(12) STANDARD PATENT (11) Application No. **AU 2015266863 B2**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Multi-command single utterance input method

(51) International Patent Classification(s)
G10L 15/18 (2006.01) *G10L 15/08* (2006.01)
G10L 15/26 (2006.01) *G10L 15/22* (2006.01)
G10L 15/28 (2006.01)

(21) Application No: **2015266863** (22) Date of Filing: **2015.05.28**

(87) WIPO No: **WO15/184186**

(30) Priority Data

(31) Number	(32) Date	(33) Country
62/005,556	**2014.05.30**	**US**
62/129,851	**2015.03.08**	**US**

(43) Publication Date: **2015.12.03**
(44) Accepted Journal Date: **2018.03.15**

(71) Applicant(s)
Apple Inc.

(72) Inventor(s)
Gruber, Thomas R.;Saddler, Harry J.;Bellegarda, Jerome R.;Nyeggen, Bryce H.;Sabatelli, Alessandro

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

(56) Related Art
YU ZHONG ET AL, "JustSpeak", W4A '14 Proceedings of the 11th Web for All Conference Article No. 36 , Seoul April 7 2014.
GB 2343285 A

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 December 2015 (03.12.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/184186 A1

(51) International Patent Classification:
G10L 15/18 (2013.01) *G10L 15/08* (2006.01) *G10L 15/26* (2006.01) *G10L 15/22* (2006.01) *G10L 15/28* (2013.01)

(21) International Application Number: PCT/US2015/033051

(22) International Filing Date: 28 May 2015 (28.05.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/005,556 30 May 2014 (30.05.2014) US
62/129,851 8 March 2015 (08.03.2015) US

(71) Applicant: **APPLE INC.** [US/US]; 1 Infinite Loop, Cupertino, CA 95014 (US).

(72) Inventors: **GRUBER, Thomas, R.**; C/o Apple Inc., 1 Infinite Loop, Cupertino, CA 95014 (US). **SADDLER, Harry, J.**; C/o Apple Inc., 1 Infinite Loop, Cupertino, CA 95014 (US). **BELLEGARDA, Jerome, R.**; 14118 Shadow Oak Way, Saratoga, CA 95070 (US). **NYEGGEN, Bryce, H.**; C/o Apple Inc., 1 Infinite Loop, Cupertino, CA 95014 (US). **SABATELLI, Alessandro**; 463 Ellsworth St., San Francisco, CA 94110 (US).

(74) Agents: **ARJOMAND, Mehran** et al.; Morrison & Foerster LLP, 755 Page Mill Road, Palo Alto, CA 94304-1018 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*

(54) Title: MULTI-COMMAND SINGLE UTTERANCE INPUT METHOD

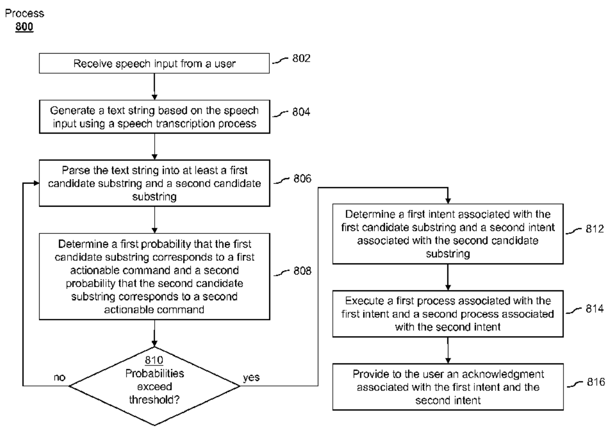

Process
800
Receive speech input from a user
802
Generate a text string based on the speech input using a speech transcription process
804
Parse the text string into at least a first candidate substring and a second candidate substring
806
Determine a first probability that the first candidate substring corresponds to a first actionable command and a second probability that the second candidate substring corresponds to a second actionable command
808
810
Probabilities exceed threshold?
no
yes
Determine a first intent associated with the first candidate substring and a second intent associated with the second candidate substring
812
Execute a first process associated with the first intent and a second process associated with the second intent
814
Provide to the user an acknowledgment associated with the first intent and the second intent
816

FIG. 8

(57) **Abstract**: Systems and processes are disclosed for handling a multi-part voice command for a virtual assistant. Speech input can be received from a user that includes multiple actionable commands within a single utterance. A text string can be generated from the speech input using a speech transcription process. The text string can be parsed into multiple candidate substrings based on domain keywords, imperative verbs, predetermined substring lengths, or the like. For each candidate substring, a probability can be determined indicating whether the candidate substring corresponds to an actionable command. Such probabilities can be determined based on semantic coherence, similarity to user request templates, querying services to determine manageability, or the like. If the probabilities exceed a threshold, the user intent of each substring can be determined, processes associated with the user intents can be executed, and an acknowledgment can be provided to the user.

WO 2015/184186 A1

MULTI-COMMAND SINGLE UTTERANCE INPUT METHOD

Cross-Reference to Related Application

[0001] This application claims the benefit of priority of U.S. Provisional Patent Application No. 62/005,556, entitled "MULTI-COMMAND SINGLE UTTERANCE INPUT METHOD", filed May 30, 2014; and U.S. Provisional Patent Application No.62/129,851, entitled "MULTI-COMMAND SINGLE UTTERANCE INPUT METHOD" filed March 8, 2015. The content of these applications is hereby incorporated by reference in its entirety.

Field

[0002] This relates generally to speech processing for a virtual assistant and, more specifically, to processing a single utterance having multiple actionable commands for a virtual assistant.

Background

[0003] Intelligent automated assistants (or virtual assistants) provide an intuitive interface between users and electronic devices. These assistants can allow users to interact with devices or systems using natural language in spoken and/or text forms. For example, a user can access the services of an electronic device by providing a spoken user input in natural language form to a virtual assistant associated with the electronic device. The virtual assistant can perform natural language processing on the spoken user input to infer the user's intent and operationalize the user's intent into tasks. The tasks can then be performed by executing one or more functions of the electronic device, and a relevant output can be returned to the user in natural language form.

[0004] While electronic user devices continue to provide enhanced functionality, however, some users can get overwhelmed with notifications, announcements, messages, reminders, or the like. Moreover, it can be inefficient and time consuming for users to deal with each notification, announcement, message, or reminder individually. For example, using speech to interact with a virtual assistant, a user can typically address only a single item, function, or activity at one time. In addition, users may need to wait for a virtual assistant task to be completed before moving on to another task. Such delays, in addition to limiting efficiency, can also break user concentration, which can cause users to forget other items they may have had in mind.

[0005] Accordingly, in some instances, it can be time consuming, inefficient, and frustrating for users to deal with multiple tasks—one at a time—using speech to interact with a virtual assistant.

[0005A] 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

[0005B] According to a first aspect of the invention there is provided a method for processing a multi-part voice command, the method comprising: at an electronic device: receiving speech input from a user, wherein the speech input comprises a single utterance having one or more actionable commands; generating a text string based on the speech input using a speech transcription process; parsing the text string into at least a first candidate substring and a second candidate substring; determining a first probability that the first candidate substring corresponds to a first actionable command and a second probability that the second candidate substring corresponds to a second actionable command; in response to the first probability and the second probability exceeding a threshold, determining a first intent associated with the first candidate substring and a second intent associated with the second candidate substring; executing a first process associated with the first intent and a second process associated with the second intent; and providing to the user an acknowledgment associated with the first intent and the second intent.

[0005C] According to a second aspect of the invention there is provided a non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by an electronic device with a touch-sensitive display, cause the device to perform the method of the first aspect.

[0005D] According to a third aspect of the invention there is provided an electronic device, comprising: one or more processors; memory; and one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions, which when executed by the one or more processors, cause the device to perform the method of the first aspect.

[0005E] According to a fourth aspect of the invention there is provided an electronic device, comprising means for performing the method of the first aspect.

[0005F] According to a fifth aspect of the invention there is provided an electronic device, comprising a processing unit configured to perform the method of the first aspect.

[0006] Systems and processes are disclosed for processing a multi-part voice command. In one example, speech input can be received from a user that includes a single utterance having one or more actionable commands. A text string can be generated based on the speech input using a speech transcription process. The text string can be parsed into multiple candidate substrings. Probabilities can be determined for each of the candidate substrings indicating whether they are likely to correspond to actionable commands. In response to the probabilities exceeding a threshold, user intents can be determined for each of the candidate substrings. Processes associated with the user intents can then be executed. An acknowledgment can also be provided to the user associated with the various user intents.

[0007] In some examples, the text string can be parsed by identifying domain keywords. In other examples, the text string can be parsed by identifying imperative verbs. The probability that a substring corresponds to an actionable command can be determined by determining a semantic coherence of the substring. The probability can also be determined by comparing the substring to user request templates. The probability can also be determined by submitting the substring to a service and receiving a likelihood that the service can resolve an actionable command from the substring.

[0008] In addition, in some examples, user intent for a substring can be determined based on words in a previous substring. User intent can also be determined based on displayed information. Displayed information can include a list, and user intent can be determined based on ordinal descriptors associated with items in the list. Displayed information can include notifications and emails. User intent can also be determined by determining potential user requests based on displayed information.

[0009] Moreover, in some examples, an acknowledgment can include an audible confirmation or haptic feedback. Providing an acknowledgment can also include providing tasks associated with user intents, including displaying the tasks. Providing an acknowledgment can also include providing a completion indicator, including displaying a completion indicator like a checkmark. Providing an acknowledgment can also include providing a status indicator, including displaying a status indicator like an hourglass or a status bar. In other examples, providing an acknowledgement can include displaying different candidate substrings using different forms of emphasis, such as bold text, italic text, underlined text, circled text, outlined text, colored text, and/or clustered text.

Brief Description of the Drawings

[0010] For a better understanding of the various described embodiments, reference should be made to the Description of Embodiments below, in conjunction with the following drawings in which like reference numerals refer to corresponding parts throughout the figures.

[0011] FIG. 1 is a block diagram illustrating a system and environment for implementing a digital assistant according to various examples.

[0012] FIG. 2A is a block diagram illustrating a portable multifunction device implementing the client-side portion of a digital assistant in accordance with some embodiments.

[0013] FIG. 2B is a block diagram illustrating exemplary components for event handling according to various examples.

[0014] FIG. 3 illustrates a portable multifunction device implementing the client-side portion of a digital assistant according to various examples.

[0015] FIG. 4 is a block diagram of an exemplary multifunction device with a display and a touch-sensitive surface according to various examples.

[0016] FIG. 5A illustrates an exemplary user interface for a menu of applications on a portable multifunction device according to various examples.

[0017] 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.

[0018] FIG. 6A illustrates a personal electronic device according to various examples.

[0019] FIG. 6B is a block diagram illustrating a personal electronic device according to various examples.

[0020] FIG. 7A is a block diagram illustrating a digital assistant system or a server portion thereof according to various examples.

[0021] FIG. 7B illustrates the functions of the digital assistant shown in FIG. 7A according to various examples.

[0022] FIG. 7C illustrates a portion of an ontology according to various examples.

[0023] FIG. 8 illustrates an exemplary process for handling multiple actionable commands in a single user utterance.

[0024] FIG. 9 illustrates an exemplary parsed multi-part voice command.

[0025] FIG. 10 illustrates an exemplary display with context for interpreting a multi-part voice command.

[0026] FIG. 11 illustrates an exemplary display with multiple notifications of various types usable as context for interpreting a multi-part voice command.

[0027] FIG. 12 illustrates an exemplary display with an email application usable as context for interpreting a multi-part voice command.

[0028] FIG. 13 illustrates an exemplary user interface for conveying the status of a multi-part voice command.

[0029] FIG. 14A and FIG. 14B illustrate exemplary user interfaces for conveying recognition of a multi-part voice command.

[0030] FIG. 15 illustrates a functional block diagram of an electronic device configured to process a multi-part voice command according to various examples.

Description of Embodiments

[0031] In the following description of examples, reference is made to the accompanying drawings in which it is 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.

[0032] Below, FIGS. 2A-2B, 3, 4, 5A-5B, and 6A-6B provide a description of exemplary devices for performing the techniques for processing multi-part voice commands. FIGS. 10-14B illustrate exemplary user interfaces. The user interfaces in the figures are also used to illustrate the processes described below, including the process 800 in FIG. 8.

[0033] 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 can both be outputs and, in some cases, can be separate and different inputs.

[0034] 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.

