaurants based on factors such as popularity, relevance, distance for the user’s current location, or price. Second spoken response 908 includes the additional description “Chinese restaurant,” “close by,” and “moderately priced” regarding the recommended restaurant “Mandarin Gourmet,” which is helpful to the user for determining whether to accept the recommendation. Although in this example, the preamble does not indicate the one or more second attributes of block 818, it should be recognized that in other examples, second spoken response 908 is generated at block 822 with a similar spoken preamble as spoken response 904 to help contextualize the recommended restaurant that follows. For example, second spoken

response 908 could instead be “How about Chinese cuisine then? There’s Mandarin Gourmet close by that is moderately priced.”

**[0226]** It should be appreciated that, in some examples, blocks 814 through 824 of process 800 are repeated multiple times to help the user explore the available content. For example, through one or more cycles of blocks 814 through 824, the user provides several speech inputs (block 814) rejecting the various recommendations of the digital assistant and each time, the digital assistant responds by provide a different recommendation (blocks 820-824) by determining different attributes (block 818) related to the original spoken request. In this way, the digital assistant helps the user explore a large diverse body of content (e.g., restaurant).

**[0227]** In some examples, the digital assistant requests guidance from the user to assist with providing a suitable recommendation. For example, referring back to block 816, in response to determining that the speech input of block 814 corresponds to a rejection of the subset of the list of data items, one or more of blocks 826 to 834 are performed. Blocks 826 to 834 are directed to obtaining guidance from the user and providing a recommendation to the user based on the obtained guidance.

**[0228]** At block 826, a spoken prompt for the user to provide additional attributes to refine the spoken user request is provided. The spoken prompt serves as a request for guidance from the user to help the digital assistant provide recommendations that better match the user’s current preferences. In some examples, the spoken prompt is provided in response to receiving a predetermined number of speech inputs rejecting the recommendations of the digital assistant. For example, in FIG. 9A, speech input 910 “No...” is received in response to providing second spoken response 908. In this example, the digital assistant determines that two speech inputs (e.g., 906 and 910) have been received rejecting the digital assistant’s restaurant recommendations (e.g., 904 and 908) and in response to the determination, the digital assistant provides spoken prompt 912 “OK. What cuisine are you in the mood for?” Specifically, spoken prompt 912 asks the user to provide a preferred cuisine attribute to help the digital assistant refine spoken user request 902.

**[0229]** At block 828, a second speech input is received (e.g., at I/O processing module 728 and via microphone 213). The second speech input is responsive to the spoken prompt of block

826. For example, the second speech input defines one or more attributes related to the spoken user request of block 902. As shown in FIG. 9A, second speech input 914 “How about Mexican?” is received. In this example, second speech input 914 defines the attribute “Mexican” cuisine, which helps digital assistant 905 refine spoken user request 902.

**[0230]** In some examples, a determination is made (e.g., using natural language processing module 732) as to whether the second speech input defines one or more attributes related to the spoken user request. In response to determining that the second speech input defines one or more attributes related to the spoken user request, block 830 is performed. Conversely, in response to determining that the second speech input does not define one or more attributes related to the spoken user request, the digital assistant either provides the spoken prompt of block 826 again or provides a spoken response indicating an error.

**[0231]** At block 830, a third list of data items is obtained (e.g., using natural language processing module 732, task flow processing module 736, and/or service processing module 738) based on the user request and one or more attributes defined in the second speech input. Block 830 is similar to block 808, except that the third list of data items is based on the one or more attributes defined in the second speech input of block 828 rather than the one or more attributes of block 806. For example, based on the one or more attributes defined in the second speech input of block 828 and/or one or more attributes defined in the spoken user request of block 802, one or more information sources are searched to obtain the third list of data items. In the example shown in FIG. 9A, a search for Mexican restaurants is performed to obtain a list of Mexican restaurants. The search is based on the attribute “Mexican” cuisine defined in second speech input 914 and the “restaurant” domain corresponding to spoken user request 902. In some examples, based on additional relevant parameters such as the distance from the user’s current location and popularity (e.g., user ratings), the search is further refined to obtain the list of Mexican restaurants.

**[0232]** At block 832, a third spoken response is generated (e.g., by task flow processing module 736, dialogue processing module 734, and/or speech synthesis module 740). The third spoken response comprises a subset of the third list of data items. Block 832 is similar to block

810, except that the third spoken response comprises a subset of the third list of data items rather than a subset of the list of data items.

**[0233]** At block 834, the third spoken response is provided (e.g., using speech synthesis module 740, I/O processing module 728, and/or speaker 211). Block 834 is similar to block 812.

**[0234]** Referring back to the example of FIG. 9A, third spoken response 916 is generated (block 832) and provided (block 834) to the user. As shown, third spoken response 916 includes the Mexican restaurant “Aqui Cal-Mex,” which is one (i.e., a subset) of the Mexican restaurants in the obtained list of Mexican restaurants (e.g., third list of data items obtained at block 830). “Aqui Cal-Mex” is selected from the list of Mexican restaurants based on factors such as popularity, relevance, distance from the user’s current location, or price. Third spoken response 916 includes the additional description “rated four stars,” “close by,” and “well known for their Industrial Strength Margaritas” regarding the recommended restaurant, which is helpful to the user for determining whether to accept the recommendation.

**[0235]** With reference back to block 816, in response to determining that the speech input of block 814 does not correspond to a rejection of the subset of the list of data items, one or more of blocks 836 or 838 are performed. At block 836 (FIG. 8C), a determination is made as to whether the speech input corresponds to an acceptance of a data item in the subset of the list of data items. The determination is made using natural language processing (e.g., with natural language processing module 732). For example, the words and phrases in the speech input are parsed and analyzed to determine whether they correspond to a user intent of accepting a data item in the subset of the list of data items provided in the spoken response. In a specific example, it is determined that the phrases “Yes” or “Cool” correspond to a positive response and based on this determination, the speech input is determined to correspond to an acceptance of a data item in the subset of the list of data items. Other examples of speech input corresponding to an acceptance of the subset of the list of data items include “Sounds good,” “Yeah,” “Good,” “Sure,” “Why not,” “Let’s do it,” or the like.

**[0236]** In response to determining that the speech input of block 814 does not correspond to an acceptance of a data item in the subset of the list of data items, process 800 determines that the speech input of block 814 corresponds to a new spoken user request and thus process 800

returns to block 804 of FIG. 8A. Alternatively, in response to determining that the speech input corresponds to an acceptance of a data item in the subset of the list of data items, block 838 is performed. Specifically, in response to determining that the speech input corresponds to an acceptance of a data item in the subset of the list of data items, block 838 is performed automatically without additional input from the user. At block 838, content associated with the accepted data item is provided (e.g., using task flow processing module 736 and service processing module 738).

**[0237]** Blocks 836 and 838 are further described with reference to the examples illustrated in FIGs. 9A and 9B. For example, in FIG. 9A, speech input 918 “That sounds good” is received from the user in response to third spoken response 916 recommending the “Aqui Cal-Mex” Mexican restaurant. In this example, it is determined that the phrase “sounds good” corresponds to a positive response and based on this determination, speech input 918 is determined (block 836) to correspond to an acceptance of the recommended “Aqui Cal-Mex” Mexican restaurant. In response to this determination, the digital assistant provides content associated with the “Aqui Cal-Mex” Mexican restaurant. For example, as indicated in spoken response 920 of FIG. 9A, digital assistant 905 retrieves the address associated with the “Aqui Cal-Mex” Mexican restaurant and display, on user device 903, directions to the restaurant from the user’s current location. Alternatively, in another example, digital assistant 905 retrieves a webpage that provides additional information (e.g., user reviews, menu, photos, etc.) related to the “Aqui Cal-Mex” Mexican restaurant and display the webpage on user device 903.

**[0238]** In another example illustrated in FIG. 9B, user 901 provides speech input 926 “Sure,” in response to spoken response 924 recommending the alternative song “Under the Blacklight” by Rilo Kiley. In this example, it is determined that the phrase “Sure” corresponds to a positive response and based on this determination, speech input 926 is determined (block 836) to correspond to an acceptance of the recommended song “Under the Blacklight.” In response to this determination, digital assistant 905 provides content associated with the alternative song “Under the Blacklight.” For example, as indicated by spoken response 928 and musical notes 959 depicted in FIG. 9B, audio data corresponding to song “Under the Blacklight” is retrieved by digital assistant 905 and the audio data is played on user device 903. Alternatively, in another example, information related to the alternative song “Under the Blacklight” is retrieved by digital

assistant 905. For example, a webpage that provides critic reviews or user reviews of the song “Under the Blacklight” and the band “Rilo Kiley” is retrieved and displayed on user device 903.

**[0239]** With reference back to block 804, in response to determining that a degree of specificity of the spoken user request is not less than a threshold level, one or more of blocks 840-850 are performed. Blocks 840-850 are directed to retrieving a specific list of requested data items and reading the list to the user in a manner than is easily comprehended and retained by the user.

**[0240]** At block 840, a fourth list of data items is obtained (e.g., using natural language processing module 732, task flow processing module 736, and/or service processing module 738) based on the spoken user request. In particular, a structured query based on the determined user intent of the spoken user request is generated (e.g., by natural language processing module 732). The structured query includes the attributes defined in the spoken user request. Unlike in block 806, because the spoken use request is determined to be sufficiently specific (at block 804), additional attributes are not determined to further refine the spoken user request. A search is then performed (e.g., by task flow processing module 736 and/or service processing module 738) in accordance with the structured query to obtain the fourth list of data items that satisfy the spoken user request. For example, based on the one or more attributes defined in the spoken user request of block 802, one or more information sources are searched to obtain the fourth list of data items.

**[0241]** At block 842, a determination is made as to whether the number of data items in the fourth list of data items exceeds a predetermined number. The predetermined number is the maximum number of data items that the user is inferred to be able to easily comprehend and retain via voice interaction. User experience is thus negatively impacted if more than the predetermined number of data items are provided to the user in the spoken response (e.g., at blocks 846 or 850).

**[0242]** The predetermined number is based on a number of factors. For example, the predetermined number is based on a degree of familiarity of the user with data items in the fourth list of data items. If the user is more familiar with the data items in the fourth list of data items (higher degree of familiarity), the predetermined number is greater where a larger number of data



items is provided to the user in the spoken response. This is because the user is able to comprehend and retain a larger number of data items if the user is already familiar with the data items. Conversely, if the user is less familiar with the data items in the fourth list of data items (lower degree of familiarity), the predetermined number is lower where a smaller number of data items are provided to the user in the spoken response.

**[0243]** In some examples, the degree of familiarity is determined based on whether the data items in the fourth list of data items are associated with the user. For example, data items obtained from the user device (e.g., contact information obtained from contacts module 237) are determined to have a high degree of familiarity. Similarly, data items obtained from a library associated with the user (e.g., the user's personal media library) are determined to have a high degree of familiarity. In other examples, the degree of familiarity is determined based on the frequency at which the data items in the fourth list of data items have been previously requested or reviewed by the user. For example, it is determined from usage logs that the user frequently searches and listens to Katy Perry songs. Thus, if the fourth list of data items contains a list of Katy Perry songs, the degree of familiarity of the user with data items in the fourth list of data items is determined to be high.

**[0244]** In some examples, the predetermined number is based on the amount of metadata associated with each data item in the fourth list of data items. In particular, the predetermined number is determined to be greater if the amount of metadata associated with each data item in the fourth list of data items is small. A smaller amount of metadata corresponds to less information that is to be provided with each data item and thus more data items are provided to the user without overwhelming the user. Conversely, the predetermined number is determined to be lower if the amount of metadata associated with each data item in the fourth list of data items is large.

**[0245]** In some examples, the predetermined number is based on a cognitive load associated with the user at the time the spoken user request is received. The cognitive load refers to the total amount of mental effort being used in the working memory of the user. In some examples, the cognitive load associated with the user is inferred based on the number of activities and/or the types of activities the user engages in. For example, a greater number of activities or

mentally challenging types of activities is associated with a higher cognitive load. The number of activities and the types of activities that the user is engaged in are determined based on context information received by the user device and context information generated by or stored on the user device. For example, based on data generated by the accelerometers and GPS sensor of the user device, the user device detects that the user is accelerating and traveling in a manner consistent with being in a moving car. Further, the user device detects a Bluetooth pairing between the user device and a device associated with the car. Based on this context information, the user device determines that the user is driving a car and infers a corresponding amount of cognitive load associated with the user. In another example, the user device determines that the user is watching a video based on detecting a corresponding operating status associated with the media application of the user device. Based on this context information, the user device infers a corresponding amount of cognitive load associated with the user. The predetermined number is determined to be higher if the cognitive load associated with the user is small. A smaller cognitive load indicates that the user has greater mental capacity to process information and thus more data items are provided to the user without overwhelming the user. Conversely, the predetermined number is determined to be lower if the cognitive load associated with the user is large.

**[0246]** In response to determining that the number of data items in the fourth list of data items exceeds a predetermined number, one or more of blocks 844-846 are performed. Blocks 844-846 are directed to generating and providing a response containing only a subset of the fourth list of data items. Providing only a subset rather than the entire fourth list of data items is desirable to prevent overwhelming the user with too much information and to enhance the amount of information retained by the user.

**[0247]** At block 844, a fourth spoken response is generated (e.g., by task flow processing module 736, dialogue processing module 734, and/or speech synthesis module 740). The fourth spoken response comprises a subset of the fourth list of data items. Block 844 is similar to block 810, described above. In some examples, the number of data items in the subset of the fourth list of data items is less than or equal to the predetermined number. Restricting the number of data items presented in the fourth spoken response based on the predetermined number improves user

experience. In particular, it enables the most relevant data item(s) to be presented to the user first and prevent the user from becoming overwhelmed by too much information at one time.

**[0248]** In some examples, block 844 includes selecting the subset of the fourth list of data items from the fourth list of data items based on an attribute defined in the spoken user request. For example, the attribute defined in the spoken user request imposes an order on the fourth list of data items. The subset of the fourth list of data items is then selected based on the imposed order. Specifically, in one example, the spoken user request is “What are the most popular movies playing nearby?” In this example, the spoken user request defines the attribute “popular.” Based on this attribute, the digital assistant retrieves a list of movies playing near the user’s current location and rank the list of movies based on popularity (e.g., box office results, critic reviews, user ratings, etc.) The N highest ranked movies with respect to popularity in the list of movies is then selected as the subset of the list of movies (where N is an integer less than or equal to the predetermined number).

**[0249]** In other examples, the fourth list of data items has an inherent order and the subset of the fourth list of data items is selected from the subset of the fourth list of data items based on the inherent order. For example, as discussed below with reference to FIG. 9D, definitions retrieved from a dictionary reference have an inherent order. In these examples, based on the inherent order, the subset of the fourth list of data items includes the first N data items of the fourth list of data items, where N is an integer less than or equal to the predetermined number.