[0035] 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

[0036] FIG. 1 illustrates a block diagram of system 100 according to various examples. In some examples, system 100 can implement a digital assistant. The terms "digital assistant," "virtual assistant," "intelligent automated assistant," or "automatic digital assistant" can 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 can perform 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.

[0037] Specifically, a digital assistant can be 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 can seek either an informational answer or performance of a task by the digital assistant. A satisfactory response to the user request can be a provision of the requested informational answer, a performance of the requested task, or a combination of the two. For example, a user can ask the digital assistant a question, such as "Where am I right now?" Based on the user's current location, the digital assistant can answer, "You are in Central Park near the west gate." The user can also request 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 can sometimes interact 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 can also provide responses in other visual or audio forms, e.g., as text, alerts, music, videos, animations, etc.

[0038] As shown in FIG. 1, in some examples, a digital assistant can be implemented according to a client-server model. The digital assistant can include 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 can communicate with DA server 106 through one or more networks 110. DA client 102 can provide client-side functionalities such as user-facing input and output processing and communication with DA server 106. DA server 106 can provide server-side functionalities for any number of DA clients 102 each residing on a respective user device 104.

[0039] In some examples, DA server 106 can include 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 can facilitate the client-facing input and output processing for DA server 106. One or more processing modules 114 can 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 can communicate with external services 120 through network(s) 110 for task completion or information acquisition. I/O interface to external services 118 can facilitate such communications.

[0040] User device 104 can be any suitable electronic device. For example, user device 104 can be 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 can be, for example, a mobile telephone that also contains other functions, such as PDA and/or music player functions. Specific examples of portable multifunction devices can include the iPhone®, iPod Touch®, and iPad® devices from Apple Inc. of Cupertino, California. Other examples of portable multifunction devices can include, without limitation, laptop or tablet computers. Further, in some examples, user device 104 can be a non-portable multifunctional device. In particular, user device 104 can be a desktop computer, a game console, a television, or a television set-top box. In some examples, user device 104 can include a touch-sensitive surface (e.g., touch screen displays and/or touchpads). Further, user device 104 can optionally include 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.

[0041] Examples of communication network(s) 110 can include local area networks (LAN) and wide area networks (WAN), e.g., the Internet. Communication network(s) 110 can be 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.

[0042] Server system 108 can be implemented on one or more standalone data processing apparatus or a distributed network of computers. In some examples, server system 108 can also employ 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.

[0043] In some examples, user device 104 can communicate with DA server 106 via second user device 122. Second user device 122 can be similar or identical to user device 104. For example, second user device 122 can be similar to devices 200, 400, or 600 described below with reference to FIGs. 2A, 4, and 6A-B. User device 104 can be 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 can be configured to act as a proxy between user device 104 and DA server 106. For example, DA client 102 of user device 104 can be 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 can process the information and return relevant data (e.g., data content responsive to the user request) to user device 104 via second user device 122.

[0044] In some examples, user device 104 can be 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 can be 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 can include any number and type of user devices configured in this proxy configuration to communicate with DA server system 106.

[0045] Although the digital assistant shown in FIG. 1 can include 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 can be 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 can be 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

[0046] 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.

[0047] 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).

[0048] 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.

[0049] 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.

[0050] Memory 202 may include one or more computer-readable storage mediums. The computer-readable storage mediums may be tangible and non-transitory. Memory 202 may include high-speed random access memory and may also include 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 may control access to memory 202 by other components of device 200.

[0051] In some examples, a non-transitory computer-readable storage medium of memory 202 can be used to store instructions (e.g., for performing aspects of process 800, 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 process 800, described below) can be stored on a non-transitory computer-readable storage medium (not shown) of the server system 108 or can be divided between the non-transitory computer-readable storage medium of memory 202 and the non-transitory computer-readable storage medium of server system 108. In the context of this document, a "non-transitory computer-readable storage medium" can be any medium that can contain or store the program for use by or in connection with the instruction execution system, apparatus, or device.

[0052] Peripherals interface 218 can be 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 may be implemented on a single chip, such as chip 204. In some other embodiments, they may be implemented on separate chips.

[0053] 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.

[0054] 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 may be 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).

[0055] 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).

[0056] A quick press of the push button may disengage 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) may turn power to device 200 on or off. The user may be 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.

[0057] 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 may include graphics, text, icons, video, and any combination thereof (collectively termed "graphics"). In some embodiments, some or all of the visual output may correspond to user-interface objects.

[0058] 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.

[0059] Touch screen 212 may use 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 may 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.

[0060] A touch-sensitive display in some embodiments of touch screen 212 may be 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.

[0061] A touch-sensitive display in some embodiments of touch screen 212 may be 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.

[0062] Touch screen 212 may have a video resolution in excess of 100 dpi. In some embodiments, the touch screen has a video resolution of approximately 160 dpi. The user may make 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.

[0063] In some embodiments, in addition to the touch screen, device 200 may include 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 may be a touch-sensitive surface that is separate from touch screen 212 or an extension of the touch-sensitive surface formed by the touch screen.

[0064] Device 200 also includes power system 262 for powering the various components. Power system 262 may include 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.

[0065] Device 200 may also include 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 may include 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 may capture 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 may be 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 may be 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 may be used along with the touch screen display for both video conferencing and still and/or video image acquisition.

[0066] 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.

[0067] Device 200 may also include one or more proximity sensors 266. FIG. 2A shows proximity sensor 266 coupled to peripherals interface 218. Alternately, proximity sensor 266 may be coupled to input controller 260 in I/O subsystem 206. Proximity sensor 266 may perform 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).

[0068] 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.

[0069] Device 200 may also include one or more accelerometers 268. FIG. 2A shows accelerometer 268 coupled to peripherals interface 218. Alternately, accelerometer 268 may be coupled to an input controller 260 in I/O subsystem 206. Accelerometer 268 may perform 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.

[0070] 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 can store 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.

[0071] 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.

[0072] 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.

[0073] 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.

[0074] 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).

[0075] 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.

[0076] 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.

[0077] 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.

[0078] 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.

[0079] Text input module 234, which may be a component of graphics module 232, provides soft keyboards for entering text in various applications (e.g., contacts 237, e mail 240, IM 241, browser 247, and any other application that needs text input).

[0080] 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).

[0081] Digital assistant client module 229 can include various client-side digital assistant instructions to provide the client-side functionalities of the digital assistant. For example, digital assistant client module 229 can be 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 can also be 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 can be 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 can communicate with DA server 106 using RF circuitry 208.

[0082] User data and models 231 can 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 can includes 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.

[0083] In some examples, digital assistant client module 229 can utilize 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 can provide 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 can also use the contextual information to determine how to prepare and deliver outputs to the user. Contextual information can be referred to as context data.

[0084] In some examples, the contextual information that accompanies the user input can include 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 can be provided to DA server 106 as contextual information associated with a user input.

[0085] In some examples, the digital assistant client module 229 can selectively provide 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 can also elicit 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 can pass 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.

[0086] 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.

[0087] Applications 236 may 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 may include 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.

[0088] Examples of other applications 236 that may be 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.

[0089] In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, contacts module 237 may be 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.

[0090] 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 may be 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 may use any of a plurality of communications standards, protocols, and technologies.

[0091] 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.

[0092] 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.

[0093] 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 may 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).

[0094] 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.

[0095] 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.

[0096] 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.

[0097] 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.

[0098] 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.

[0099] 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 may 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).

[0100] 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 may be used by a user to create widgets (e.g., turning a user-specified portion of a web page into a widget).

[0101] 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.

[0102] 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.).

[0103] 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.

[0104] 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 may be 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.

[0105] 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.

[0106] 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 may be combined or otherwise rearranged in various embodiments. For example, video player module may 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 may store a subset of the modules and data structures identified above. Furthermore, memory 202 may store additional modules and data structures not described above.

[0107] 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 may be reduced.

[0108] 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.

[0109] 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).

[0110] 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.

[0111] 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.

[0112] 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.

[0113] 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).

[0114] In some embodiments, event sorter 270 also includes a hit view determination module 272 and/or an active event recognizer determination module 273.

[0115] 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.

[0116] 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 may 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 may be called the hit view, and the set of events that are recognized as proper inputs may be determined based, at least in part, on the hit view of the initial touch that begins a touch-based gesture.

[0117] 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.

[0118] 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.

[0119] 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.

[0120] 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.

[0121] 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 may utilize or call 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.

[0122] 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 may include sub-event delivery instructions).

[0123] 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 may also include 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.

[0124] 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.

[0125] 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.

[0126] 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.

[0127] 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.

[0128] 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 may 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.

[0129] 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.

[0130] 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.

[0131] 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.

[0132] 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.

[0133] 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.

[0134] 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.

[0135] Device 200 may also include one or more physical buttons, such as “home” or menu button 304. As described previously, menu button 304 may be used to navigate to any application 236 in a set of applications that may be 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.

[0136] 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.

[0137] 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.

[0138] Each of the above-identified elements in FIG. 4 may be 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 may be combined or otherwise rearranged in various embodiments. In some embodiments, memory 470 may store a subset of the modules and data structures identified above. Furthermore, memory 470 may store additional modules and data structures not described above.

[0139] Attention is now directed towards embodiments of user interfaces that may be implemented on, for example, portable multifunction device 200.

[0140] 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 may be implemented on device 400. In some embodiments, user interface 500 includes the following elements, or a subset or superset thereof:

- 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.

[0141] 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 may optionally be 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.

[0142] 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.

[0143] 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.

[0144] 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.

[0145] FIG. 6A illustrates exemplary personal electronic device 600. Device 600 includes body 602. In some embodiments, device 600 can include 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) may have 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) can provide output data that represents the intensity of touches. The user interface of device 600 can respond to touches based on their intensity, meaning that touches of different intensities can invoke different user interface operations on device 600.

[0146] Techniques for detecting and processing touch intensity may be 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.

[0147] In some embodiments, device 600 has one or more input mechanisms 606 and 608. Input mechanisms 606 and 608, if included, can be 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 may permit device 600 to be worn by a user.

[0148] FIG. 6B depicts exemplary personal electronic device 600. In some embodiments, device 600 can include 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 can be connected to display 604, which can have touch-sensitive component 622 and, optionally, touch-intensity sensitive component 624. In addition, I/O section 614 can be 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 can include input mechanisms 606 and/or 608. Input mechanism 606 may be a rotatable input device or a depressible and rotatable input device, for example. Input mechanism 608 may be a button, in some examples.

[0149] Input mechanism 608 may be a microphone, in some examples. Personal electronic device 600 can include 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 can be operatively connected to I/O section 614.

[0150] Memory 618 of personal electronic device 600 can be a non-transitory computer-readable storage medium, for storing computer-executable instructions, which, when executed by one or more computer processors 616, for example, can cause the computer processors to perform the techniques described below, including process 800 (FIG. 8). The computer-executable instructions can also be 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. For purposes of this document, a "non-transitory computer-readable storage medium" can be any medium that can tangibly contain or store computer-executable instructions for use by or in connection with the instruction execution system, apparatus, or device. The non-transitory computer-readable storage medium can include, but is not limited to, magnetic, optical, and/or semiconductor storages. Examples of such storage include magnetic disks, optical discs based on CD, DVD, or Blu-ray technologies, as well as persistent solid-state memory such as flash, solid-state drives, and the like. 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.

[0151] As used here, the term "affordance" refers to a user-interactive graphical user interface object that may be 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) may each constitute an affordance.

[0152] 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).

[0153] 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 may include 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.

[0154] In some embodiments, a portion of a gesture is identified for purposes of determining a characteristic intensity. For example, a touch-sensitive surface may receive 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 may be 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 may be 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.

[0155] The intensity of a contact on the touch-sensitive surface may be 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.

[0156] 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.

[0157] 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).

[0158] 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).

[0159] 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

[0160] FIG. 7A illustrates a block diagram of digital assistant system 700 in accordance with various examples. In some examples, digital assistant system 700 can be implemented on a standalone computer system. In some examples, digital assistant system 700 can be distributed across multiple computers. In some examples, some of the modules and functions of the digital assistant can be 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 can be 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, may combine two or more components, or may have a different configuration or arrangement of the components. The various components shown in FIG. 7A can be 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.

[0161] Digital assistant system 700 can include 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.

[0162] In some examples, memory 702 can include 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).

[0163] In some examples, I/O interface 706 can couple 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, can receive 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 can include 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 can represent 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).

[0164] In some examples, the network communications interface 708 can include wired communication port(s) 712 and/or wireless transmission and reception circuitry 714. The wired communication port(s) can receive and send communication signals via one or more wired interfaces, e.g., Ethernet, Universal Serial Bus (USB), FIREWIRE, etc. The wireless circuitry 714 can receive and send RF signals and/or optical signals from/to communications networks and other communications devices. The wireless communications can 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 can enable 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.

[0165] In some examples, memory 702, or the computer-readable storage media of memory 702, can store 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, can store instructions for performing process 800, described below. One or more processors 704 can execute these programs, modules, and instructions, and reads/writes from/to the data structures.

[0166] Operating system 718 (e.g., Darwin, RTXC, LINUX, UNIX, iOS, OS X, WINDOWS, or an embedded operating system such as VxWorks) can include 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.

[0167] Communications module 720 can facilitate communications between digital assistant system 700 with other devices over network communications interface 708. For example, communications module 720 can communicate 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 can also include various components for handling data received by wireless circuitry 714 and/or wired communications port 712.

[0168] User interface module 722 can receive 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 can also prepare and deliver 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.).

[0169] Applications 724 can 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 can 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 can include resource management applications, diagnostic applications, or scheduling applications, for example.

[0170] Memory 702 can also store digital assistant module 726 (or the server portion of a digital assistant). In some examples, digital assistant module 726 can include 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 can have 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.

[0171] 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.

[0172] In some examples, as shown in FIG. 7B, I/O processing module 728 can interact 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 can optionally obtain 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 can include 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 can also send 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 can include speech input, I/O processing module 728 can forward the speech input to STT processing module 730 (or speech recognizer) for speech-to-text conversions.

[0173] STT processing module 730 can include 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 can include a front-end speech pre-processor. The front-end speech pre-processor can extract representative features from the speech input. For example, the front-end speech pre-processor can perform 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 can include one or more speech recognition models (e.g., acoustic models and/or language models) and can implement one or more speech recognition engines. Examples of speech recognition models can include Hidden Markov Models, Gaussian-Mixture Models, Deep Neural Network Models, n-gram language models, and other statistical models. Examples of speech recognition engines can 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 can be 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 can be 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 can be passed to natural language processing module 732 for intent deduction.

[0174] 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.

[0175] In some examples, STT processing module 730 can include and/or access a vocabulary of recognizable words via phonetic alphabet conversion module 731. Each vocabulary word can be associated with one or more candidate pronunciations of the word represented in a speech recognition phonetic alphabet. In particular, the vocabulary of recognizable words can include a word that is associated with a plurality of candidate pronunciations. For example, the vocabulary may include the word "tomato" that is associated with the candidate pronunciations of /tə'meɪroʊ/ and /tə'mɑtoʊ/. Further, vocabulary words can be associated with custom candidate pronunciations that are based on previous speech inputs from the user. Such custom candidate pronunciations can be stored in STT processing module 730 and can be associated with a particular user via the user's profile on the device. In some examples, the candidate pronunciations for words can be determined based on the spelling of the word and one or more linguistic and/or phonetic rules. In some examples, the candidate pronunciations can be manually generated, e.g., based on known canonical pronunciations.

[0176] In some examples, the candidate pronunciations can be ranked based on the commonness of the candidate pronunciation. For example, the candidate pronunciation /tə'meɪɾoʊ/ can be ranked higher than /tə'mɑtoʊ/, 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 can be ranked based on whether the candidate pronunciation is a custom candidate pronunciation associated with the user. For example, custom candidate pronunciations can be 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 can be associated with one or more speech characteristics, such as geographic origin, nationality, or ethnicity. For example, the candidate pronunciation /tə'meɪɾoʊ/ can be associated with the United States, whereas the candidate pronunciation /tə'mɑtoʊ/ can be associated with Great Britain. Further, the rank of the candidate pronunciation can be 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'meɪɾoʊ/ (associated with the United States) can be ranked higher than the candidate pronunciation /tə'mɑtoʊ/ (associated with Great Britain). In some examples, one of the ranked candidate pronunciations can be selected as a predicted pronunciation (e.g., the most likely pronunciation).

[0177] When a speech input is received, STT processing module 730 can be 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 can first identify the sequence of phonemes /tə'meɪɾoʊ/ 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.”

[0178] In some examples, STT processing module 730 can use approximate matching techniques to determine words in an utterance. Thus, for example, the STT processing module 730 can determine that the sequence of phonemes /tə'meɪroʊ/ corresponds to the word “tomato,” even if that particular sequence of phonemes is not one of the candidate sequence of phonemes for that word.

[0179] Natural language processing module 732 (“natural language processor”) of the digital assistant can take 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” can represent 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 can be 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 can be 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, can also be dependent on the assistant’s ability to infer the correct “actionable intent(s)” from the user request expressed in natural language.

[0180] In some examples, in addition to the sequence of words or tokens obtained from STT processing module 730, natural language processing module 732 can also receive contextual information associated with the user request, e.g., from I/O processing module 728. The natural language processing module 732 can optionally use 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 can include, 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 can be dynamic, and can change with time, location, content of the dialogue, and other factors.

[0181] In some examples, the natural language processing can be based on, e.g., ontology 760. Ontology 760 can be 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" can represent a task that the digital assistant is capable of performing, i.e., it is "actionable" or can be acted on. A "property" can represent 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 can define how a parameter represented by the property node pertains to the task represented by the actionable intent node.

[0182] In some examples, ontology 760 can be made up of actionable intent nodes and property nodes. Within ontology 760, each actionable intent node can be linked to one or more property nodes either directly or through one or more intermediate property nodes. Similarly, each property node can be 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 can include a "restaurant reservation" node (i.e., an actionable intent node). Property nodes "restaurant," "date/time" (for the reservation), and "party size" can each be directly linked to the actionable intent node (i.e., the "restaurant reservation" node).

[0183] In addition, property nodes "cuisine," "price range," "phone number," and "location" can be sub-nodes of the property node "restaurant," and can each be 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 can also include a "set reminder" node (i.e., another actionable intent node). Property nodes "date/time" (for setting the reminder) and "subject" (for the reminder) can each be linked to the "set reminder" node. Since the property "date/time" can be relevant to both the task of making a restaurant reservation and the task of setting a reminder, the property node "date/time" can be linked to both the "restaurant reservation" node and the "set reminder" node in ontology 760.

[0184] An actionable intent node, along with its linked concept nodes, can be described as a "domain." In the present discussion, each domain can be 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 can include 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 can include the actionable intent node "set reminder," and property nodes "subject" and "date/time." In some examples, ontology 760 can be made up of many domains. Each domain can share one or more property nodes with one or more other domains. For example, the "date/time" property node can be 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.

[0185] While FIG. 7C illustrates two example domains within ontology 760, other domains can 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 can be associated with a "send a message" actionable intent node, and may further include property nodes such as "recipient(s)," "message type," and "message body." The property node "recipient" can be further defined, for example, by the sub-property nodes such as "recipient name" and "message address."

[0186] In some examples, ontology 760 can include all the domains (and hence actionable intents) that the digital assistant is capable of understanding and acting upon. In some examples, ontology 760 can be modified, such as by adding or removing entire domains or nodes, or by modifying relationships between the nodes within the ontology 760.

[0187] In some examples, nodes associated with multiple related actionable intents can be clustered under a "super domain" in ontology 760. For example, a "travel" super-domain can include a cluster of property nodes and actionable intent nodes related to travel. The actionable intent nodes related to travel can include "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) can 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” can share one or more of the property nodes “start location,” “destination,” “departure date/time,” “arrival date/time,” and “party size.”

[0188] In some examples, each node in ontology 760 can be 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 can be the so-called “vocabulary” associated with the node. The respective set of words and/or phrases associated with each node can be 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” can include 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” can include words and phrases such as “call,” “phone,” “dial,” “ring,” “call this number,” “make a call to,” and so on. The vocabulary index 744 can optionally include words and phrases in different languages.

[0189] Natural language processing module 732 can receive the token sequence (e.g., a text string) from STT processing module 730, and determine 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 can “trigger” or “activate” those nodes. Based on the quantity and/or relative importance of the activated nodes, natural language processing module 732 can select 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 can be selected. In some examples, the domain having the highest confidence value (e.g., based on the relative importance of its various triggered nodes) can be selected. In some examples, the domain can be 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.

[0190] User data 748 can include 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 can use 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 can be 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.

[0191] 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.

[0192] 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 can generate a structured query to represent the identified actionable intent. In some examples, the structured query can include 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 may say "Make me a dinner reservation at a sushi place at 7." In this case, natural language processing module 732 can be 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 may include 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 can generate 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} may not be specified in the structured query based on the information currently available.

In some examples, natural language processing module 732 can populate 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 can populate a {location} parameter in the structured query with GPS coordinates from the user device.

[0193] In some examples, natural language processing module 732 can pass the generated structured query (including any completed parameters) to task flow processing module 736 ("task flow processor"). Task flow processing module 736 can be 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 can be provided in task flow models 754. In some examples, task flow models 754 can include procedures for obtaining additional information from the user and task flows for performing actions associated with the actionable intent.

[0194] As described above, in order to complete a structured query, task flow processing module 736 may need 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 can invoke dialogue flow processing module 734 to engage in a dialogue with the user. In some examples, dialogue flow processing module 734 can determine how (and/or when) to ask the user for the additional information and receives and processes the user responses. The questions can be provided to and answers can be received from the users through I/O processing module 728. In some examples, dialogue flow processing module 734 can present 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 can generate 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 can then populate 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.

[0195] Once task flow processing module 736 has completed the structured query for an actionable intent, task flow processing module 736 can proceed to perform the ultimate task associated with the actionable intent. Accordingly, task flow processing module 736 can execute 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" can include 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 can perform 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.

[0196] In some examples, task flow processing module 736 can employ 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 can act 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 can be specified by a respective service model among service models 756. Service processing module 738 can access 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.

[0197] For example, if a restaurant has enabled an online reservation service, the restaurant can submit 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 can establish 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.

[0198] In some examples, natural language processing module 732, dialogue flow processing module 734, and task flow processing module 736 can be 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 can be a dialogue response to the speech input that at least partially fulfills the user's intent. Further, in some examples, the generated response can be output as a speech output. In these examples, the generated response can be 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 can be data content relevant to satisfying a user request in the speech input.

[0199] Speech synthesis module 740 can be 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 can be in the form of a text string. Speech synthesis module 740 can convert the text string to an audible speech output. Speech synthesis module 740 can use 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 can be configured to synthesize individual words based on phonemic strings corresponding to the words. For example, a phonemic string can be associated with a word in the generated dialogue response. The phonemic string can be stored in metadata associated with the word. Speech synthesis model 740 can be configured to directly process the phonemic string in the metadata to synthesize the word in speech form.

[0200] In some examples, instead of (or in addition to) using speech synthesis module 740, speech synthesis can be performed on a remote device (e.g., the server system

108), and the synthesized speech can be 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 can be possible to obtain higher quality speech outputs than would be practical with client-side synthesis.

[0201] 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.

[0202] FIG. 8 illustrates exemplary process 800 for handling multiple actionable commands in a single user utterance according to various examples. Process 800 can, for example, be executed on processing modules 114 of server system 108 discussed above with reference to FIG. 1. In other examples, process 800 can be executed on processor 704 of digital assistant system 700 discussed above with reference to FIG. 7. In still other examples, processing modules 114 of server system 108 and processor 704 of digital assistant system 700 can be used together to execute some or all of process 800. At block 802, speech input can be received from a user (e.g., from microphone 213 of Fig. 2). In some examples, the speech input can be directed to a virtual assistant, and can include one actionable command (e.g., "What's the weather going to be today?") or multiple actionable commands (e.g., "Navigate to Jessica's house and send her a message that I'm on my way."). An actionable command as used herein can include any task, process, query, action, request, or the like that a virtual assistant can perform, respond to, or otherwise handle. For example, an actionable command can include a query about a stock price, a request to send a message, a command to provide navigation services, a command to initiate a phone call, or the like.

[0203] In some examples, a user can control the capture of a multi-part command by signaling with a button, touchscreen interface, or the like. For example, to cause a user's device to continue listening for additional commands, a user can push and hold a button (e.g., of user device 104) while speaking, and can release the button after finishing as many commands as the user desires to speak. In other examples, a user device can continue listening while speech continues, or can include an alwayslistening function that monitors audio continuously (e.g., at all times or for some extended duration).

[0204] A multi-part command within a single utterance can include any number of commands with any number of arguments. For example, a user can request multiple actions associated with the same subject (e.g., email a picture to a friend, save the picture to memory, and set the picture as the wallpaper on a device display). In another example, a user can request multiple actions associated with different subjects (e.g., email a picture to a friend, send a message to a relative saying "I'll be there soon," dismiss a calendar notification, and remind the user of a particular task in an hour). Some speech input can also mix commands with dictation, as in the example of dictating a message to be sent while requesting other actions in the same utterance. Speech input with a single utterance can thus include any number of commands associated with any number of arguments.