**[0250]** In some examples, generating the fourth spoken response includes generating a fourth spoken preamble that indicates a number of data items in the fourth list of data items. For example, in response to the spoken user request “What action movies are playing today?” the fourth spoken response includes the fourth spoken preamble “There are 5 action movies playing today...” The fourth spoken preamble is desirable to provide context for the amount of information that will follow and helps the user better capture and retain the information.

**[0251]** In some examples, the subset of the fourth list of data items is contrary to an expected result inferred from the spoken user request. In these examples, the fourth spoken preamble indicates that the subset of the fourth list of data items is contrary to an expected result inferred from the spoken user request. For example, the spoken user request is “What’s the closest gas

station?” In this example, an inference is made based on the word “closest” that the user is expecting only one result. However, the digital assistant determines that there are three gas stations that are approximately equidistant from the user’s current location. In this example, the fourth spoken preamble states “Well, there are 3 gas stations nearby...” Particularly, the interjection “Well” indicates that the subset of the fourth list of data items is contrary to the user’s expected result. The subtle interjection provides context for the information that will follow. Additionally, it enables the response to be more natural and personable, thereby improving user experience.

**[0252]** In some examples, generating the fourth spoken response includes generating a fourth spoken prompt indicating that additional data items are available. The fourth spoken prompt is, for example, after the subset of the fourth list of data items. For example, after the subset of the fourth list of data items, the fourth spoken prompt includes the phrase “Let me know if you’d like to hear the rest” (e.g., spoken response 936 in FIG. 9D) to indicate that additional data items are available and to prompt the user to request for the additional data items. Further, in some examples, the fourth spoken prompt indicates the number of remaining data items in the fourth list of data items. For example, after the subset of the fourth list of data items, the fourth spoken prompt includes the phrase “There are 4 more” (spoken response 932 in FIG. 9C) to indicate that there are four remaining data items in the fourth list of data items that have yet to be presented. The fourth spoken prompt thus provides the user the option to accept the data items that have already been presented or to request that the remaining data items be presented.

**[0253]** As discussed above, the fourth spoken prompt is intelligently provided. For example, a determination is made as to whether each of a predetermined number of previous spoken responses includes a spoken prompt indicating that additional data items are available. In response to determining that each of a predetermined number of previous spoken responses does not include a spoken prompt indicating that additional data items are available, the fourth spoken prompt is generated to indicate that additional data items are available. Conversely, in response to determining that each of a predetermined number of previous spoken responses includes a spoken prompt indicating that additional data items are available, fourth spoken prompt may not be generated to indicate that additional data items are available.

**[0254]** At block 846, the fourth spoken response is provided (e.g., using speech synthesis module 740, I/O processing module 728, and/or speaker 211). Block 846 is similar to block 812.

**[0255]** Referring back to block 842, in response to determining that the number of data items in the fourth list of data items does not exceed a predetermined number, one or more of blocks 848-850 are performed. Blocks 848-850 are directed to generating and providing a response that contains the entire fourth list of data items.

**[0256]** At block 848, a fifth spoken response is generated (e.g., by task flow processing module 736, dialogue processing module 734, and/or speech synthesis module 740). In particular, the fifth spoken response comprises the fourth list of data items. Block 848 is similar to block 844, described above. Notably, because the number of data items in the fourth list of data items does not exceed the predetermined number, the entire fourth list of data items is included in the fifth spoken response without overwhelming the user. The fifth spoken response includes a fifth spoken preamble and/or a fifth spoken prompt that is similar to the fourth spoken preamble and the fourth spoken prompt described above in block 844.

**[0257]** At block 850, the fifth spoken response is provided (e.g., using speech synthesis module 740, I/O processing module 728, and/or speaker 211). Block 850 is similar to block 846.

**[0258]** Blocks 840 through 850 of process 800 are further described with reference to the examples illustrated in FIGs. 9C-D. As shown in FIG. 9C, user 901 provides spoken user request 930 “Hey Siri, what does ‘plot’ mean?” to digital assistant 905 of user device 903. In this example, digital assistant 905 determines that spoken user request 930 is a specific request associated with the “dictionary” domain. Thus, the degree of specificity of spoken user request 930 is determined to be not less than a threshold level (block 804). In response to this determination, a list of six definitions for the word “plot” are retrieved (block 840) from a dictionary reference. Because user 901 is likely not familiar with the definitions and because the information density for each definition is relatively high, the predetermined number are determined (block 842) to be relatively low. For example, it are determined that at most two definitions (i.e., predetermined number = 2) should be presented to user 901 at a given time to avoid overwhelming user 901. Digital assistant 905 then determines (block 842) that the number of obtained definitions (e.g., six) exceeds the predetermined number (e.g., two) and thus only a

subset (e.g., two definitions) of the list of definitions is selected (block 844). In this example, the obtained list of definitions has an inherent order according to the dictionary reference. Thus, the first two definitions of the dictionary reference are selected as the subset of the list of definitions. As shown in FIG. 9C, spoken response 932 containing the two selected definitions is generated (block 844) and provided (block 846) to user 901. In this example, spoken response 932 includes the spoken preamble “Plot has multiple meanings...,” which provides context to user 901 that there is more than one definition of the word “plot” found. Additionally, spoken response 932 includes ordinal numbers (e.g., “First...,” “Second...”) that preface each of the two definitions. Further, spoken response 932 includes the spoken prompt “There are 4 more, if you’d like to hear them.” The spoken prompt informs user 901 that there are additional definitions available and the exact number of additional definitions. The spoken prompt also prompts user 901 to request for the additional definitions, if desired. It should be recognized that if the list of definitions only had one or two definitions (rather than six), then the entire list of one or two definitions would be included in spoken response 932 (blocks 848-850). In such an example, no spoken prompt indicating that additional definitions are available would be included.

**[0259]** In another example shown in FIG. 9D, user 901 provides spoken user request 934 “Hey Siri, any good movies playing today?” In this example, digital assistant 905 determines that there are a limited number of movies playing that given day near the user’s current location. Thus, the degree of specificity of spoken user request 930 is determined to be not less than a threshold level (block 804). In response to this determination, for example, a list of over twenty movies currently playing near the user’s location is retrieved (block 840) from a movie database. In this example, the movie titles in the list of movies are relatively short and thus the predetermined number is determined (block 842) to be a moderate number (e.g., five). For example, it is determined that up to five movie titles can be presented to user 901 at a given time without overwhelming user 901. Digital assistant 905 then determines (block 842) that the number of movies (e.g., greater than twenty) in the list of retrieved movies exceeds the predetermined number (e.g., five) and thus only a subset (e.g., five movies) of the list of movies is selected (block 844). In this example, the list of movies does not have an inherent order. However, spoken user request 934 defines the attribute “good” and thus the list of twenty movies is ranked (block 844) according to popularity and/or reviews and the five most popular movies in

the list are selected as the subset of the list of movies. As shown in FIG. 9D, spoken response 936 containing the selected subset of the list of movies is generated (block 844) and provided (block 846) to user 901. In this example, spoken response 936 includes the spoken preamble “I found a lot of movies playing today...,” which provides context to user 901 that a large number of movies are playing and only a portion of these movies is provided in spoken response 936. Further, spoken response 932 includes the spoken prompt “Let me know if you’d like to hear the rest,” which prompts user 901 to request for the additional movies in the list, if desired.

**[0260]** In some examples, process 800 allows for the user to provide follow-up spoken requests while or after the spoken response is provided. For example, while or after providing the spoken response (e.g., the first, second, third, fourth, or fifth spoken responses of blocks 812, 824, 834, 846, or 850, respectively), the user device receives audio input (e.g., by turning on microphone 213). The audio input is then analyzed to determine whether it contains a follow-up spoken request. This therefore enables the user to request additional data items not provided in the spoken response (e.g., in response to the spoken prompt). Alternatively, the user provides a new spoken user request. In some examples, if no follow-up spoken request is detected in the audio input, the user device ceases to receive audio input after a predetermined amount of time. Thus, the user is not forced into another interaction if no follow-up request is desired.

**[0261]** In one example, while providing a spoken response (e.g., the first, second, third, fourth, or fifth spoken responses of blocks 812, 824, 834, 846, or 850, respectively), a second user request is detected. For example, in FIG. 9D, while digital assistant 905 is providing spoken response 936, user 901 interrupts and provides a second user request. In response to detecting the second user request, digital assistant 905 ceases to provide the remaining portion of spoken response 936. For example, if the second user request were detected by digital assistant 905 while or after the movie title “The Revenant” is provided, but before the movie title “The Big Short” is provided, digital assistant 905 ceases to provide the remaining portion of spoken response 936 after the movie title “The Revenant.” Digital assistant 905 then generates a spoken response based on the second user request and a portion of the fourth spoken response that coincides with detecting the second user request. For example, while digital assistant 905 is providing the movie title “The Revenant,” user 901 interrupts with the second user request “What’s that about?” Digital assistant 905 determines that the detection of the second spoken

request coincided with the portion “The Revenant” of spoken response 936. In response, digital assistant stores information related to the movie title “The Revenant” as context to process the second user request. In particular, digital assistant 905 searches for plot information based on the movie title “The Revenant” and provides the plot information as a response to the second user request.

#### 5. Other Electronic Devices

**[0262]** FIG. 10 shows a functional block diagram of electronic device 1000 configured in accordance with the principles of the various described examples. The functional blocks of the device are optionally implemented by hardware, software, or a combination of hardware and software to carry out the principles of the various described examples. It is understood by persons of skill in the art that the functional blocks described in FIG. 10 are optionally combined or separated into sub-blocks to implement the principles of the various described examples. Therefore, the description herein optionally supports any possible combination, separation, or further definition of the functional blocks described herein.

**[0263]** As shown in FIG. 10, electronic device 1000 includes touch screen display unit 1002 configured to display a graphical user interface and to receive touch input from the user, audio input unit 1004 configured to receive audio input (e.g., speech input), speaker unit 1005 configured to output audio (e.g., speech), and communication unit 1006 configured to transmit and receive information. Electronic device 1000 further includes processing unit 1008 coupled to touch screen display unit 1002, audio input unit 1004, and communication unit 1006. In some examples, processing unit 1008 includes receiving unit 1010, determining unit 1012, obtaining unit 1014, generating unit 1016, providing unit 1018, selecting unit 1020, detecting unit 1022, and ceasing unit 1024.

**[0264]** In accordance with some embodiments, processing unit 1008 is configured to receive (e.g., with receiving unit 1010 and via audio input unit 1004) a spoken user request (e.g., spoken user request of block 802) associated with a plurality of data items. Processing unit 1008 is further configured to determine (e.g., with determining unit 1012) whether a degree of specificity of the spoken user request (e.g., degree of specificity of block 804) is less than a threshold level. Processing unit 1008 is further configured to, in response to determining that a degree of



specificity of the spoken user request is less than a threshold level, determine (e.g., with determining unit 1012) one or more attributes (e.g., one or more attributes of block 806) related to the spoken user request, the one or more attributes not defined in the spoken user request. Processing unit 1008 is further configured to obtain (e.g., with obtaining unit 1014) a list of data items (e.g., list of data items of block 808) based on the spoken user request and the one or more attributes. Processing unit 1008 is further configured to generate (e.g., with generating unit 1016) a spoken response (e.g., spoken response of block 810) comprising a subset of the list of data items. Processing unit 1008 is further configured to provide (e.g., with providing unit 1018 and using speaker unit 1005) the spoken response (e.g., spoken response provided at block 812).

**[0265]** In some examples, processing unit 1008 is further configured to determine (e.g., with determining unit 1012) a number of parameters (e.g., number of parameters of block 804) defined in the spoken user request. The degree of specificity is based on the number of parameters defined in the spoken user request.

**[0266]** In some examples, processing unit 1008 is further configured to determine (e.g., with determining unit 1012) a number of possible data items (e.g., number of possible data items of block 804) that satisfy the spoken user request. The degree of specificity is based on the number of possible data items.

**[0267]** In some examples, processing unit 1008 is further configured to determine (e.g., with determining unit 1012) a size of metadata (e.g., size of metadata of block 804) associated with data items that satisfy the spoken user request. The threshold level is based on the size of the metadata.

**[0268]** In some examples, processing unit 1008 is further configured to determine (e.g., with determining unit 1012) a domain (e.g., domain of block 804) corresponding to the spoken user request. The threshold level is based on the domain.

**[0269]** In some examples, processing unit 1008 is further configured to determine (e.g., with determining unit 1012) a degree of familiarity (e.g., degree of familiarity of block 804) associated with data items that satisfy the spoken user request. The threshold level is based on the degree of familiarity.

**[0270]** In some examples, generating a spoken response includes generating a spoken preamble (e.g., spoken preamble of block 810) that describes an attribute of the one or more attributes. Further, providing the spoken response includes providing the spoken preamble prior to providing the subset of the list of data items.

**[0271]** In some examples, the spoken response includes a description (e.g., description of block 810) that specifies an additional attribute for each data item of the subset of the list of data items. The additional attribute is not defined in the spoken user request and is different from any of the one or more attributes.

**[0272]** In some examples, the subset of the list of data items has at most a predetermined number of data items.

**[0273]** In some examples, processing unit 1008 is further configured to receive (e.g., with receiving unit 1010 and via audio input unit 1004) speech input (e.g., speech input of block 814). Processing unit 1008 is further configured to, in response to receiving the speech input, determine (e.g., with determining unit 1012) whether the speech input corresponds to a rejection (e.g., rejection of block 816) of the subset of the list of data items. Processing unit 1008 is further configured to, in response to determining that the speech input corresponds to a rejection of the subset of the list of data items, determine (e.g., with determining unit 1012) one or more second attributes (e.g., one or more second attributes of block 818) related to the spoken user request. The one or more second attributes are different from the one or more attributes and are not defined in the spoken user request. Processing unit 1008 is further configured to obtain (e.g., with obtaining unit 1014) a second list of data items (e.g., second list of data items of block 820) based on the spoken user request and the one or more second attributes. Processing unit 1008 is further configured to generate (e.g., with generating unit 1016) a second spoken response (e.g., second spoken response of block 822) comprising a subset of the second list of data items. Processing unit 1008 is further configured to provide (e.g., with providing unit 1018 and speaker unit 1005) the second spoken response (e.g., second spoken response provided at block 824).

**[0274]** In some examples, processing unit 1008 is further configured to, in response to determining that the speech input corresponds to a rejection of the subset of the list of data items, provide (e.g., with providing unit 1018 and speaker unit 1005) a spoken prompt (e.g., spoken

prompt of block 826) for the user to provide additional attributes to refine the spoken user request.