[0205] At block 804, a text string can be generated based on the speech input using a speech transcription process. Any of a variety of speech transcription approaches can be used. In addition, in some examples, multiple possible transcriptions can be generated and processed in sequence or simultaneously to identify the best possible match (e.g., the most likely match).

[0206] At block 806, the text string (or multiple candidate text strings) can be parsed into at least a first candidate substring and a second candidate substring. In one example, any speech input can be parsed into any number of candidate substrings to test for multi-part commands. In other examples, parsing into multiple candidate substrings can be done based on the length of the speech input, a user indication (e.g., holding a button down while speaking multiple commands), identification of multiple imperative verbs, or the like.

[0207] FIG. 9 illustrates an exemplary parsed multi-part voice command. User speech 920 can include the transcription of a single utterance saying, "Navigate to Jessica's house and send her a message that I'm on my way." In one example, user speech 920 can be parsed into two substrings: first candidate substring 922 and second candidate substring 924. Such parsing can be done in a variety of ways. For example, domain keywords can be identified in the text string, and the string can be parsed based on their location. In the example of FIG. 9, the word "navigate" can correspond to a virtual assistant domain for providing maps, turn-by-turn navigation instructions, or other navigation assistance to a user. The word “send” can correspond to a messaging domain or multiple virtual assistant domains (e.g., email, instant messages, text messages, messaging applications, or the like). The word “message” can likewise correspond to multiple virtual assistant domains. In some instances, the combination of “send” and “message” within a certain number of words of one another can indicate a likely matching virtual assistant domain.

[0208] Based on the positions of the identified domain keywords, user speech 920 can be split into first candidate substring 922 beginning with the keyword “Navigate” and ending prior to the conjunction “and,” and second candidate substring 924 beginning with the keyword “send” and ending with the end of the string. In other examples, the conjunction “and” can be included within either candidate substring or both candidate substrings. Various other domain keywords can be used in a similar manner to parse a string. For example, a weather domain might be associated with keywords or key phrases like “weather,” “temperature,” “how cold,” “how hot,” “rain,” etc. Similarly, a phone domain might be associated with keywords “call,” “dial,” etc. A calendar domain might be associated with keywords or key phrases like “meeting,” “set up a meeting,” “meet with,” “new appointment,” “schedule,” etc. A reminder domain might be associated with keywords or key phrases like “remind,” “reminder,” “set a reminder,” “remind me,” etc. It should be understood that various other domains can have a variety of associated keywords or key phrases that can be used to parse a text string.

[0209] Instead of or in combination with domain keywords, imperative verbs can be identified in the text string, and the string can be parsed based on their location. In the example of FIG. 9, the word “Navigate” is the first imperative verb in the string. The word “send” is the next imperative verb in the string. In some instances, the word “message” can be interpreted as another imperative verb. In this example, however, a variety of factors can cause “message” to correctly be recognized as a noun: it is preceded with “a,” it follows a few words behind “send,” it is followed by “that,” etc. Two imperative verbs can thus be identified in user speech 920. The positions of the imperative verbs can then be used to parse the string into first candidate substring 922 beginning with the imperative verb “Navigate” and second candidate substring 924 beginning with the imperative verb “send.” Other imperative verb examples include words like “schedule,” “set up,” “call,” “email,” “text,” “post,” “play,” “launch,” “remind,” “note,” “turn on,” “search,” etc. In addition, in some examples, a user utterance can be broken into the various parts of speech to aid in correctly identifying imperative verbs (e.g., as opposed to homonyms that might be nouns).

[0210] Instead of or in combination with domain keywords and/or imperative verbs, a single user utterance can be parsed in multiple ways, and the multiple parses can be tested to determine the best parse. In other examples, a string can be parsed based on a predetermined substring length (e.g., a number of characters, a number of words, etc.). In still other examples, a string can be parsed in all possible ways (e.g., ranging from each individual word as a substring to all words together in a single string) or nearly all possible ways (e.g., ranging from a minimum of each pair or triple of words as a substring to all words together in a single string). Given multiple ways of parsing the string, parse results can be analyzed to dismiss various parse approaches based on a variety of factors. For example, if a parse results in a candidate substring without a verb, that parse can be dismissed. In some examples, some or all of the various parse results can be analyzed as described below with reference to block 808 of process 800 to identify the parse with the highest likelihood of accurately reflecting a user’s intentions. A text string from a single utterance of user speech can thus be parsed in a variety of ways into at least a first candidate substring and a second candidate substring.

[0211] Referring again to process 800 of FIG. 8, at block 808, a first probability that the first candidate substring corresponds to a first actionable command and a second probability that the second candidate substring corresponds to a second actionable command can be determined. In some examples, each candidate substring can be analyzed to determine a probability that it corresponds to a valid, actionable command. As noted above, this can be done, in some examples, to verify the accuracy of a particular parse and/or to compare the likelihoods of different parses resulting in different candidate substrings. In some examples, the probabilities can be used to select the parse and its associated substrings that may be most likely to accurately reflect the user’s intention (e.g., accurately split up a single utterance into distinct actionable commands). The probabilities for the various candidate substrings can be determined in a variety of ways, and multiple approaches can be combined to arrive at an integrated probability (e.g., using linear interpolation or other combination mechanisms).

[0212] In one example, each candidate substring (from one parse or from multiple different potential parses) can be analyzed for semantic coherence. In some examples, semantic coherence can be determined for each candidate substring based on only the words within the substring (e.g., without considering context or previous or subsequent words). Semantic coherence can include a binary yes/no result or a probability. Semantic coherence can reflect whether a substring can stand alone as a coherent command (or request), having meaning in the target language (e.g., English). For example, as noted above, a substring without a verb could be identified as lacking semantic coherence or as having a low probability of being semantically coherent (e.g., "to Jessica's house," "progress report meeting," etc.). In another example, a substring lacking a subject, argument, entity, or the like could be identified as lacking semantic coherence or as having a low probability of being semantically coherent (e.g., "Navigate to," "Tell John that," etc.). In yet another example, a substring with an incomplete prepositional phrase or preposition lacking a noun could be identified as lacking semantic coherence or as having a low probability of being semantically coherent (e.g., "Set a reminder to buy milk at," "Schedule a meeting for," etc.). In addition, in some examples, a candidate substring can be analyzed according to the parts of speech to verify adherence to grammar rules, and semantic coherence can be determined accordingly. It should be understood that semantic coherence can be determined for a substring in a variety of other ways, and different rules and tests can be applied for different languages, dialects, regions, or the like.

[0213] Instead of or in combination with semantic coherence, each candidate substring can be analyzed in view of user request templates of a virtual assistant. For example, a virtual assistant can process user requests in part by matching a spoken request to a template. The template can have associated processes, tasks, algorithms, or the like that a virtual assistant can use to handle a user's request. In addition, a template can have designated variables where entities, arguments, or the like can be expected (e.g., a contact name in the template "Call [contact]."). Each candidate substring can thus be compared to some or all of the various templates of a virtual assistant.

[0214] In one example, a similarity score or probability of a match can be determined based on how similar a substring is to a user request template. In some examples, words corresponding to expected entities, arguments, or the like can be ignored in the comparison. In another example, a binary yes/no result can be produced based on whether or not one or more matching (or nearly matching) templates can be found from the comparison.

[0215] Referring to first candidate substring 922 of FIG. 9 as an example, the substring "Navigate to Jessica's house" can be compared to some or all templates of a virtual assistant. In one example, only templates associated with navigation can be used given the keyword "Navigate." In other examples, other or all templates can be used. In a virtual assistant, templates associated with providing maps, guidance, or navigation can include "Navigate to [location]," "Get me directions to [location]," "Show me a map of [location]," or the like. In comparing candidate substring 922 to such navigation-related templates, at least one matching template can be identified: "Navigate to [location]," where the location variable can allow the words "Jessica's house" to be ignored for purposes of comparison. In this example, at least one matching template can be identified, so a yes result or a high similarity score or probability can be produced.

[0216] In another example, a substring might be similar to a template without actually matching the template (e.g., having extraneous words, including unexpected conjunctions, etc.). In such examples, a lower similarity score or probability can be produced, or, depending on a particular application, either a yes or a no result can be produced. In other examples, no matching or similar template may be found, and a no result or nil similarity score or probability can be produced. Candidate substrings can thus be compared to user request templates of a virtual assistant to identify matches, likelihoods of a match, similarity scores, or the like. It should be understood that any of a variety of probabilities, scores, or the like can be produced as desired for a particular application based on comparisons with virtual assistant request templates.

[0217] Instead of or in combination with semantic coherence and/or template matching, each candidate substring can be analyzed by testing them with various services of a virtual assistant. For example, each candidate substring can be submitted to one or more services of the virtual assistant (e.g., a messaging service, a navigation service, a phone service, etc.). The services can be queried to determine whether they can resolve an actionable command from the candidate substring. In one example, services can deny a request outright or reject a substring (e.g., a zero likelihood of resolving the substring). In another example, services can be configured to produce a likelihood of being able to resolve the substring with more information (e.g., context, user data, etc.). In yet another example, services can be configured to produce a likelihood of the service being the appropriate service to handle a user request (e.g., based on domain keywords, available functions, and the like).

[0218] Referring to second candidate substring 924 of FIG. 9 as an example, a messaging service can, in some examples, produce a high likelihood of being able to resolve the substring into an actionable command. For example, in the substring "send her a message that I'm on my way," the words "send" and "message" can correspond to messaging domain keywords or templates and can lead a messaging service to produce a high likelihood that it will be able to resolve the substring to an actionable command. In addition, the messaging service can identify that additional context may be necessary to resolve to a complete command given that "her" may not be known from the substring alone, but the messaging service can produce a moderate likelihood of being able to resolve the substring given additional contextual information. Moreover, the messaging service can compare the substring to a messaging template to recognize that the substring may likely be intended for the messaging service rather than another service. A messaging service could thus produce a high likelihood of being able to resolve candidate substring 924 into an actionable command. In contrast, a navigation service, phone service, or search service might produce a low or nil likelihood of being able to resolve candidate substring 924 based on the lack of keywords, matching templates, salient entities, and the like. Candidate substrings can thus be submitted to services to determine the likelihood that one of the services can resolve the substring to an actionable command. It should be understood that any of a variety of probabilities, likelihoods, binary results, or the like can be produced as desired for a particular application based on submission to various virtual assistant services.

[0219] In some examples, a combination of semantic coherence, template matching, and/or service testing can be used to determine a probability that a candidate substring corresponds to an actionable command. For example, the results of two or more tests can be combined using known methods (e.g., linear interpolation, etc.). In addition, a variety of other tests or approaches can be used instead of or in addition to those discussed above to determine the probability that a candidate substring corresponds to an actionable command. For example, ontologies could be used to attempt to resolve a candidate substring into an actionable command, and a probability can be produced based on the success or failure of the ontology traversal. Various other approaches can also be used to test candidate substrings and verify the accuracy of a parse.

[0220] Referring again to process 800 of FIG. 8, at block 810, a determination can be made as to whether the probabilities determined at block 808 exceed a threshold. For example, a minimum threshold can be determined below which a particular parse can be deemed unlikely or unacceptable. In some examples, the probability of each candidate substring corresponding to an actionable command can be compared to a threshold level, and the failure of any candidate substring of a parse to meet the level can be deemed a failure of the parse leading to the "no" branch of block 810. In other examples, the probabilities of each candidate substring of a parse can be combined in a variety of ways (e.g., normalization, linear interpolation, etc.) and then compared to a threshold. The threshold level can be determined empirically, based on virtual assistant thresholds, or the like.

[0221] In other examples, however, block 808 (and block 810) can be omitted from process 800. For example, in some instances, a parse can be determined to be sufficiently accurate such that additional probability testing or verification may be unnecessary. Common user requests, common combinations in multi-part commands, high probabilities of semantically meaningful parses, or the like can be used to determine that a parse may be trusted enough to proceed without additional verification or accuracy testing. Moreover, in some examples, the approach used to parse a multi-part command can be sufficiently accurate so as to warrant omitting verification or accuracy testing. The testing described with regard to block 808 can thus be optional.