**[0275]** In some examples, processing unit 1008 is further configured to receive (e.g., with receiving unit and via audio input unit 1004) a second speech input (e.g., second speech input of block 828) responsive to the spoken prompt. Processing unit 1008 is further configured to obtain (e.g., with obtaining unit 1014) a third list of data items (e.g., third list of data items of block 830) based on the spoken user request and one or more attributes defined in the second speech input. Processing unit 1008 is further configured to generate (e.g., with generating unit 1016) a third spoken response (e.g., third spoken response of block 832) comprising a subset of the third list of data items. Processing unit 1008 is further configured to provide (e.g., with providing unit 1018 and speaker unit 1005) the third spoken response (e.g., third spoken response provided at block 834).

**[0276]** In some examples, processing unit 1008 is further configured to, in response to determining that the speech input does not correspond to a rejection of the subset of the list of data items, determine (e.g., with determining unit 1012) whether the speech input corresponds to an acceptance (e.g., acceptance of block 836) of a data item in the subset of the list of data items. Processing unit 1008 is further configured to, in response to determining that the speech input corresponds to an acceptance of a data item in the subset of the list of data items, provide (e.g., with providing unit 1018 and speaker unit 1005) content (e.g., content of block 838) associated with the accepted data item.

**[0277]** In some examples, processing unit 1008 is further configured to determine (e.g., with determining unit 1012) whether each of a predetermined number of previous spoken responses include a spoken prompt indicating that additional data items are available (e.g., blocks 810 and 812). Processing unit 1008 is further configured to, in response to determining that each of a predetermined number of previous spoken responses do not include a spoken prompt indicating that additional data items are available, provide (e.g., with providing unit 1018 and speaker unit 1005), in the spoken response, a spoken prompt indicating that additional data items are available (e.g., block 810 and 812). Processing unit 1008 is further configured to, in response to determining that each of a predetermined number of previous spoken responses include a spoken

prompt indicating that additional data items are available, forgo providing (e.g., with providing unit 1018), in the spoken response, a spoken prompt indicating that additional data items are available (e.g., block 810 and 812).

**[0278]** In some examples, processing unit 1008 is further configured to, in response to determining that a degree of specificity of the spoken user request is not less than a threshold level, obtain (e.g., with obtaining unit 1014) a fourth list of data items (e.g., fourth list of data items of block 840) based on the spoken user request. Processing unit 1008 is further configured to determine (e.g., with determining unit 1012) whether a number of data items in the fourth list of data items exceeds a predetermined number (e.g., block 842). Processing unit 1008 is further configured to, in response to determining that a number of data items in the fourth list of data items exceeds a predetermined number, generate (e.g., with generating unit 1016) a fourth spoken response (e.g., fourth spoken response of block 844) comprising a subset of the fourth list of data items. Processing unit 1008 is further configured to provide (e.g., with providing unit 1018 and speaker unit 1005) the fourth spoken response (e.g., fourth spoken response provided at block 846).

**[0279]** In some examples, generating the fourth spoken response includes generating a fourth spoken preamble (e.g., fourth spoken preamble of block 844) that indicates a number of data items in the fourth list of data items.

**[0280]** In some examples, the subset of the fourth list of data items is contrary to an expected result inferred from the spoken user request, and wherein the fourth spoken preamble indicates that the subset of the fourth list of data items is contrary to an expected result inferred from the spoken user request.

**[0281]** In some examples, a number of data items in the subset of the fourth list of data items is less than or equal to the predetermined number.

**[0282]** In some examples, processing unit 1008 is further configured to, in response to determining that a number of data items in the fourth list of data items does not exceed a predetermined number, generate (e.g., with generating unit 1016) a fifth spoken response (e.g., fifth spoken response of block 848) comprising the fourth list of data items. Processing unit

1008 is further configured to provide (e.g., with providing unit 1018 and speaker unit 1005) the fifth spoken response (e.g., fifth spoken response provided at block 850).

**[0283]** In some examples, processing unit 1008 is further configured to select (e.g., with selecting unit 1020), based on an attribute defined in the spoken user request, the subset of the fourth list of data items from the fourth list of data items (e.g., block 844).

**[0284]** In some examples, the fourth list of data items has a predetermined order, wherein the subset of the fourth list of data items data items includes a first N data items in the fourth list of data items, and wherein N is less than or equal to the predetermined number (e.g., block 844).

**[0285]** In some examples, the spoken user request is associated with a user, and wherein the predetermined number is based on a degree of familiarity of the user with data items in the fourth list of data items.

**[0286]** In some examples, the predetermined number is based on an amount of metadata associated with each data item in the fourth list of data items.

**[0287]** In some examples, the predetermined number is based on a cognitive load associated with the user at the time the spoken user request is received.

**[0288]** In some examples, the fourth spoken response includes a second spoken prompt indicating that additional data items are available.

**[0289]** In some examples, the second spoken prompt indicates a number of remaining data items in the fourth list of data items.

**[0290]** In some examples, processing unit 1008 is further configured to, while providing the fourth spoken response, detect (e.g., with detecting unit 1022) a second user request (e.g., second user request in speech input of block 814). Processing unit 1008 is further configured to, in response to detecting the second user request, cease (e.g., with ceasing unit 1024) to provide the fourth spoken response. Processing unit 1008 is further configured to generate (e.g., with generating unit 1016) a sixth spoken response based on the second user request and a portion of the fourth spoken response that coincides with detecting the second user request. Processing unit

1008 is further configured to provide (e.g., with providing unit 1018 and speaker unit 1005) the sixth spoken response.

**[0291]** In some examples, the one or more attributes are determined based on a frequency that an attribute is specified in a plurality of user requests from a plurality of users.

**[0292]** In some examples, the one or more attributes are determined based a user profile associated with the spoken user request.

**[0293]** In some examples, determining the one or more attributes includes randomly selecting an attribute from a category of attributes.

**[0294]** The operations described above with reference to FIG. 8A-D are optionally implemented by components depicted in FIGS. 1-4, 6A-B, and 7A-C. For example, the operations of process 800 may be implemented by one or more of operating system 718, applications module 724, I/O processing module 728, STT processing module 730, natural language processing module 732, dialogue flow processing module 734, task flow processing module 736, service processing module 738, or processor(s) 220, 410, 704. It would be clear to a person having ordinary skill in the art how other processes are implemented based on the components depicted in FIGS. 1-4, 6A-B, and 7A-C.

**[0295]** In accordance with some implementations, a computer-readable storage medium (e.g., a non-transitory computer readable storage medium) is provided, the computer-readable storage medium storing one or more programs for execution by one or more processors of an electronic device, the one or more programs including instructions for performing any of the methods or processes described herein.

**[0296]** In accordance with some implementations, an electronic device (e.g., a portable electronic device) is provided that comprises means for performing any of the methods or processes described herein.

**[0297]** In accordance with some implementations, an electronic device (e.g., a portable electronic device) is provided that comprises a processing unit configured to perform any of the methods or processes described herein.

**[0298]** In accordance with some implementations, an electronic device (e.g., a portable electronic device) is provided that comprises one or more processors and memory storing one or more programs for execution by the one or more processors, the one or more programs including instructions for performing any of the methods or processes described herein.

**[0299]** The foregoing description, for purpose of explanation, has been described with reference to specific embodiments. However, the illustrative discussions above are not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the techniques and their practical applications. Others skilled in the art are thereby enabled to best utilize the techniques and various embodiments with various modifications as are suited to the particular use contemplated.

**[0300]** Although the disclosure and examples have been fully described with reference to the accompanying drawings, it is to be noted that various changes and modifications will become apparent to those skilled in the art. Such changes and modifications are to be understood as being included within the scope of the disclosure and examples as defined by the claims.

**[0301]** As described above, one aspect of the present technology is the gathering and use of data available from various sources to improve the delivery to users of invitational content or any other content that may be of interest to them. The present disclosure contemplates that in some instances, this gathered data may include personal information data that uniquely identifies or can be used to contact or locate a specific person. Such personal information data can include demographic data, location-based data, telephone numbers, email addresses, home addresses, or any other identifying information.

**[0302]** The present disclosure recognizes that the use of such personal information data, in the present technology, can be used to the benefit of users. For example, the personal information data can be used to deliver targeted content that is of greater interest to the user. Accordingly, use of such personal information data enables calculated control of the delivered content. Further, other uses for personal information data that benefit the user are also contemplated by the present disclosure.

**[0303]** The present disclosure further contemplates that the entities responsible for the collection, analysis, disclosure, transfer, storage, or other use of such personal information data will comply with well-established privacy policies and/or privacy practices. In particular, such entities should implement and consistently use privacy policies and practices that are generally recognized as meeting or exceeding industry or governmental requirements for maintaining personal information data private and secure. For example, personal information from users should be collected for legitimate and reasonable uses of the entity and not shared or sold outside of those legitimate uses. Further, such collection should occur only after receiving the informed consent of the users. Additionally, such entities would take any needed steps for safeguarding and securing access to such personal information data and ensuring that others with access to the personal information data adhere to their privacy policies and procedures. Further, such entities can subject themselves to evaluation by third parties to certify their adherence to widely accepted privacy policies and practices.

**[0304]** Despite the foregoing, the present disclosure also contemplates embodiments in which users selectively block the use of, or access to, personal information data. That is, the present disclosure contemplates that hardware and/or software elements can be provided to prevent or block access to such personal information data. For example, in the case of advertisement delivery services, the present technology can be configured to allow users to select to “opt in” or “opt out” of participation in the collection of personal information data during registration for services. In another example, users can select not to provide location information for targeted content delivery services. In yet another example, users can select to not provide precise location information, but permit the transfer of location zone information.

**[0305]** Therefore, although the present disclosure broadly covers use of personal information data to implement one or more various disclosed embodiments, the present disclosure also contemplates that the various embodiments can also be implemented without the need for accessing such personal information data. That is, the various embodiments of the present technology are not rendered inoperable due to the lack of all or a portion of such personal information data. For example, content can be selected and delivered to users by inferring preferences based on non-personal information data or a bare minimum amount of personal information, such as the content being requested by the device associated with a user, other non-



1002069204

personal information available to the content delivery services, or publically available information.

2018100133 31 Jan 2018

## WHAT IS CLAIMED IS:

1. A method for operating a digital assistant, the method comprising:  
at a portable electronic device with a processor and memory:
  - receiving, from an audio input unit, a spoken user request associated with a plurality of data items;
  - determining whether a degree of specificity of the spoken user request is less than a threshold level;
  - in response to determining that a degree of specificity of the spoken user request is less than a threshold level:
    - determining one or more attributes related to the spoken user request, the one or more attributes not defined in the spoken user request;
    - obtaining a list of data items based on the spoken user request and the one or more attributes;
    - generating a spoken response comprising a subset of the list of data items;
  - and
  - providing, to a speaker unit, the spoken response.
2. The method of claim 1, further comprising:
  - determining a number of parameters defined in the spoken user request, wherein the degree of specificity is based on the number of parameters defined in the spoken user request.
3. The method of any of claims 1-2, wherein generating a spoken response includes generating a spoken preamble that describes an attribute of the one or more attributes, and wherein providing the spoken response includes providing the spoken preamble prior to providing the subset of the list of data items.
4. The method of any of claims 1-3, wherein the spoken response includes a description that specifies an additional attribute for each data item of the subset of the list of data items, and wherein the additional attribute is not defined in the spoken user request and is different from any of the one or more attributes.

1002491134

5. A computer-readable medium having instructions stored thereon, the instructions, when executed by one or more processors, cause the one or more processors to perform any of the methods of claims 1-4.

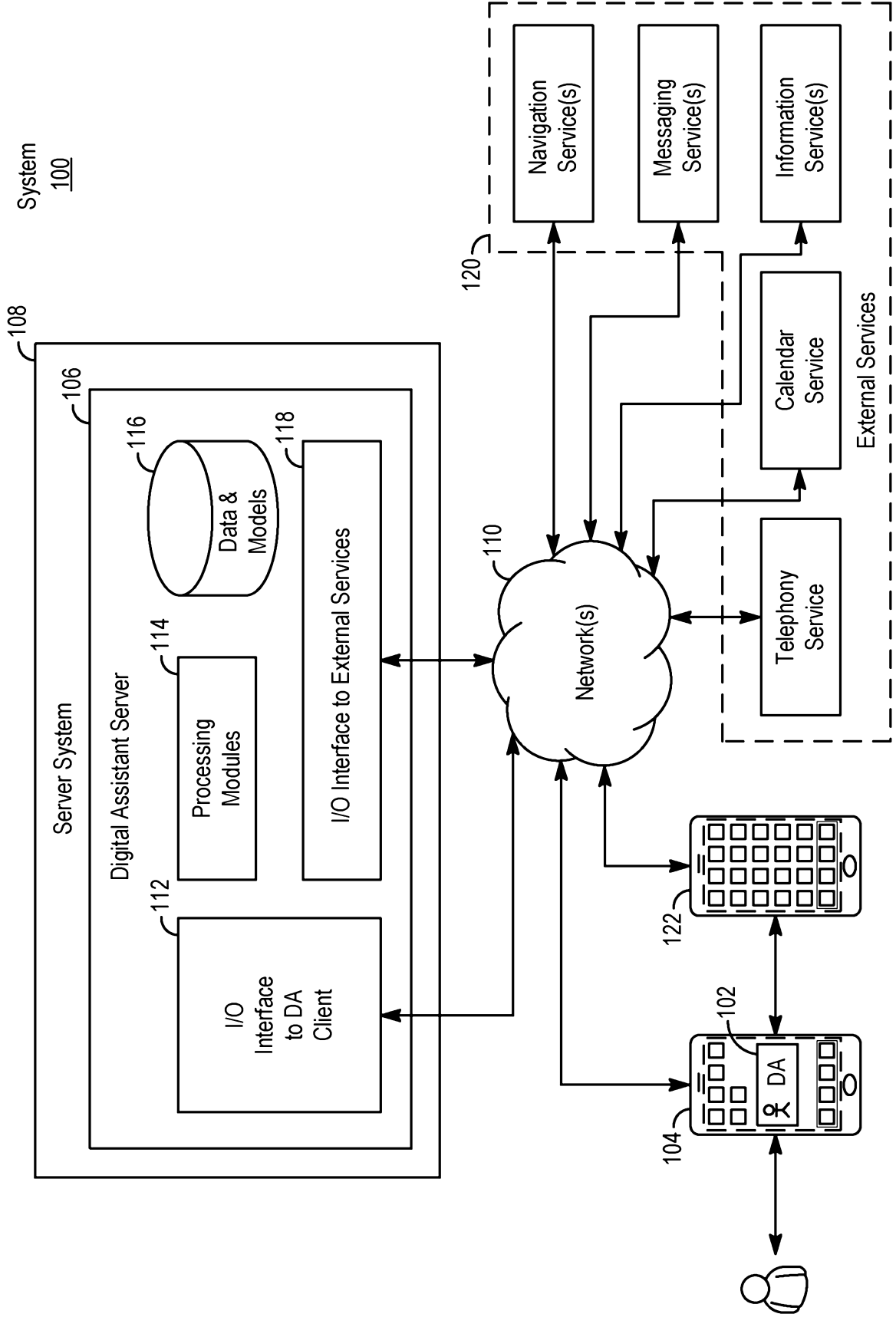


FIG. 1

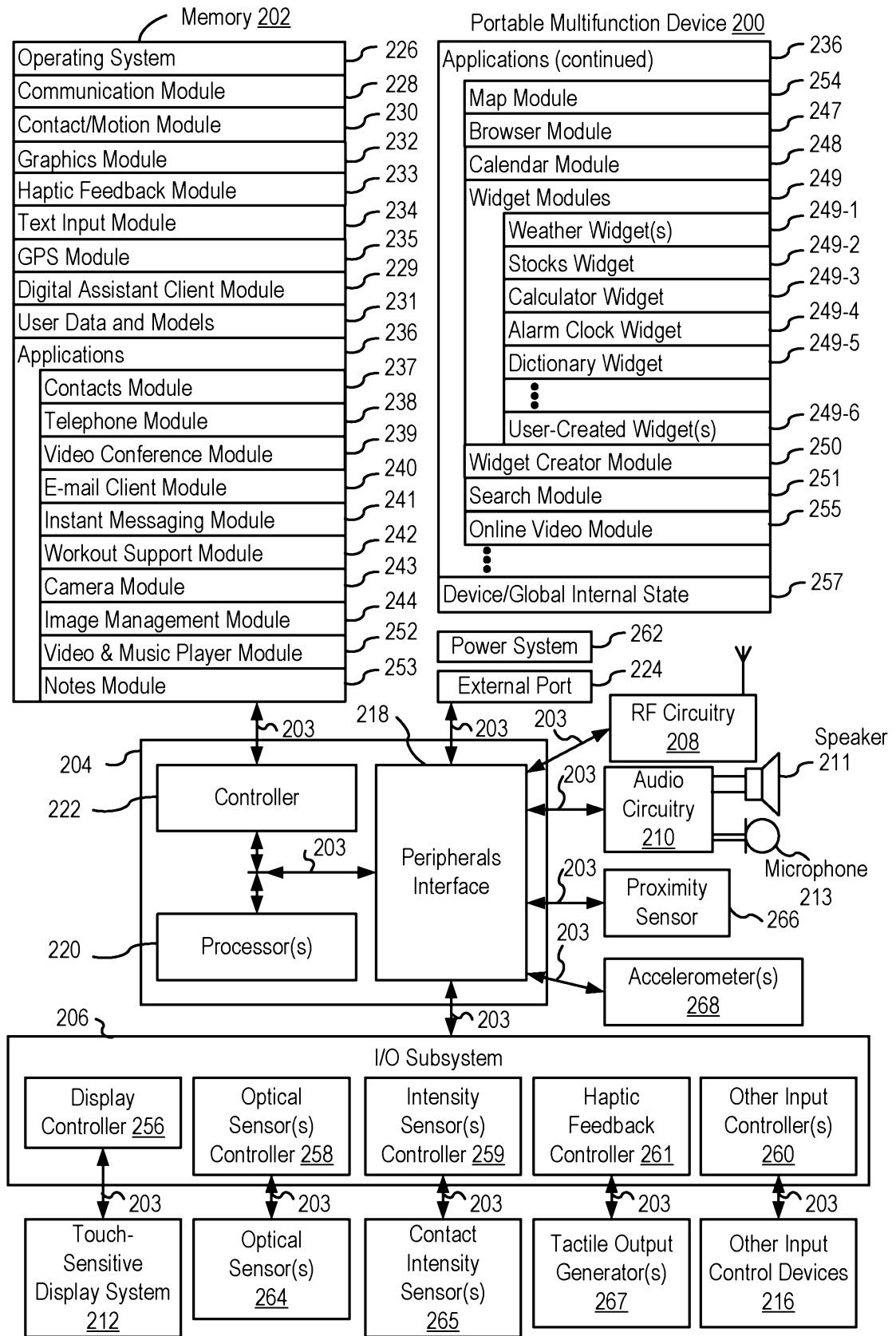


FIG. 2A

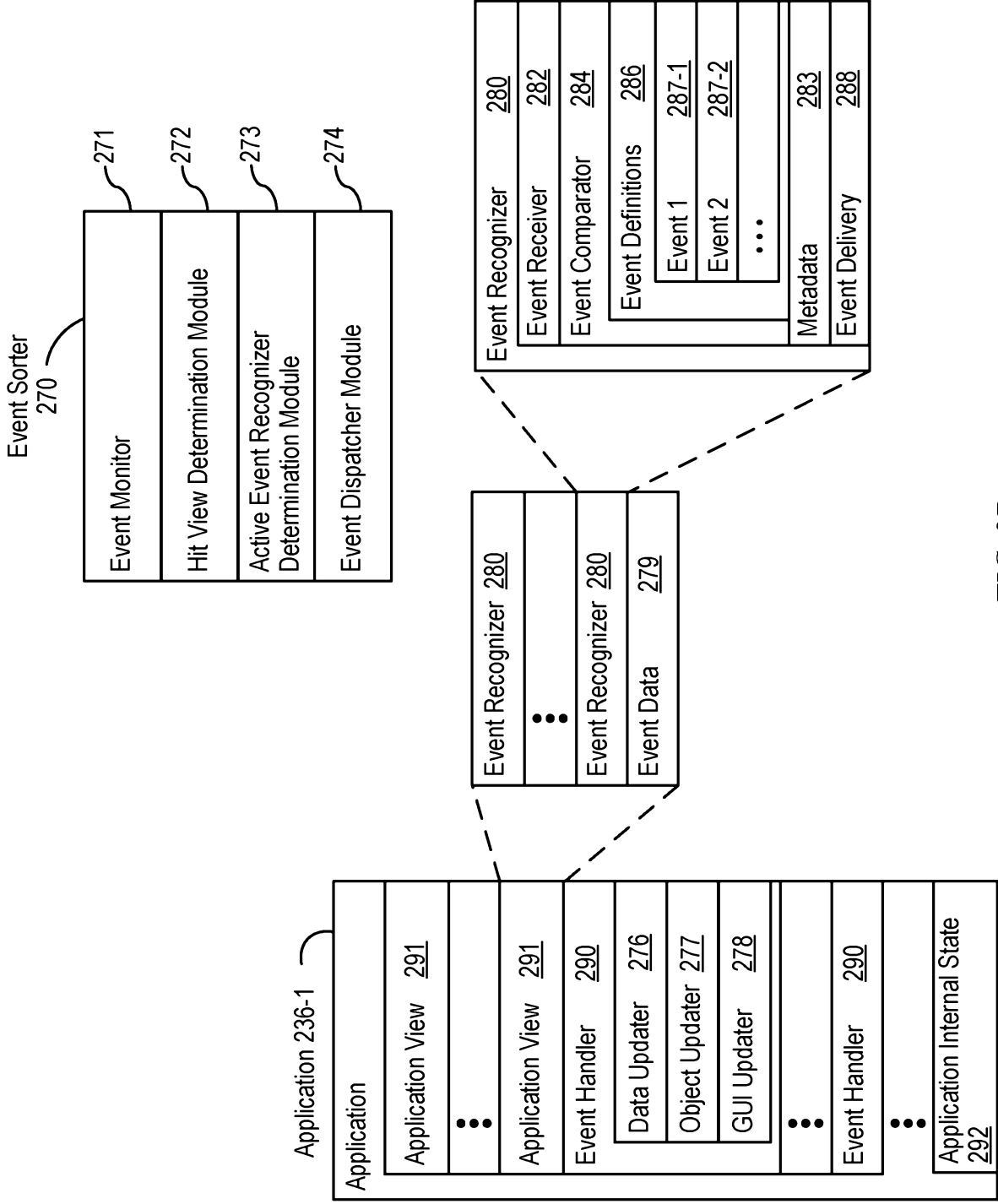


FIG. 2B

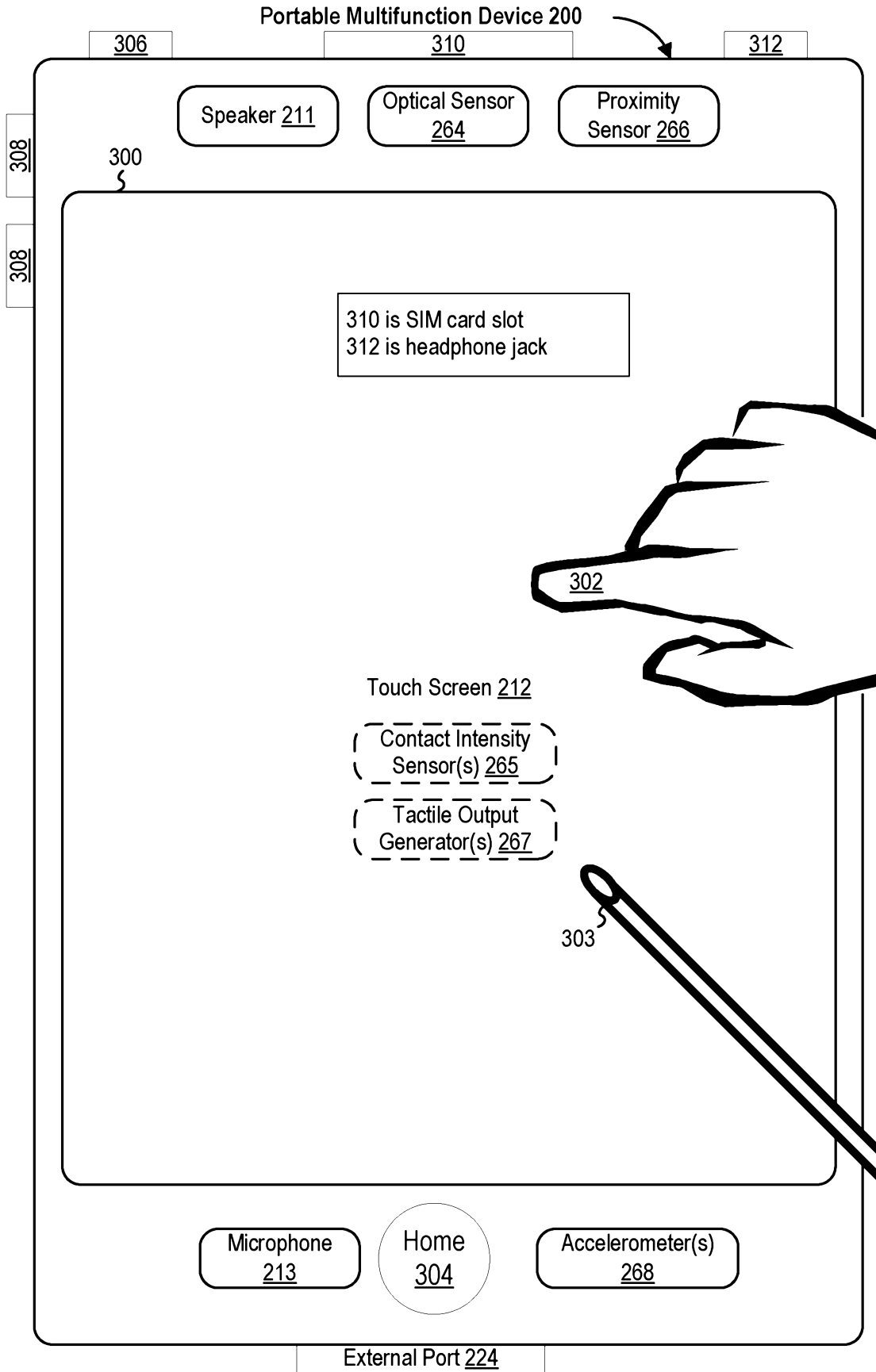


FIG. 3