[0222] If the probabilities from block 808 fail to meet or exceed the threshold at block 810 (e.g., the "no" branch), in some examples, process 800 can return to block 806 to attempt a different parse of the text string. For example, given the failure of the parse through the first pass, a new parse can be attempted or an adjustment can be made at block 806 to attempt to identify a more accurate parse on a second or subsequent pass. In other examples, however, most or all possible parses can be tested simultaneously, in which case a failure at block 810 can result in the virtual assistant querying for more information, asking for a repeat of the commands, returning a failure message, or the like. Similarly, should repeated parses fail to yield an accurate result, the virtual assistant can query for more information, ask for a repeat of the commands, return a failure message, or the like. In some examples, if one or more of a set of candidate substrings of a parse exceeds a threshold, those candidate substrings can be processed further as discussed below while the virtual assistant addresses the remaining (failing) substring or substrings through dialogue, error messages, or the like. In this manner, portions of a multi-part voice command can be handled whether or not other portions can be accurately resolved without additional information.

[0223] If the probabilities from block 808 meet or exceed the threshold at block 810 (e.g., the "yes" branch), process 800 can continue to block 812. At block 812, a first intent associated with the first candidate substring and a second intent associated with the second candidate substring can be determined. In one example, a virtual assistant can determine a user's intent from speech input by matching the user's speech to a particular domain with tasks, processes, and the like that the virtual assistant can perform or execute. Determining a user's intent can also include resolving ambiguous words, pronouns, and the like using context (e.g., user data, displayed information, previous requests, etc.). Thus, at block 812, an intent can be determined for each candidate substring in preparation for performing according to the user's multi-part command.

[0224] In some examples, information from previous blocks of process 800 can be used to determine user intent at block 812. For example, domain keywords, request templates, service matching, and the like from previous blocks discussed above can be used to identify the user intent of a particular candidate substring.

[0225] Referring to the example of FIG. 9, an intent can be determined for first candidate substring 922 and second candidate substring 924. In one example, first candidate substring 922 can be matched to the navigation domain based on the command word "Navigate." A matching template "Navigate to [location]" can also be identified within that domain. The location "Jessica's house" can then be resolved using user data, such as identifying a house address from the user's contact information for a contact named Jessica.

[0226] Second candidate substring 924 can be matched to a messaging domain based on the words “send” and “message” and/or a corresponding template. For example, a template “send [recipient] a message that [message]” can be identified. The recipient can be resolved based on the context of the previous request from first candidate substring 922. In particular, the intended recipient “Jessica” (and her contact information from the user’s contacts) can be identified to replace “her” based on the previous command in the multi-part command. The text in the message variable can be transcribed directly from the substring (e.g., “I’m on my way.”).

[0227] In another example, context from a previous command in a multi-part command can similarly be used to resolve ambiguous words. In the multi-part command “Send Steve a message that I’ll be late, and remind me to call him in twenty minutes,” the pronoun “him” can be ambiguous without the context of the prior command. In particular, in one example, a first actionable command can be identified for sending Steve a message saying “I’ll be late.” The second actionable command, however, can utilize the context of the previous command for accurate intent interpretation of which person the user would like to be reminded to call in twenty minutes. The intent of the second command can thus be interpreted based on the context of the previous command in the utterance to accurately determine that the user would like to be reminded to call Steve in twenty minutes.

[0228] A variety of other contextual information can similarly be used to determine user intent for a particular candidate substring. For example, FIG. 10 illustrates exemplary user device 104 with display 1030 having context for interpreting a multi-part voice command. It should be understood that FIG. 10 illustrates one example of a user device 104 according to various examples herein, but many other examples are possible (including devices without displays, devices with different proportions, and the like). Likewise, it should be appreciated that although display 1030 is illustrated as being incorporated with user device 104 (e.g., as a touchscreen), in other examples, display 1030 can be separate from user device 104 (e.g., as in a television, computer monitor, separate user device, or the like).

[0229] In one example, when viewing the content shown in FIG. 10, a user can utter a multi-part command, such as “Save that picture, send it to my dad, and set it as my home screen wallpaper.” Such a multi-part command could be parsed into three candidate substrings: “Save that picture,” “send it to my dad,” and “set it as my home screen wallpaper." For the first substring, the subject "that picture" can be ambiguous without additional contextual information. Here, the content displayed on display 1030, including links 1034 and picture 1032, can be used to disambiguate (or resolve) the first substring. In particular, given the context of the appearance of picture 1032 on display 1030, the subject "that picture" can be resolved to picture 1032. The user's intent for the first substring can thus be determined to be saving picture 1032 (e.g., to memory on the user's device). Other content elements can similarly be used as context, including displayed text, emails, notifications, reminders, albums, songs, lists, calendar entries, or the like.

[0230] The second and third substrings can be similarly ambiguous with the pronoun "it" in "send it to my dad" and "set it as my home screen wallpaper." To resolve the ambiguity, context from the resolved first substring can be used. In particular, the resolution of the subject "that picture" to picture 1032 on display 1030 can be used to identify the intended subject of the second and third substrings. The user intent of the second substring can thus be determined to be sending picture 1032 to a user's father (e.g., as identified in a contact list or the like), and the user intent of the third substring can be determined to be setting picture 1032 as a home screen wallpaper of a user's device. Displayed content can thus be used as context for interpreting user intent from user speech, and commands can be disambiguated based on previously issued commands within the same utterance (or from a previous utterance).

[0231] FIG. 11 illustrates exemplary user device 104 with display 1030 having multiple notifications of various types that can be used as context for interpreting multi-part voice commands. In particular, display 1030 shows notifications 1040, including emails 1142, messages 1144 (e.g., instant messages, text messages, etc.), reminders 1146, and meeting request 1148. A multi-part voice command as discussed herein can be an efficient way for a user to manage multiple notifications at one time. For example, a user might utter "Reply to those emails with my out of office reply, remind me to call mom in twenty minutes, tell Joe sure, snooze those reminders an hour, and accept that meeting request." In one example, such a lengthy multi-part command can be uttered with pauses while a user thinks about next actions, and a virtual assistant can continue to await further commands without interrupting the user (e.g., based on a button being held down, a user preference or setting, or the like). In addition, a virtual assistant can begin to take action immediately upon interpreting user intent for one command without waiting to resolve subsequent commands in a multi-part command. For example, a virtual assistant might begin sending out of office emails to Jane Doe and Jennifer Smith while the user continues with commands related to messages 1144, reminders 1146, and meeting request 1148. In other examples, a virtual assistant can wait for a user to finish all commands before beginning to execute processes associated with them.

[0232] A multi-part command associated with the content shown in FIG. 11 can be disambiguated (as needed) using the types of notifications displayed, names associated with those notifications, details about those notifications, or the like. For example, for the substring "Reply to those emails with my out of office reply," the subject "those emails" can be ambiguous, but the content displayed can be used to determine that "those emails" correspond to emails 1142 shown on display 1030. In particular, the email notification or content type can be used to accurately associate "those emails" with the displayed email-related notifications. Similarly, the substring "tell Joe sure" can be ambiguous (e.g., which Joe), but the content displayed can be used to determine that "Joe" corresponds to the sender of one of messages 1144. In particular, the name "Joe" can be checked against names shown on display 1030 (as well as against user contacts and the like) to accurately identify which Joe the user intended.

[0233] The substring "snooze those reminders an hour" can similarly be ambiguous, but the content displayed can be used to determine that "those reminders" correspond to reminders 1146 shown on display 1030. In particular, the reminder notification type can be used to accurately associate "those reminders" with reminders 1146. Finally, the substring "accept that meeting request" can similarly be ambiguous, but the content displayed can be used to determine that "that meeting request" corresponds to meeting request 1148 shown on display 1030. In particular, the meeting request notification type can be used to accurately associate "that meeting request" with meeting request 1148.

[0234] In other examples, a user can reference notifications 1040 in other ways, and the displayed content can similarly be used to disambiguate a user's request. For example, a user can reference notifications by type (e.g., mail, reminder, etc.), names (e.g., Jane, Jennifer, Mom, etc.), subject matter (e.g., dinner invitation, report on progress, milk, progress report meeting, etc.), position reference or ordinal descriptor (e.g., the top two, the first email, the second reminder, the last three, etc.), or the like.

[0235] FIG. 12 illustrates exemplary user device 104 with display 1030 having multiple emails 1142 shown in email application 1250 that can be used as context for interpreting a multi-part voice command. A multi-part voice command as discussed herein can be an efficient way for a user to manage lists of items, including the list of emails 1142 illustrated in FIG. 12. It should be understood that email application 1250 is provided as an example of how emails might be displayed in a list, but many other configurations are possible (e.g., including a preview pane or the like), and any type of list can be used as context for interpreting multi-part voice commands.

[0236] In one example, a user can refer to listed emails 1142 by ordinal descriptors in issuing commands. For example, a user might utter “Send my out of office reply to the first three, add the sender of the last one to the blocked contact list, and move the last one to spam.” The ordinal position of emails 1142 within the list can then be used to disambiguate the user’s multi-part command. The descriptor “first three” in the first substring can be ambiguous, but the content displayed can be used to determine that “the first three” corresponds to the first three emails 1142 shown on display 1030 (e.g., emails from Jane, Jennifer, and Alan). In particular, the ordinal position of the first three emails can be used to accurately associate “the first three” with the first, second, and third emails 1142 in the list to determine the user’s intent. Similarly, the descriptor “the last one” in the second and third substrings can be ambiguous, but the content displayed can be used to determine that “the last one” corresponds to the last email 1142 shown on display 1030 (e.g., the email from Investor Tip Co.). In particular, the ordinal position of the fourth and final email in the list can be used to accurately associate “the last one” with the fourth email 1142 in the list to determine the user’s intent. In other examples, a user can refer to listed items in other ways that can be similarly disambiguated (e.g., second email, third reminder, second to last, penultimate, final, middle one, last two entries, top three items, bottom four, etc.).

[0237] In other examples, a user can refer to an entire list of content in issuing commands. For example, a user might utter “Mark all of those as read, and move them all to trash.” The appearance of a list of emails 1142 on display 1030 can be used to disambiguate the user’s multi-part command. The descriptors “all of those” and “them all” can be ambiguous, but the content displayed can be used to determine that “all of those” and “them all” correspond to the complete list of four emails 1142 shown in FIG. 12. In still other examples, various other ambiguous reference terms can be employed and similarly disambiguated based on list content.

[0238] In addition to using displayed content to disambiguate terms in a multi-part command, various other methods can be employed for determining user intent for candidate substrings. In one example, potential (or likely) user requests can be determined based on displayed information, and user intent can be determined based on the potential user requests. Referring again to FIG. 12 as an example, a user may be likely to issue a certain set of commands related to email management when viewing emails 1142 in email application 1250. For example, potential user requests related to email management can include replying, deleting, moving, filing, forwarding, marking as read, marking as unread, marking as spam, archiving, blocking a sender, and the like. When determining user intent for substrings, these potential user requests can be given additional weight or priority, or can be used as a comparison (e.g., comparing templates) to accurately interpret user commands. In particular, based on emails 1142 appearing on display 1030, potential user requests associated with email management can be used in interpreting user commands based on at least some likelihood that a user will issue commands associated with what is displayed.

[0239] In other examples, potential user requests can be determined based on a variety of other content to provide accurate intent interpretation. For example, referring again to FIG. 11, potential user requests can be identified for handling notifications 1140, and these potential user requests can be used in determining user intent. Associated commands can include dismissing an item, snoozing a reminder, replying to a message, accepting a meeting request, proposing a new time for a meeting, or the like. In another example, referring again to FIG. 10, potential user requests can be identified for interacting with the displayed content, including links 1034 and picture 1032, and these potential user requests can be used in determining user intent. Associated commands can include saving a picture, sharing a picture, sharing a link, printing a page, selecting a displayed item, or the like. Potential user requests can thus be identified based on displayed content, and the identified potential user requests can be used to determine user intent for substrings in a multi-part command.

[0240] Referring again to process 800 of FIG. 8, at block 814, a first process associated with the first intent and a second process associated with the second intent can be executed. With user intents determined for each candidate substring (or some candidate substrings), the processes associated with the user intents can be executed. For example, messages can be composed and sent, emails can be deleted, notifications can be dismissed, or the like. In some examples, multiple tasks or processes can be associated with individual user intents, and the various tasks or processes can be executed at block 814. In other examples, the virtual assistant can engage the user in a dialogue to acquire additional information as necessary for completing task flows. As mentioned above, in some examples, if only a subset of the substrings of a multi-part command can be interpreted into user intents, the processes associated with those user intents can be executed, and the virtual assistant can handle the remaining substrings in a variety of ways (e.g., request more information, return an error, etc.).

[0241] Referring again to process 800 of FIG. 8, at block 816, an acknowledgment associated with the first intent and the second intent can be provided to the user. For example, an acknowledgment can be provided to the user to indicate acceptance of one or more commands, status of executing various commands, interpretations of various commands, errors associated with particular commands, information being needed for some commands, or the like. In one example, such an acknowledgment can include an audible confirmation (e.g., a tone, spoken words, or the like). For example, a virtual assistant can repeat back to the user a list of commands—as interpreted—as confirmation (or to allow the user the opportunity to correct, interject, cancel, etc.). In another example, a confirmatory tone can be played to indicate valid interpretation of a user's commands. Various other audible forms of feedback can likewise be used.

[0242] In another example, acknowledgment of a multi-part command can include haptic feedback. For example, a user device can be vibrated to either confirm valid interpretation of a user's commands or to indicate an error or lack of information. Haptic feedback can also be performed in patterns to indicate certain information, such as vibrating a set number of pulses to indicate the number of identified commands.

[0243] In yet another example, acknowledgment of a multi-part command can include providing tasks to the user by, for example, speaking the tasks and/or displaying them on a display. In particular, user intents (e.g., the first intent and the second intent of blocks 812, 814, and 816 of process 800) can be associated with particular virtual assistant tasks, and the virtual assistant can paraphrase those tasks and provide them to the user by, for example, speaking and/or displaying them.

[0244] FIG. 13 illustrates exemplary user device 104 with display 1030 showing paraphrased tasks 1360 associated with a multi-part command. It should be understood that tasks 1360 can be spoken instead of or in addition to being displayed. As illustrated, three tasks are displayed as an acknowledgment to the user of a multi-part command. For example, such a command might have been uttered with reference to FIG. 12, and could include a multi-part command such as “Put the last email into spam, and add its sender to the blocked list, and reply to the first three saying I’m on vacation in Bali.” Three substrings may have been identified that could be associated with the paraphrased tasks shown in FIG. 13. In particular, a first paraphrased task can include “Put ‘Great New Opportunity’ email into spam folder,” a second paraphrased task can include “Add ‘Investor Tip Co.’ to blocked sender list,” and a third paraphrased task can include “Send email to Jane, Jennifer, and Alan: ‘I’m on vacation in Bali.’” Acknowledgment of a multi-part user command can thus include providing a paraphrased task to the user for each interpreted command by speaking and/or displaying them.

[0245] In some examples, paraphrased tasks can be displayed within a dialogue interface of a virtual assistant (e.g., in a conversation format), with or without a transcription of a user’s speech. In other examples, paraphrased tasks can be displayed in a pop-up window, notification area, or the like. In still other examples, displayed tasks 1360 (or a transcription of a user’s speech) can be selectable for editing, canceling, prioritizing, or the like. For example, a user can select one of tasks 1360, and a menu of actions can be provided for editing the task parameters, canceling the task, prioritizing the task, delaying the task, or the like. In another example, a default action can be associated with selecting a displayed task, such as editing the task, pausing the task, canceling the task, or the like.

[0246] Moreover, in some examples, completion indicators and/or status indicators can be provided to the user by, for example, playing an associated indicator tone, speaking an associated status indicator, or displaying associated indicators. FIG. 13 illustrates exemplary indicators associated with tasks 1360. In one example, upon completion of a process or task, a completion indicator can be provided to the user, such as displaying checkmark 1362 by the first task 1360 to indicate completion. In another example, a tone could be played or words could be spoken to indicate completion (e.g., "first task completed," or the like).

[0247] In another example, while a process or task is still being executed (before completion), a processing status indicator can be provided to the user. Status could be provided by sounding a tone or speaking the status (e.g., "processing the second task," or the like). Status could also be provided by displaying a processing status indicator such as hourglass 1364 and/or status bar 1366 (or an empty box where checkmark 1362 can be placed upon completion of the task). In one example, hourglass 1364 can indicate that the task is still being executed. In other examples, however, hourglass 1364 can indicate that more information is needed, that the request is still being interpreted, that a search is being conducted, or the like. Status bar 1366 can also be used to indicate task execution status, and can reflect the percentage of completion of a particular task. In other examples, processing status indicators can include animations, graphs, font changes (e.g., text color, size, etc.), or the like.

[0248] Various other acknowledgments can also be provided to a user to convey information about a multi-part command (e.g., illuminating indicator lights, animating emails being trashed, animating messages being composed and sent, etc.). In addition, any combination of acknowledgments can be provided, such as displaying a list of tasks, speaking the tasks aloud, and displaying status and completion indicators.

[0249] Furthermore, in some examples, acknowledgments can be provided while a user is still uttering commands. For example, confirmatory tones, haptic feedback, speech, or the like can be provided to a user upon, for example, detecting a complete actionable command. In particular, a virtual assistant can briefly speak "okay," play a tone, vibrate a user device, display user speech, or the like as commands are being spoken. This can provide confirmation to a user that a command is understood, thereby allowing the user to continue issuing commands with confidence that the virtual assistant is handling the commands along the way.

[0250] In other examples, the virtual assistant can provide visual confirmation to the user that it is recognizing multiple commands in a single stream of user speech. FIG. 14A and FIG. 14B illustrate exemplary virtual assistant interfaces 1470 for conveying recognition of a multi-part voice command on display 1030 of user device 104. In one example, different commands recognized within a stream of user speech can be emphasized differently than other commands to indicate that the virtual assistant has correctly recognized that a multi-part command is being issued. In one example, such emphasis can be added after the user finishes speaking. In other examples, however, the user's speech can be transcribed as the user speaks, and the candidate substrings can be dynamically emphasized differently while the transcribed text is streaming on a display of a user device. This dynamic feedback while the user is speaking can confirm to the user that the virtual assistant is understanding that a multi-part command is being issued, which can allow the user to confidently continue to issue commands as desired.

[0251] FIG. 14A illustrates one example of emphasizing different candidate substrings differently in a virtual assistant interface 1470. As illustrated, interface 1470 can include a conversational dialogue interface with assistant greeting 1472 prompting a user to make a request. Transcribed user speech 1474 illustrates one example of differently emphasizing candidate substrings in user speech as it is transcribed. In particular, the first candidate substring is shown in bold, italic, and underlined text. The second candidate substring is shown in bold, underlined text. The third candidate substring is shown in italic, underlined text. In this manner, different candidate substrings or commands can be emphasized so as to convey that the virtual assistant recognizes that a multi-part command is being issued, and that the virtual assistant is correctly recognizing the different commands individually.

[0252] In the example of FIG. 14A, transcribed user speech 1474 is shown as incomplete as the user may still be issuing a command. In particular, the third command begins with "reply to the first three saying," but the text of the associated message has not yet been spoken. In some examples, auto-complete suggestions 1476 can be displayed, and the user can select one of the auto-complete suggestions to complete a command that is currently being issued. For example, the user can tap on the auto-complete suggestion "I'm out of the office" to complete the third command and provide the message text of the reply. In other examples, auto-complete suggestions 1476 can provide example commands related to previous commands in an utterance, commands related to objects shown on a display, commands frequently issued by the user, or the like. Moreover, in some examples, auto-complete suggestions 1476 can be positioned next to, in-line with, or in another position relative to the current command being issued. For example, auto-complete suggestions 1476 can be positioned near an object or partial command and connected to it with a line or other graphical association (e.g., dots).

[0253] FIG. 14B illustrates another example of emphasizing different candidate substrings differently in a virtual assistant interface 1470. In particular, transcribed user speech 1478 illustrates emphasizing different candidate substrings using tag-cloud-like clustering. In one examples, the various commands can be spatially separated with the words of each command clustered together. In some examples, words that can be edited, keywords, command words, subject words, or otherwise important words can be emphasized with bold text, larger font size, or the like. In other examples, words can be emphasized differently based on importance. For example, the primary command word (e.g., send) can be shown in the largest, boldest font, subject words (e.g., recipient names) can be shown in a medium font, and other words can be shown in a small font. The words can be clustered together in variety of ways, including out of order and in different orientations. In some examples, different commands can be sized differently based on interpretation confidence associated with a command (e.g., how confident the virtual assistant is that the interpretation is accurate based on the user's speech). Command clusters can be separated and displayed as a user is speaking commands or after a user finishes speaking a multi-part command. Various other cluster-type displays can also be used to emphasize multi-part command distinctions.

[0254] Different commands or candidate substrings in a multi-part command can also be emphasized in a variety of other ways. For example, individual commands can be circled, outlined, or otherwise separated by drawing graphical markers (e.g., drawing a shape around a command, drawing separating lines or graphics between commands, placing different commands in different speech bubbles, or the like). In another example, different commands can be colored differently (e.g., red text for a first command, green text for a second command, blue text for a third command, etc.). In yet another example, various combinations of the above approaches can be used to differently emphasize different candidate substrings, commands, tasks, or the like (e.g., using differently colored words within a clustered command that is spatially separated from another clustered command). Accordingly, using any of these various visual display approaches, the virtual assistant can show that multiple commands are being recognized in a single stream. Moreover, in other examples, these visual display approaches can be combined with other forms of feedback (e.g., audible, haptic, etc.) discussed herein.

[0255] Process 800 of FIG. 8 can thus be used to handle multi-part commands. As noted above, and as will be understood by one of ordinary skill in the art, various modifications can be made, including removing blocks of process 800, modifying the order of operations, expanding the quantities of various operations, or the like.

[0256] In addition, in any of the various examples discussed herein, various aspects can be personalized for a particular user. As discussed above, user data including contacts, preferences, location, and the like can be used to interpret voice commands. The various processes discussed herein can also be modified in various other ways according to user preferences, contacts, text, usage history, profile data, demographics, or the like. In addition, such preferences and settings can be updated over time based on user interactions (e.g., frequently uttered commands, frequently selected applications, etc.). Gathering and use of user data that is available from various sources can be used 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 can 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.

[0257] 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.

[0258] 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 as 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.

[0259] Despite the foregoing, the present disclosure also contemplates examples 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 not to provide precise location information, but permit the transfer of location zone information.

[0260] Therefore, although the present disclosure broadly covers use of personal information data to implement one or more various disclosed examples, the present disclosure also contemplates that the various examples can also be implemented without the need for accessing such personal information data. That is, the various examples 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-personal information available to the content delivery services, or publicly available information.

[0261] In accordance with some examples, FIG. 15 shows a functional block diagram of an electronic device 1500 configured in accordance with the principles of the various described examples. The functional blocks of the device can be 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. 15 can be combined or separated into sub-blocks to implement the principles of the various described examples. Therefore, the description herein optionally supports any possible combination or separation or further definition of the functional blocks described herein.

[0262] As shown in FIG. 15, electronic device 1500 can include an input unit 1502 configured to receive information (e.g., a microphone for capturing user speech, a device for receiving user speech through a network, etc.). Electronic device 1500 can further include an output unit 1504 configured to output information (e.g., a speaker for playing sounds, a display for displaying information, a device for transmitting information through a network, etc.). Electronic device 1500 can further include processing unit 1506 coupled to input unit 1502 and output unit 1504. In some examples, processing unit 1506 can include a speech input receiving unit 1508, text string generating unit 1510, text string parsing unit 1512, probability determining unit 1514, intent determining unit 1516, process execution unit 1518, and acknowledgment providing unit 1520.

[0263] Processing unit 1506 can be configured to receive speech input from a user (e.g., through input unit 1502 using speech input receiving unit 1508), wherein the speech input comprises a single utterance having one or more actionable commands. Processing unit 1506 can be further configured to generate a text string (e.g., using text string generating unit 1510) based on the speech input using a speech transcription process. Processing unit 1506 can be further configured to parse the text string (e.g., using text string parsing unit 1512) into at least a first candidate substring and a second candidate substring. Processing unit 1506 can be further configured to determine (e.g., using probability determining unit 1514) a first probability that the first candidate substring corresponds to a first actionable command and a second probability that the second candidate substring corresponds to a second actionable command. Processing unit 1506 can be further configured to, in response to the first probability and the second probability exceeding a threshold, determine (e.g., using intent determining unit 1516) a first intent associated with the first candidate substring and a second intent associated with the second candidate substring. Processing unit 1506 can be further configured to execute (e.g., using process execution unit 1518) a first process associated with the first intent and a second process associated with the second intent. Processing unit 1506 can be further configured to provide to the user (e.g., through output unit 1504 using acknowledgment providing unit 1520) an acknowledgment associated with the first intent and the second intent.