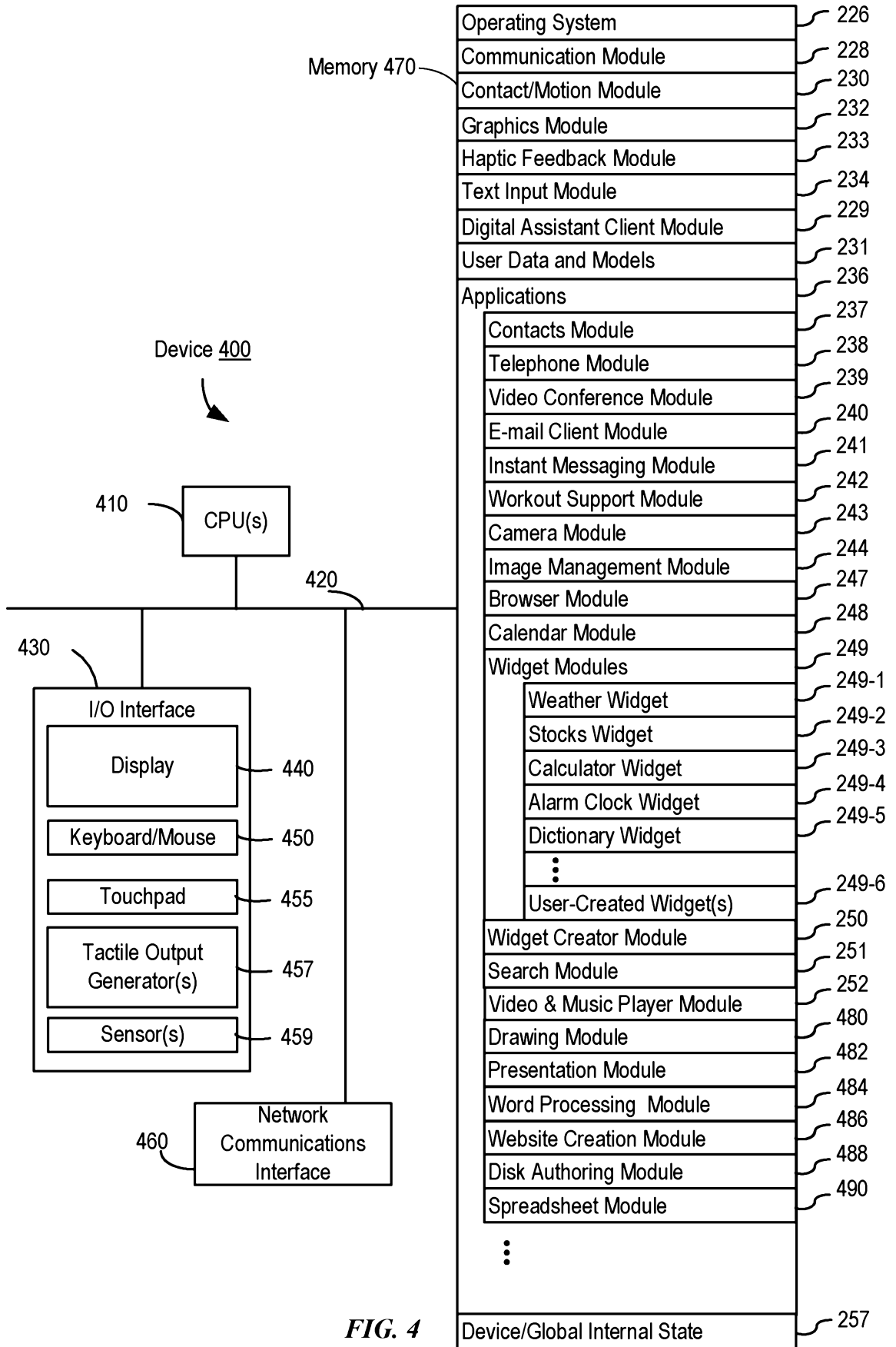


FIG. 4



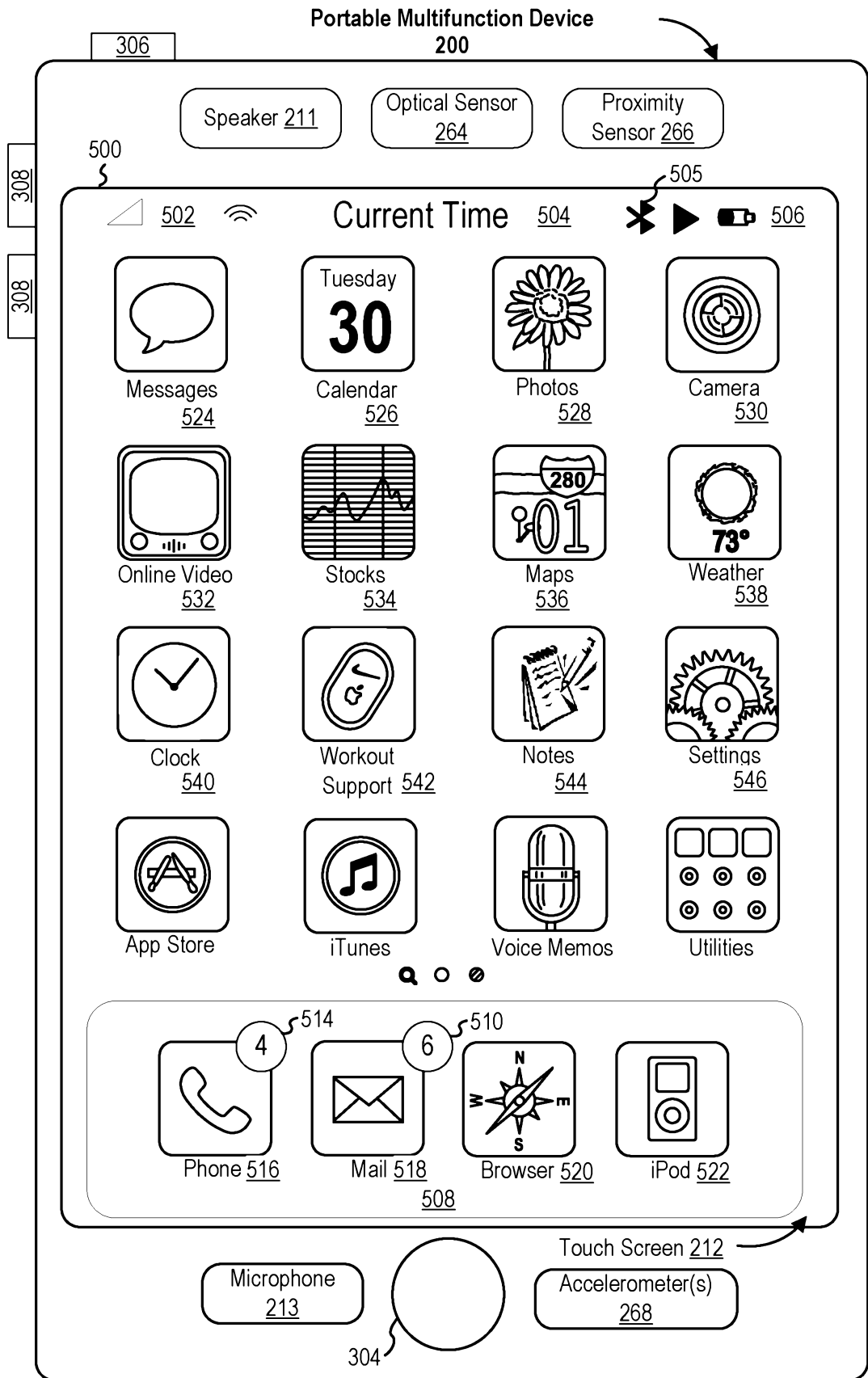


FIG. 5A

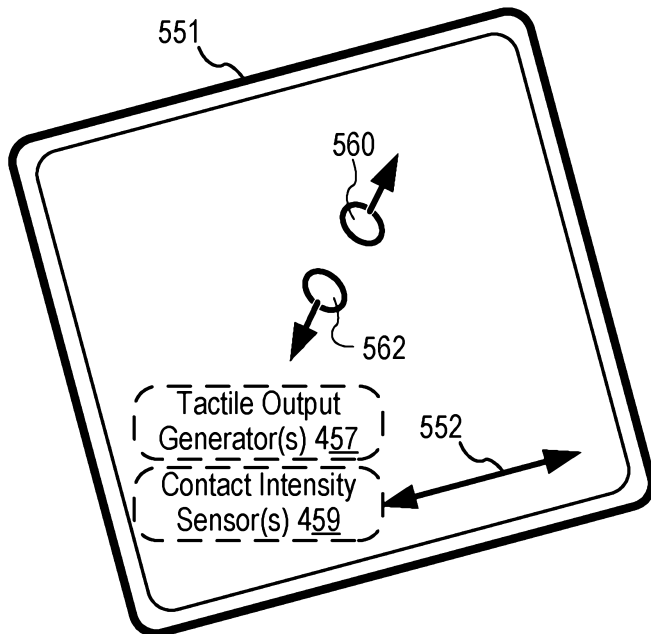
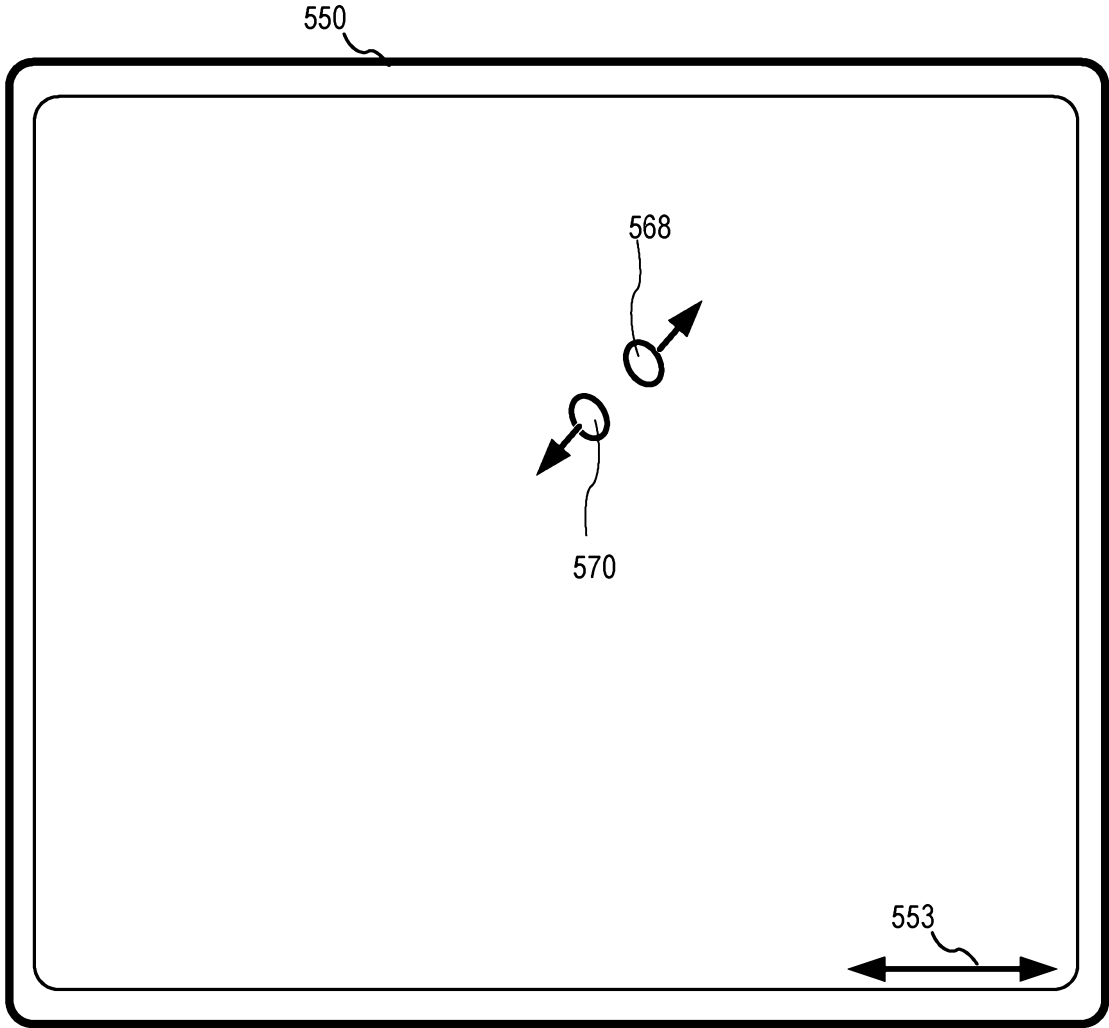
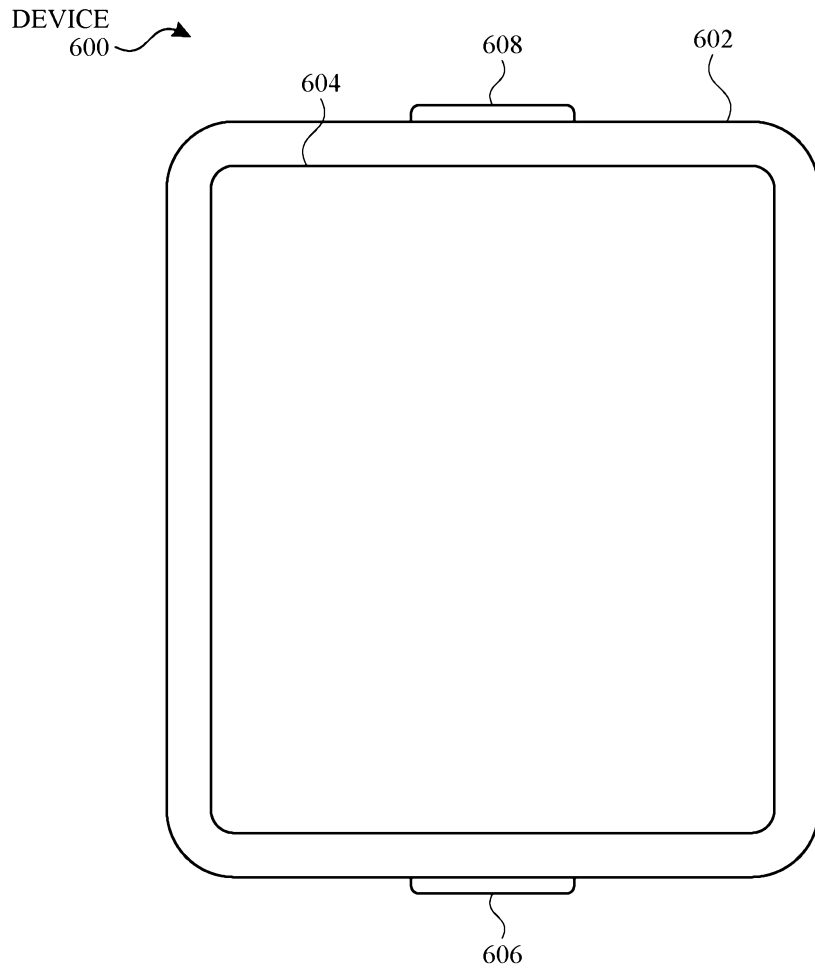


FIG. 5B



**FIG. 6A**

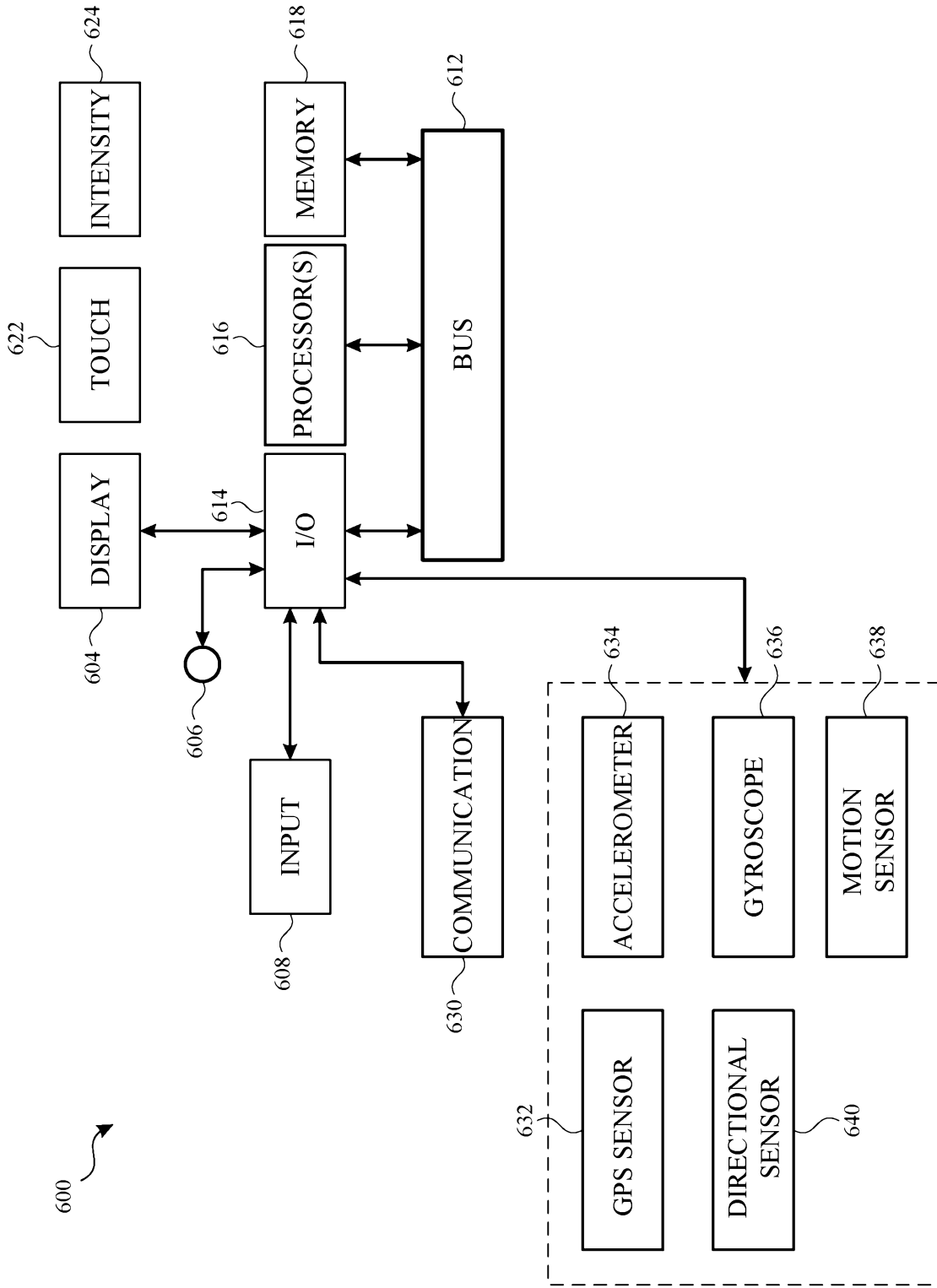


FIG. 6B

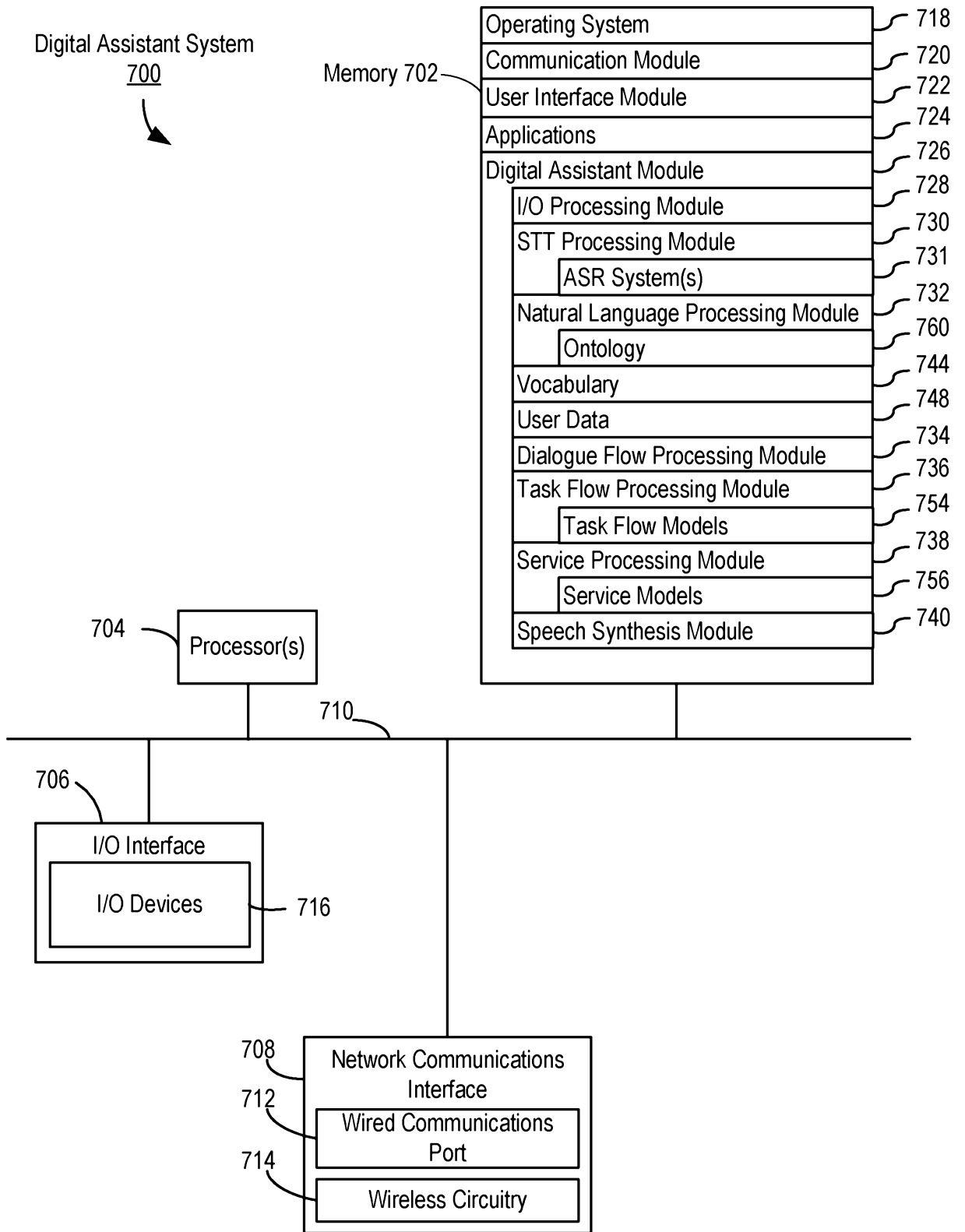


FIG. 7A

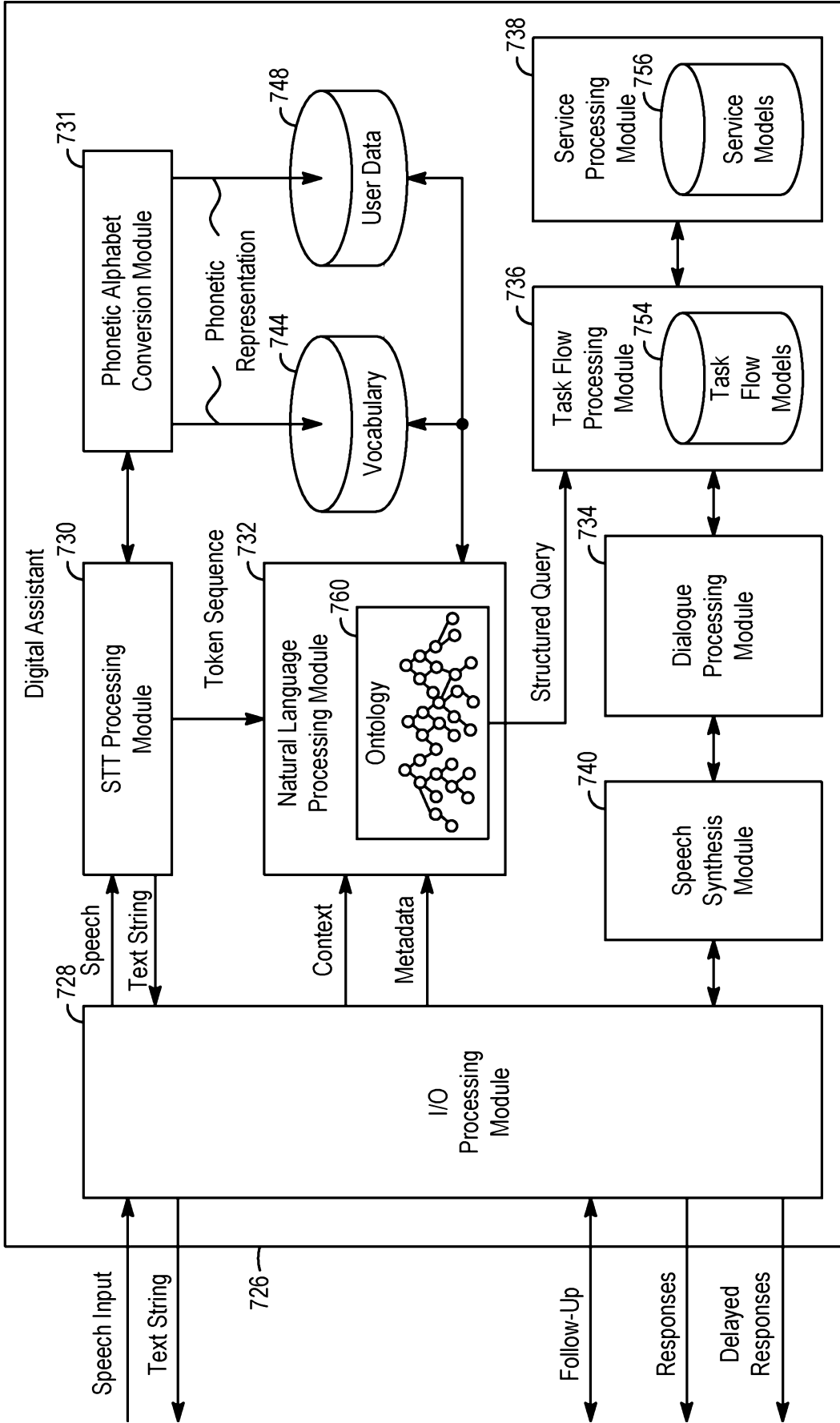


FIG. 7B

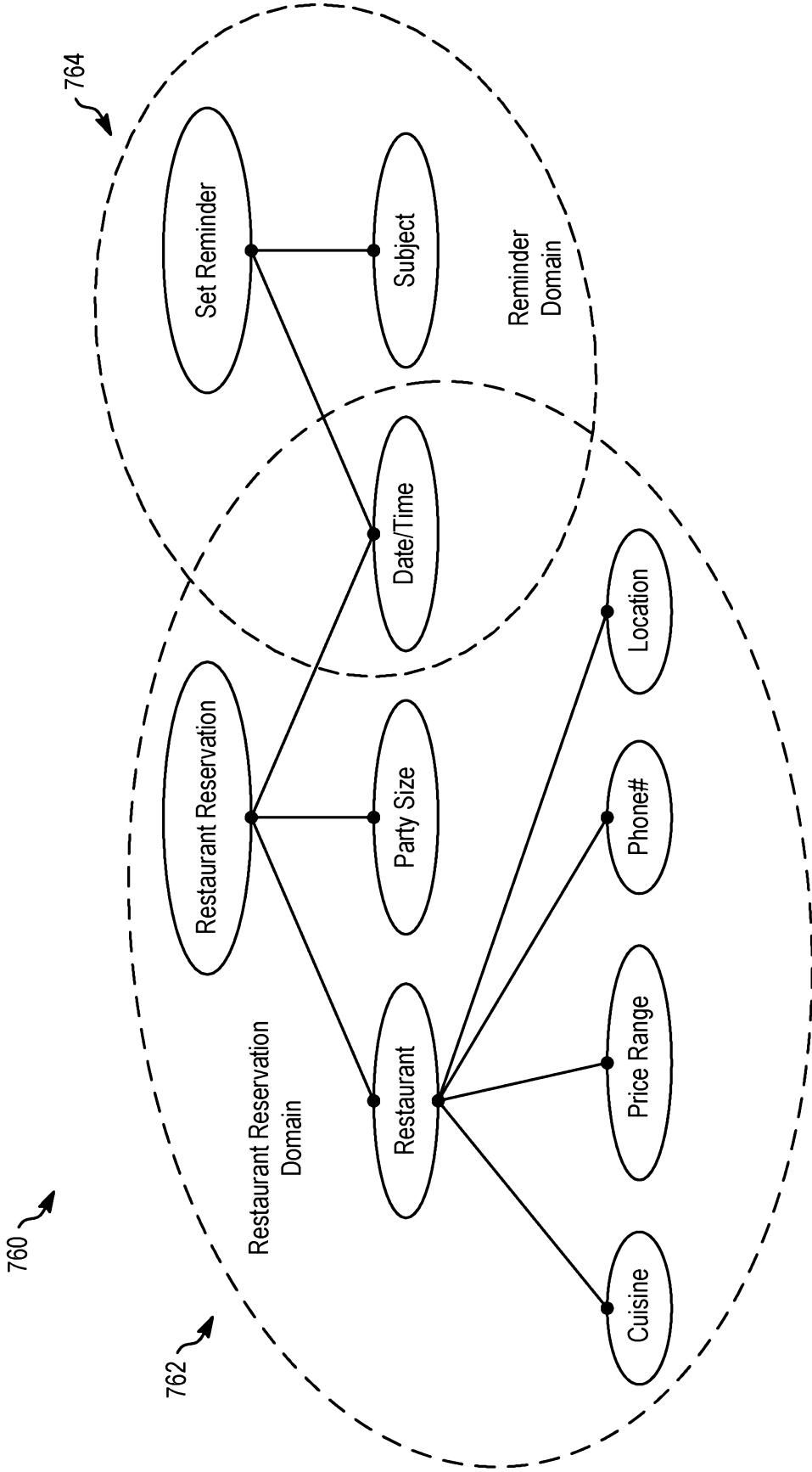
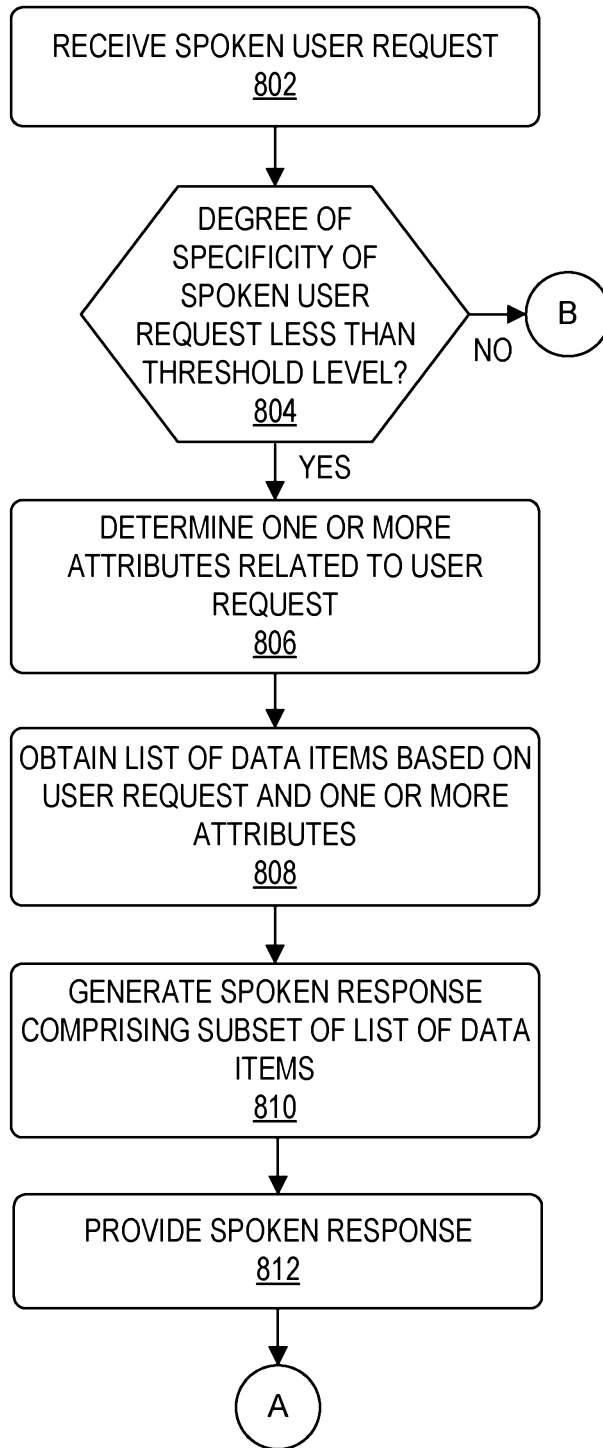