[0264] In some examples, parsing the text string (e.g., using text string parsing unit 1512) into at least the first candidate substring and the second candidate substring comprises identifying a first keyword in the text string that corresponds to a first domain to determine the first candidate substring, and identifying a second keyword in the text string that corresponds to a second domain to determine the second candidate substring. In other examples, parsing the text string (e.g., using text string parsing unit 1512) into at least the first candidate substring and the second candidate substring comprises identifying a first imperative verb in the text string to determine the first candidate substring, and identifying a second imperative verb in the text string to determine the second candidate substring.

[0265] In some examples, determining (e.g., using probability determining unit 1514) the first probability that the first candidate substring corresponds to the first actionable command and the second probability that the second candidate substring corresponds to the second actionable command comprises determining a first semantic coherence of the first candidate substring and a second semantic coherence of the second candidate substring, and determining the first probability and the second probability based on the first semantic coherence and the second semantic coherence. In other examples, determining (e.g., using probability determining unit 1514) the first probability that the first candidate substring corresponds to the first actionable command and the second probability that the second candidate substring corresponds to the second actionable command comprises comparing the first candidate substring and the second candidate substring to one or more user request templates, and determining the first probability and the second probability based on the comparison. In still other examples, determining (e.g., using probability determining unit 1514) the first probability that the first candidate substring corresponds to the first actionable command and the second probability that the second candidate substring corresponds to the second actionable command comprises submitting the first candidate substring and the second candidate substring to at least a first service and a second service, receiving a first likelihood that the first service can resolve the first actionable command and a second likelihood that the second service can resolve the second actionable command, and determining the first probability and the second probability based on the first likelihood and the second likelihood.

[0266] In some examples, determining (e.g., using intent determining unit 1516) the first intent associated with the first candidate substring and the second intent associated with the second candidate substring comprises determining the second intent based on at least one word in the first candidate substring. In other examples, determining (e.g., using intent determining unit 1516) the first intent associated with the first candidate substring and the second intent associated with the second candidate substring comprises determining the first intent or the second intent based on information displayed on a display associated with the electronic device. In some examples, the information comprises a list, and determining (e.g., using intent determining unit 1516) the first intent associated with the first candidate substring and the second intent associated with the second candidate substring comprises determining the first intent or the second intent based on an ordinal descriptor in the first candidate substring or the second candidate substring, wherein the ordinal descriptor is associated with one or more items in the list. In other examples, the information comprises one or more notifications. In still other examples, the information comprises one or more emails. In some examples, determining (e.g., using intent determining unit 1516) the first intent associated with the first candidate substring and the second intent associated with the second candidate substring comprises determining one or more potential user requests based on the information displayed on the display, and determining the first intent or the second intent based on the one or more potential user requests.

[0267] In some examples, the acknowledgment (e.g., from acknowledgment providing unit 1520) comprises an audible confirmation. In other examples, the acknowledgment comprises haptic feedback. In some examples, providing to the user (e.g., through output unit 1504 using acknowledgment providing unit 1520) the acknowledgment associated with the first intent and the second intent comprises providing a first task associated with the first intent and a second task associated with the second intent. In one example, providing the first task associated with the first intent and the second task associated with the second intent comprises displaying the first task and the second task.

[0268] In some examples, processing unit 1506 can be further configured to, in response to completing the first process, provide (e.g., using acknowledgment providing unit 1520) a first indicator associated with the first task, and in response to completing the second process, provide a second indicator associated with the second task. In one example, providing the first indicator associated with the first task comprises displaying the first indicator, and providing the second indicator associated with the second task comprises displaying the second indicator. In other examples, processing unit 1506 can be further configured to, before completing the first process, provide (e.g., using acknowledgment providing unit 1520) a first processing status indicator associated with the first task, and before completing the second process, provide a second processing status indicator associated with the second task. In one example, providing the first processing status indicator associated with the first task comprises displaying the first processing status indicator, and providing the second processing status indicator associated with the second task comprises displaying the second processing status indicator. In some examples, the first indicator and the second indicator comprise a checkmark. In other examples, the first processing status indicator and the second processing status indicator comprise one or more of an hourglass, an animation, or a status bar.

[0269] In other examples, providing to the user (e.g., through output unit 1504 using acknowledgment providing unit 1520) the acknowledgment associated with the first intent and the second intent comprises displaying the first candidate substring using a first emphasis and displaying the second candidate substring using a second emphasis that is different than the first emphasis. In some examples, the first emphasis and the second emphasis comprise one or more of bold text, italic text, underlined text, circled text, outlined text, colored text, and clustered text.

[0270] The operations described above with reference to FIG. 8 are, optionally, implemented by components depicted in FIGs. 1A-1B or FIG. 15. For example, receiving operation 802, generating operation 804, parsing operation 806, determining operations 808-812, executing operation 814, and providing operation 816 are, optionally, implemented by event sorter 170, event recognizer 180, and event handler

190. Event monitor 171 in event sorter 170 detects a contact on touch-sensitive display 112, and event dispatcher module 174 delivers the event information to application 136-1. A respective event recognizer 180 of application 136-1 compares the event information to respective event definitions 186, and determines whether a first contact at a first location on the touch-sensitive surface (or whether rotation of the device) corresponds to a predefined event or sub-event, such as selection of an object on a user interface, or rotation of the device from one orientation to another. When a respective predefined event or sub-event is detected, event recognizer 180 activates an event handler 190 associated with the detection of the event or sub-event. Event handler 190 optionally uses or calls data updater 176 or object updater 177 to update the application internal state 192. In some embodiments, event handler 190 accesses a respective GUI updater 178 to update what is displayed by the application. Similarly, it would be clear to a person having ordinary skill in the art how other processes can be implemented based on the components depicted in FIGs. 1A-1B.

[0271] 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.

[0272] 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.

WHAT IS CLAIMED IS:

1. A method for processing a multi-part voice command, the method comprising: at an electronic device:

receiving speech input from a user, wherein the speech input comprises a single utterance having one or more actionable commands;

generating a text string based on the speech input using a speech transcription process;

parsing the text string into at least a first candidate substring and a second candidate substring;

determining a first probability that the first candidate substring corresponds to a first actionable command and a second probability that the second candidate substring corresponds to a second actionable command;

in response to the first probability and the second probability exceeding a threshold, determining a first intent associated with the first candidate substring and a second intent associated with the second candidate substring;

executing a first process associated with the first intent and a second process associated with the second intent; and

providing to the user an acknowledgment associated with the first intent and the second intent.

2. The method of claim 1, wherein parsing the text string into at least the first candidate substring and the second candidate substring comprises:

identifying a first keyword in the text string that corresponds to a first domain to determine the first candidate substring; and

identifying a second keyword in the text string that corresponds to a second domain to determine the second candidate substring.

3. The method of any of claims 1-2, wherein parsing the text string into at least the first candidate substring and the second candidate substring comprises:

identifying a first imperative verb in the text string to determine the first candidate substring; and

identifying a second imperative verb in the text string to determine the second candidate substring.

4. The method of any of claims 1-3, wherein determining the first probability that the first candidate substring corresponds to the first actionable command and the second probability that the second candidate substring corresponds to the second actionable command comprises:

determining a first semantic coherence of the first candidate substring and a second semantic coherence of the second candidate substring; and

determining the first probability and the second probability based on the first semantic coherence and the second semantic coherence.

5. The method of any of claims 1-4, wherein determining the first probability that the first candidate substring corresponds to the first actionable command and the second probability that the second candidate substring corresponds to the second actionable command comprises:

comparing the first candidate substring and the second candidate substring to one or more user request templates; and

determining the first probability and the second probability based on the comparison.

6. The method of any of claims 1-5, wherein determining the first probability that the first candidate substring corresponds to the first actionable command and the second probability that the second candidate substring corresponds to the second actionable command comprises:

submitting the first candidate substring and the second candidate substring to at least a first service and a second service;

receiving a first likelihood that the first service can resolve the first actionable command and a second likelihood that the second service can resolve the second actionable command; and

determining the first probability and the second probability based on the first likelihood and the second likelihood.

7. The method of any of claims 1-6, wherein determining the first intent associated with the first candidate substring and the second intent associated with the second candidate substring comprises:

determining the second intent based on at least one word in the first candidate substring.

8. The method of any of claims 1-7, wherein determining the first intent associated with the first candidate substring and the second intent associated with the second candidate substring comprises:

determining the first intent or the second intent based on information displayed on a display associated with the electronic device.

9. The method of claim 8, wherein the information comprises a list; and

wherein determining the first intent associated with the first candidate substring and the second intent associated with the second candidate substring comprises:

determining the first intent or the second intent based on an ordinal descriptor in the first candidate substring or the second candidate substring, wherein the ordinal descriptor is associated with one or more items in the list.

10. The method of any of claims 8-9, wherein the information comprises one or more notifications.

11. The method of any of claims 8-10, wherein the information comprises one or more emails.

12. The method of any of claims 8-11, wherein determining the first intent associated with the first candidate substring and the second intent associated with the second candidate substring comprises:

determining one or more potential user requests based on the information displayed on the display; and

determining the first intent or the second intent based on the one or more potential user requests.

13. The method of any of claims 1-12, wherein the acknowledgment comprises an audible confirmation.

14. The method of any of claims 1-13, wherein the acknowledgment comprises haptic feedback.

15. The method of any of claims 1-14, wherein providing to the user the acknowledgment associated with the first intent and the second intent comprises:

providing a first task associated with the first intent and a second task associated with the second intent.

16. The method of claim 15, wherein providing the first task associated with the first intent and the second task associated with the second intent comprises:

displaying the first task and the second task.

17. The method of any of claims 15-16, further comprising:

in response to completing the first process, providing a first indicator associated with the first task; and

in response to completing the second process, providing a second indicator associated with the second task.

18. The method of claim 17, wherein providing the first indicator associated with the first task comprises displaying the first indicator; and

wherein providing the second indicator associated with the second task comprises displaying the second indicator.

19. The method of any of claims 17-18, further comprising:

before completing the first process, providing a first processing status indicator associated with the first task; and

before completing the second process, providing a second processing status indicator associated with the second task.

20. The method of claim 19, wherein providing the first processing status indicator associated with the first task comprises displaying the first processing status indicator; and

wherein providing the second processing status indicator associated with the second task comprises displaying the second processing status indicator.

21. The method of any of claims 17-20, wherein the first indicator and the second indicator comprise a checkmark.

22. The method of any of claims 19-21, wherein the first processing status indicator and the second processing status indicator comprise one or more of an hourglass, an animation, or a status bar.

23. The method of any of claims 1-14, wherein providing to the user the acknowledgment associated with the first intent and the second intent comprises:

displaying the first candidate substring using a first emphasis and displaying the second candidate substring using a second emphasis that is different than the first emphasis.

24. The method of claim 23, wherein the first emphasis and the second emphasis comprise one or more of bold text, italic text, underlined text, circled text, outlined text, colored text, and clustered text.

25. The method of any one of claims 1-24, wherein:

generating the text string based on the speech input is performed in direct response to receiving the speech input comprising the single utterance.

26. A non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by an electronic device with a touch-sensitive display, cause the device to perform any of the methods of claims 1-25.

27. An electronic device, comprising:

one or more processors;

memory; and

one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions, which when executed by the one or more processors, cause the device to perform any of the methods of claims 1-25.