FIG. 7C

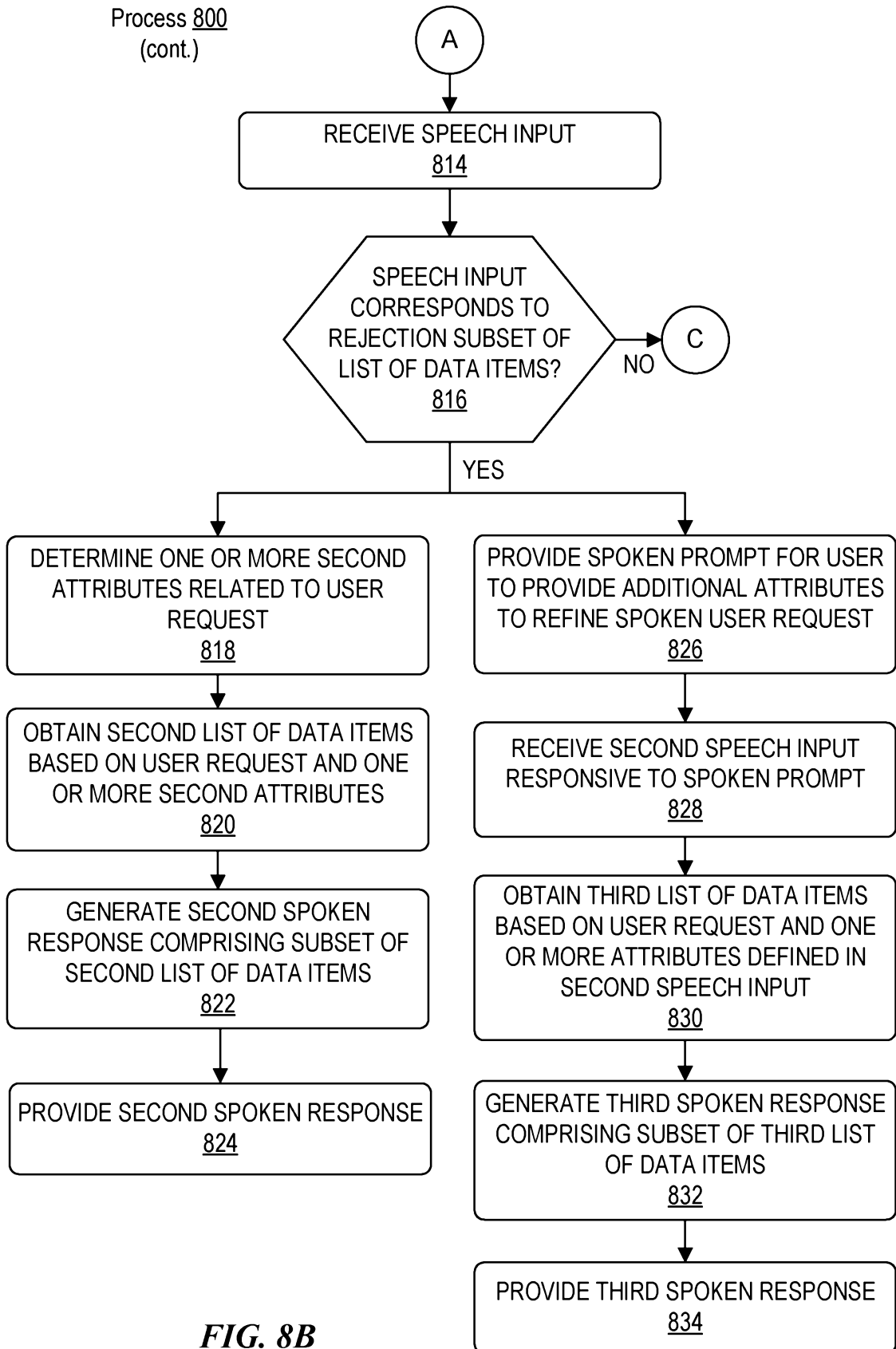
Process 800



**FIG. 8A**

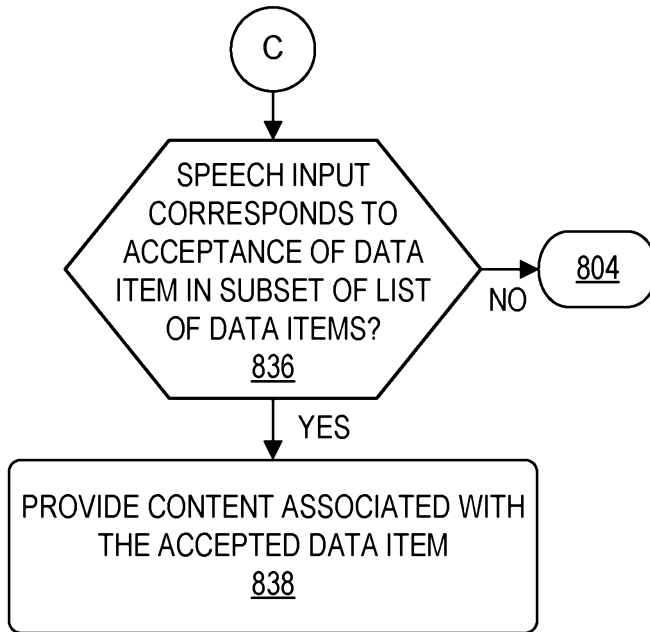


Process 800  
(cont.)



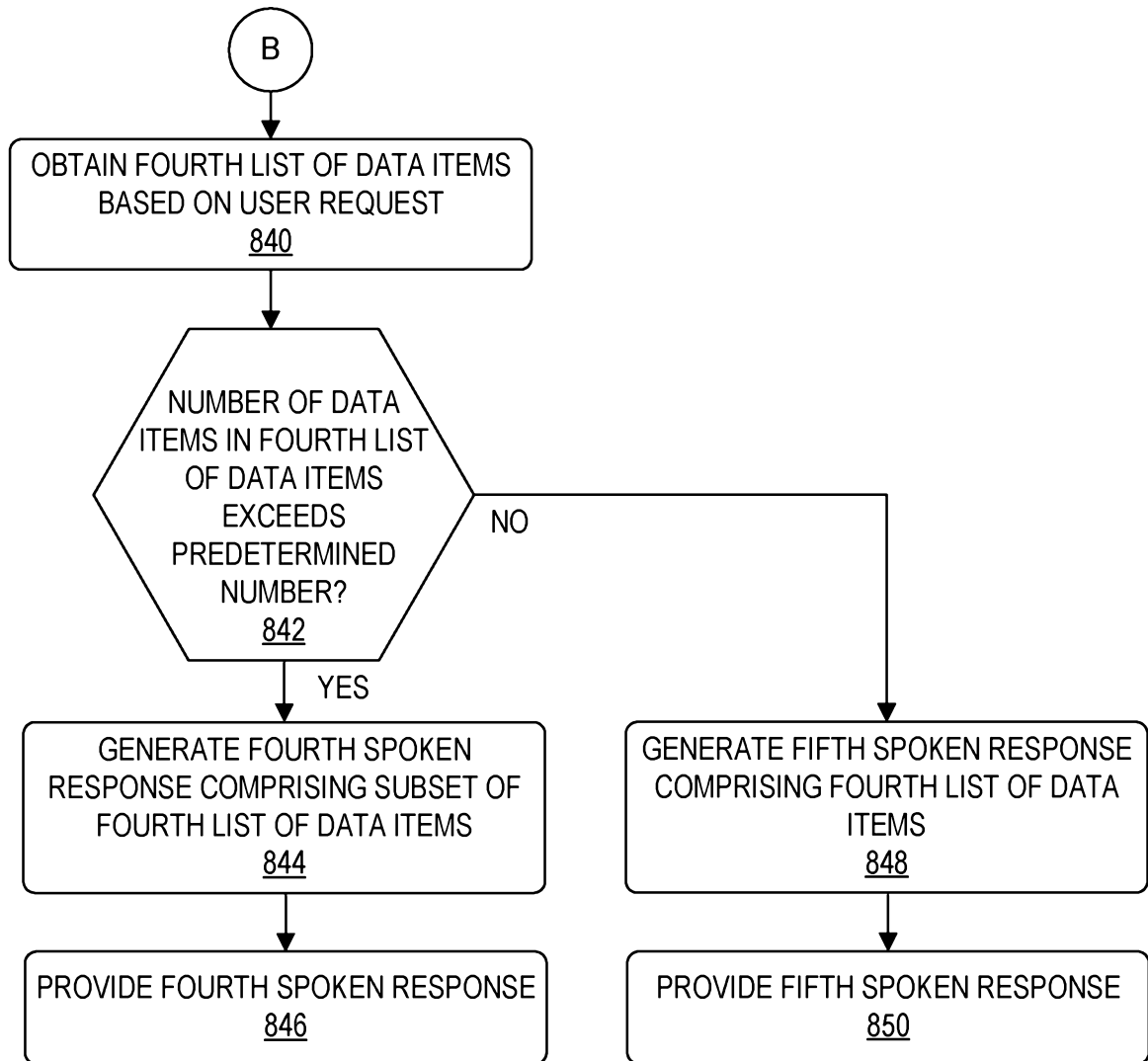
**FIG. 8B**

Process 800  
(cont.)



**FIG. 8C**

Process 800  
(cont.)



**FIG. 8D**

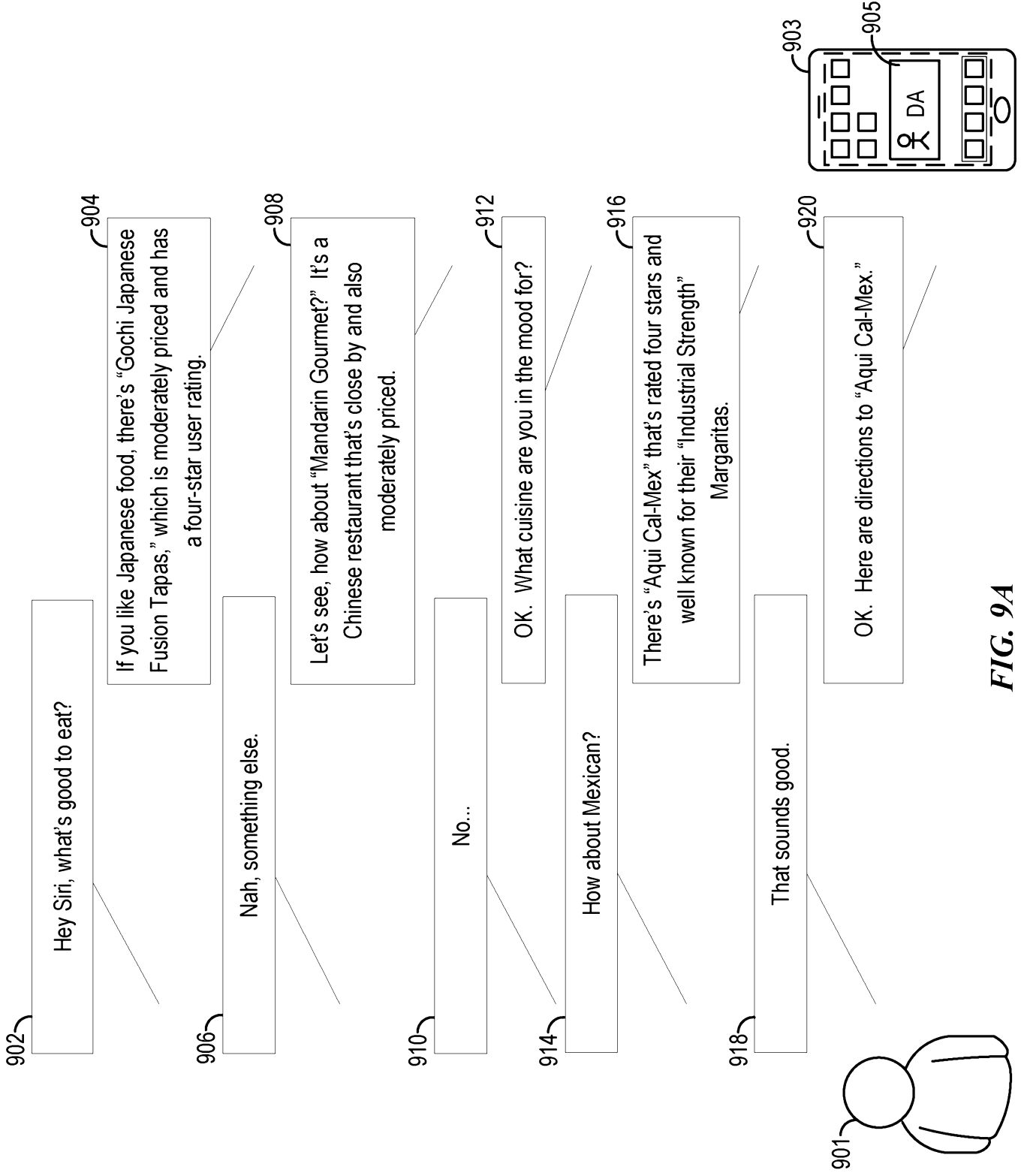
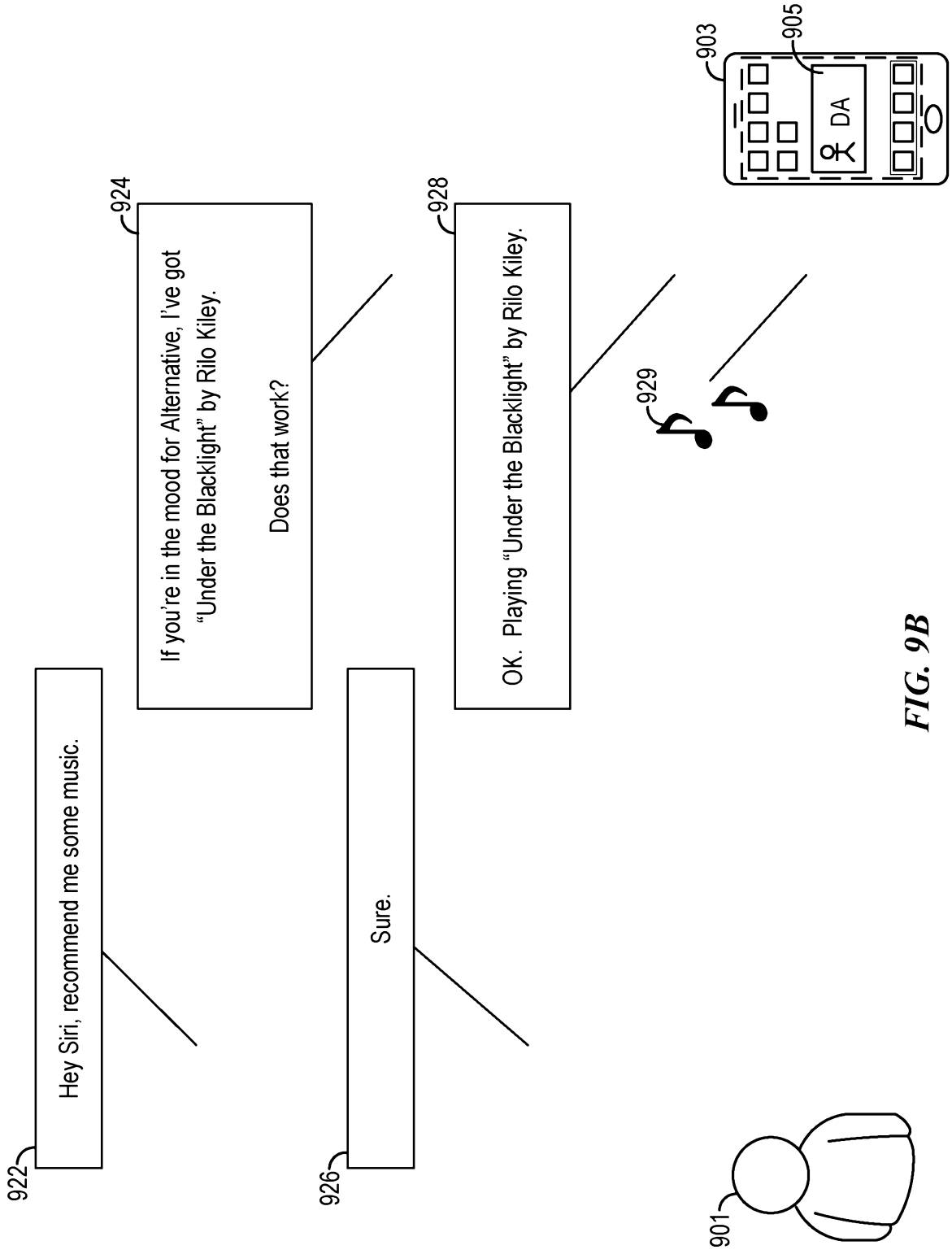


FIG. 9A



**FIG. 9B**

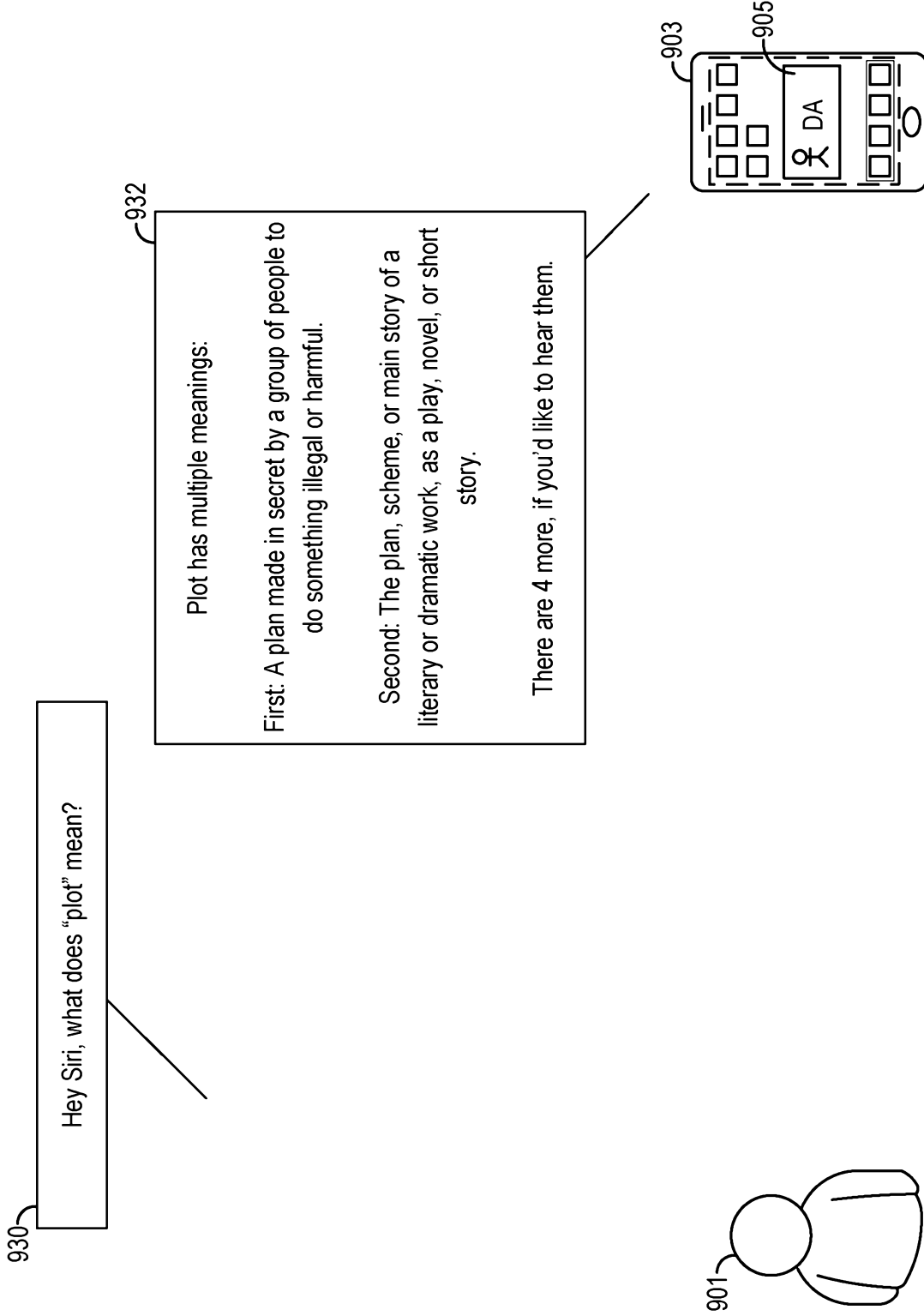


FIG. 9C

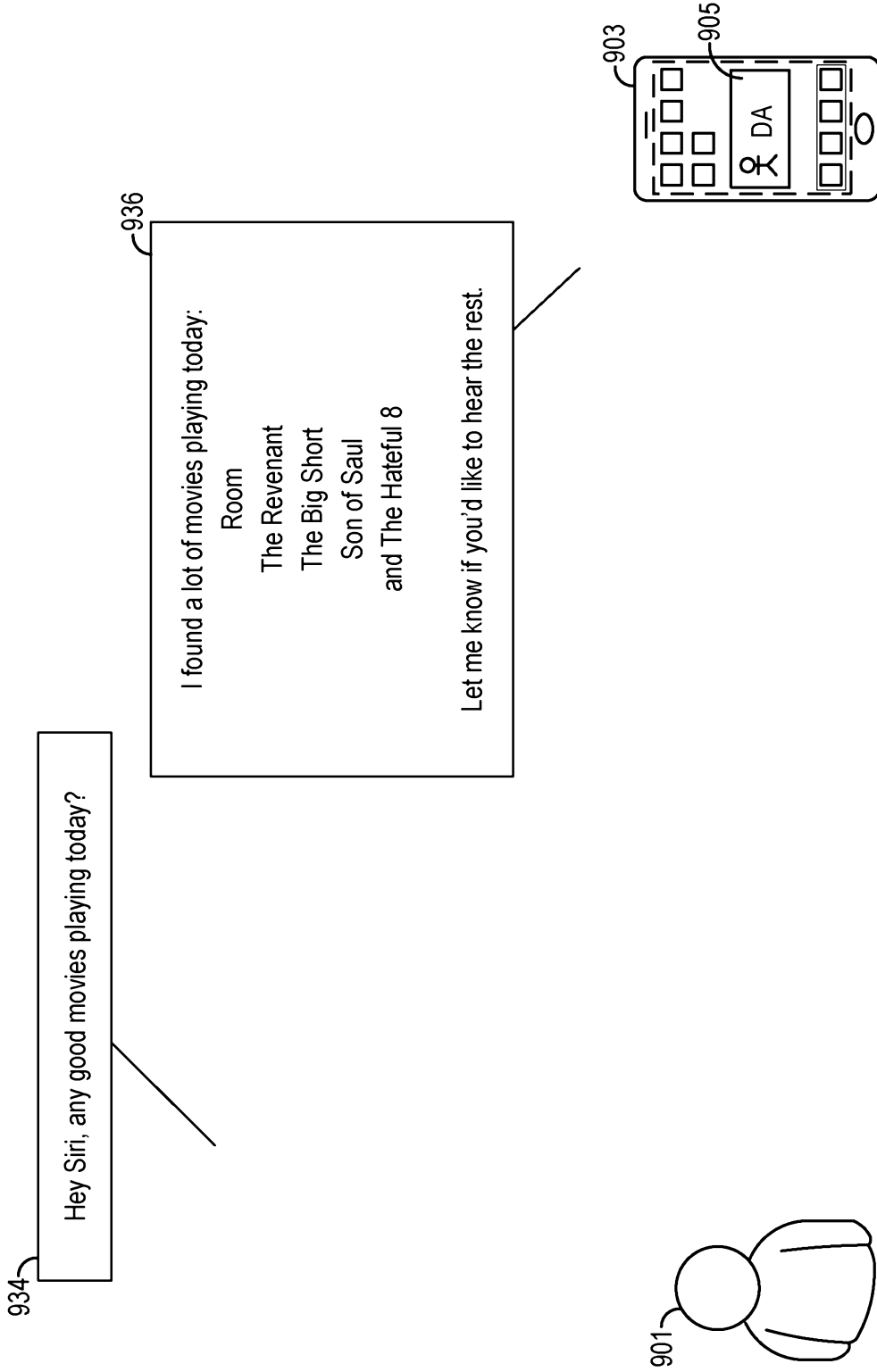


FIG. 9D

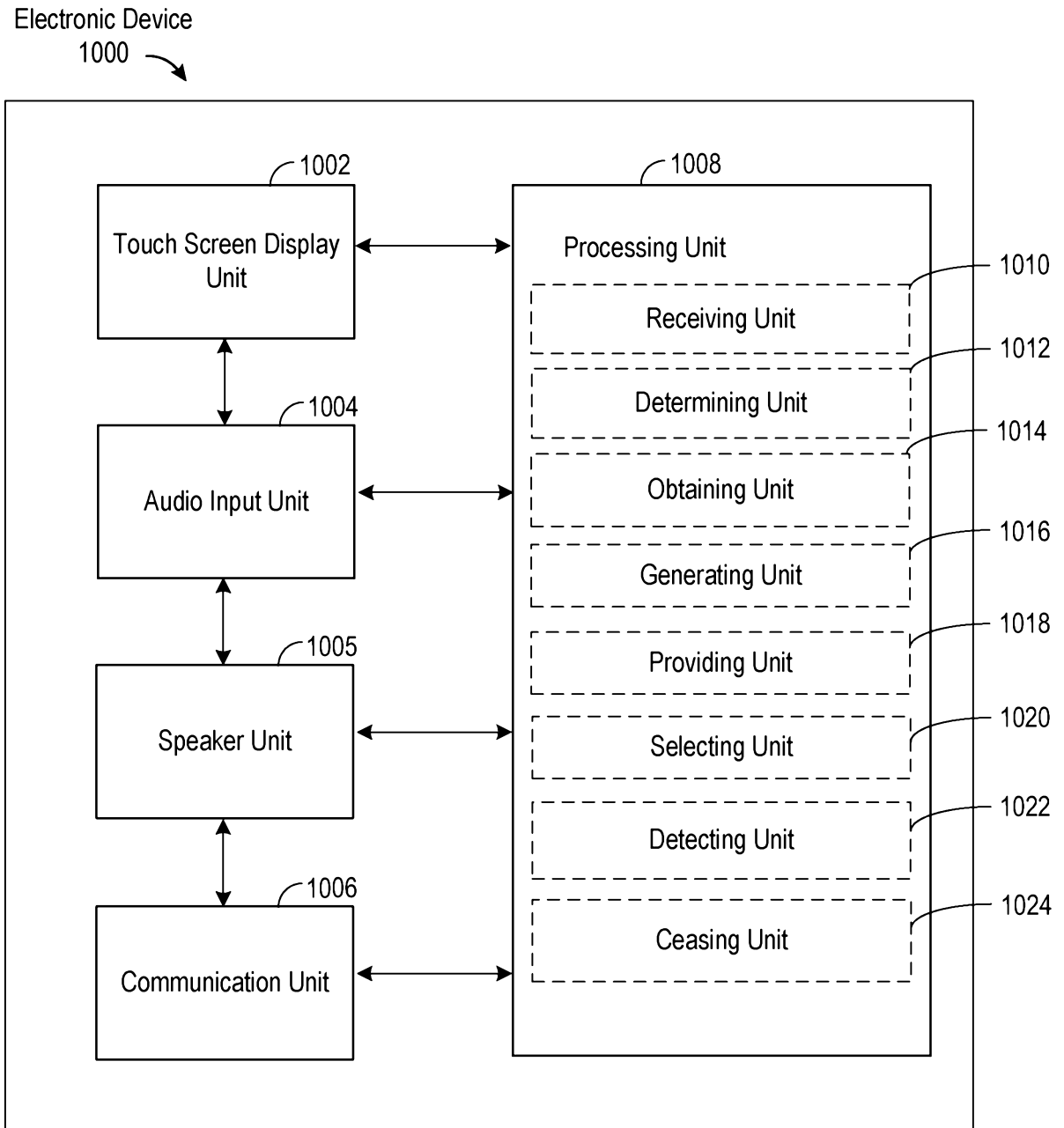


FIG. 10