28. An electronic device, comprising:
means for performing any of the methods of claims 1-25.

29. An electronic device, comprising:
a processing unit configured to perform the method of any of claims 1-25.

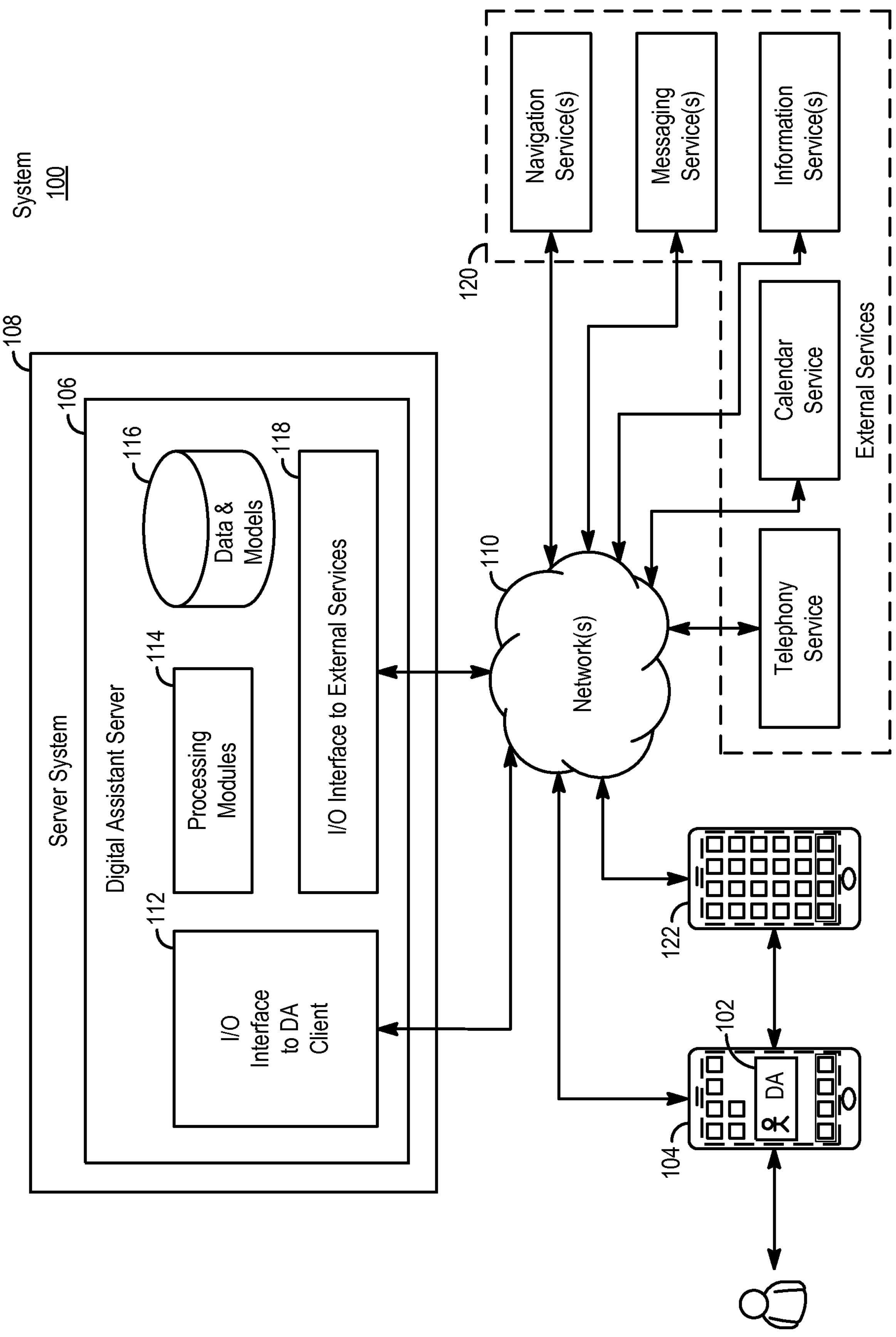

System
100
108
Server System
106
Digital Assistant Server
116
Data & Models
118
I/O Interface to External Services
114
Processing Modules
112
I/O Interface to DA Client
110
Network(s)
120
Navigation Service(s)
Messaging Service(s)
Information Service(s)
Calendar Service
Telephony Service
External Services
122
102
DA
104

FIG. 1

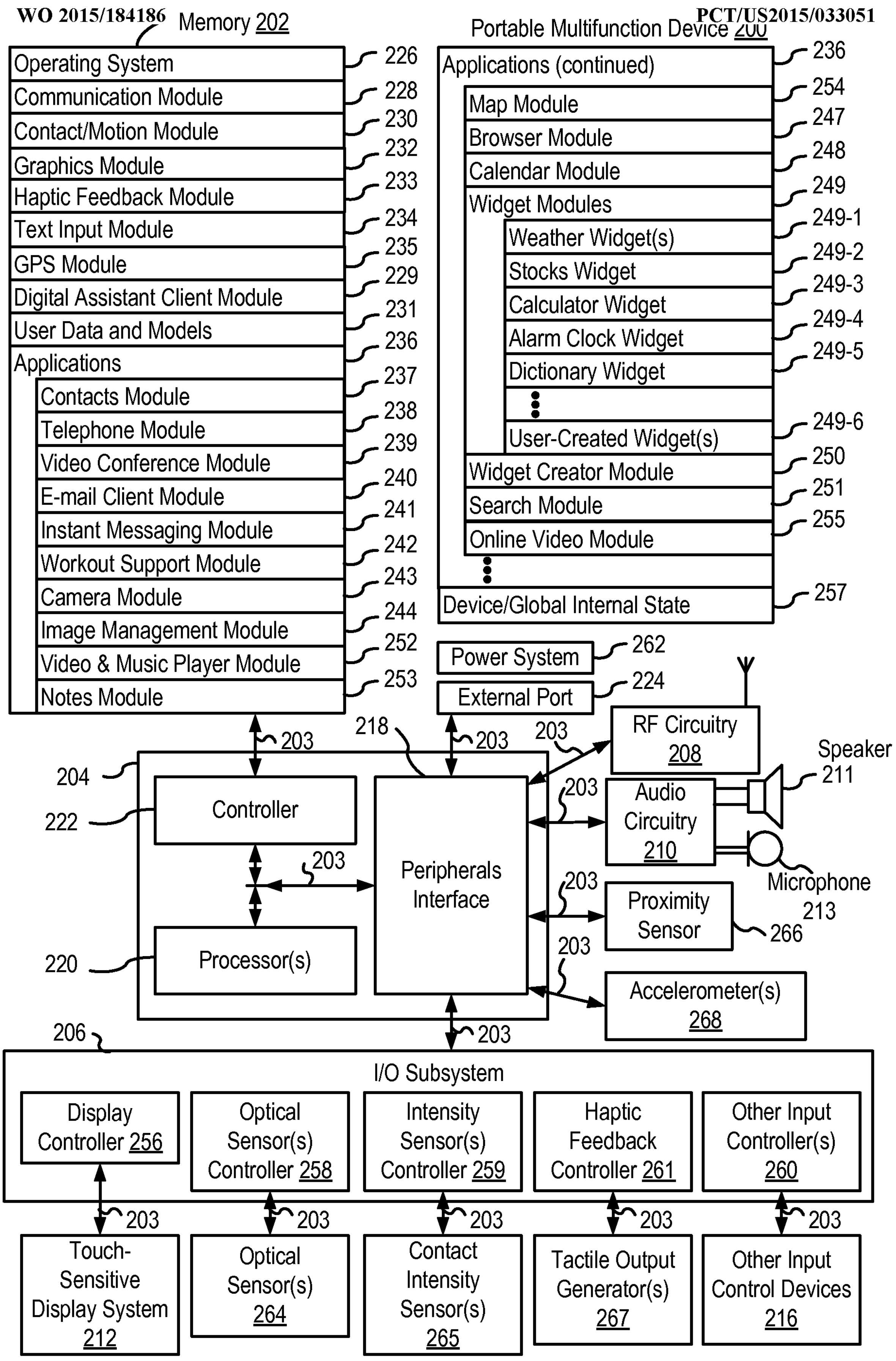

Memory 202
Operating System 226
Communication Module 228
Contact/Motion Module 230
Graphics Module 232
Haptic Feedback Module 233
Text Input Module 234
GPS Module 235
Digital Assistant Client Module 229
User Data and Models 231
Applications 236
Contacts Module 237
Telephone Module 238
Video Conference Module 239
E-mail Client Module 240
Instant Messaging Module 241
Workout Support Module 242
Camera Module 243
Image Management Module 244
Video & Music Player Module 252
Notes Module 253
Portable Multifunction Device 200
Applications (continued) 236
Map Module 254
Browser Module 247
Calendar Module 248
Widget Modules 249
Weather Widget(s) 249-1
Stocks Widget 249-2
Calculator Widget 249-3
Alarm Clock Widget 249-4
Dictionary Widget 249-5
User-Created Widget(s) 249-6
Widget Creator Module 250
Search Module 251
Online Video Module 255
Device/Global Internal State 257
Power System 262
External Port 224
RF Circuitry 208
Speaker 211
Audio Circuitry 210
Microphone 213
Proximity Sensor 266
Accelerometer(s) 268
204
Controller 222
Processor(s) 220
Peripherals Interface 218
203
206
I/O Subsystem
Display Controller 256
Optical Sensor(s) Controller 258
Intensity Sensor(s) Controller 259
Haptic Feedback Controller 261
Other Input Controller(s) 260
Touch-Sensitive Display System 212
Optical Sensor(s) 264
Contact Intensity Sensor(s) 265
Tactile Output Generator(s) 267
Other Input Control Devices 216

FIG. 2A

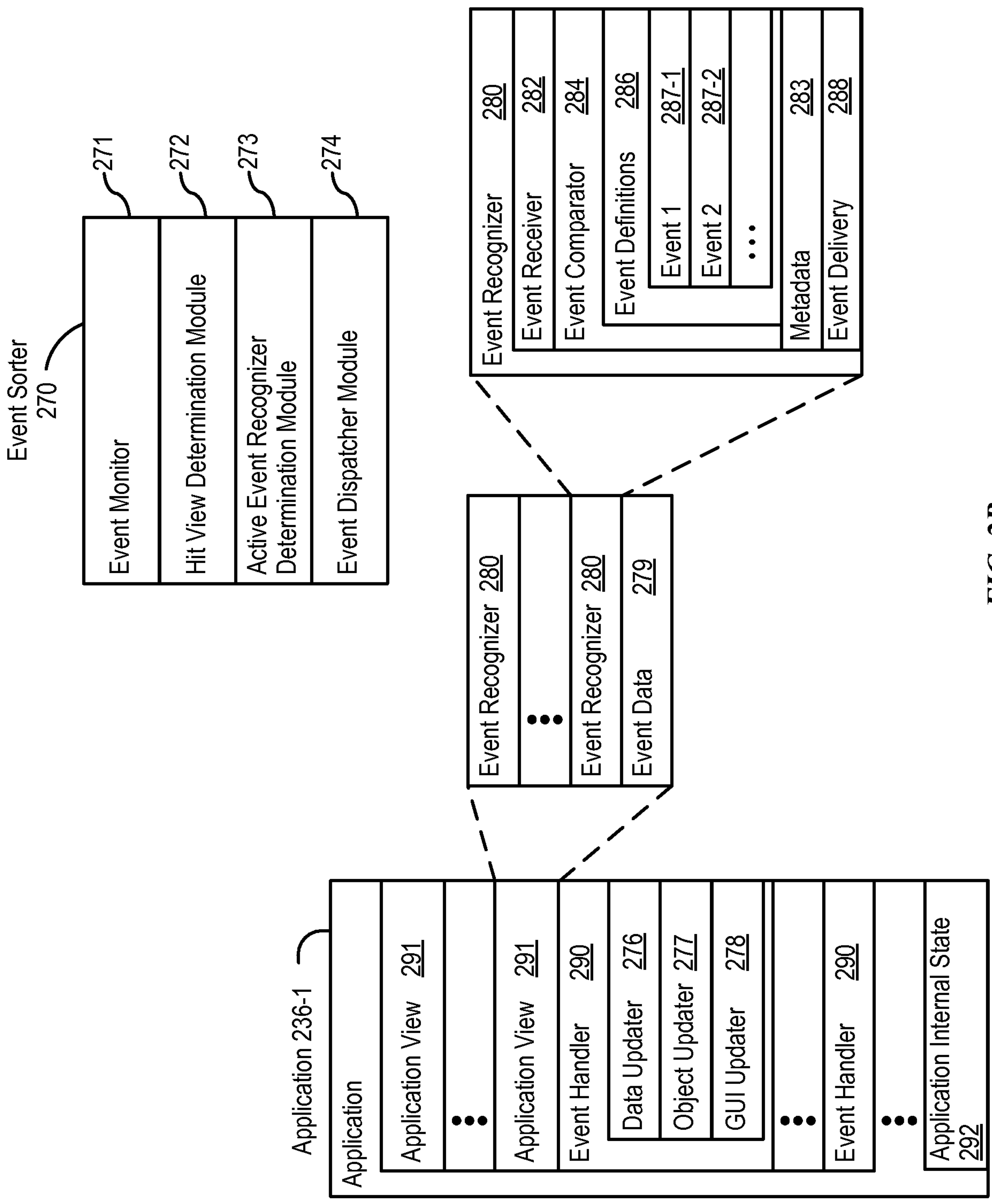

Event Sorter 270
Event Monitor
271
Hit View Determination Module
272
Active Event Recognizer Determination Module
273
Event Dispatcher Module
274
Event Recognizer 280
Event Receiver 282
Event Comparator 284
Event Definitions 286
Event 1 287-1
Event 2 287-2
• • •
Metadata 283
Event Delivery 288
Event Recognizer 280
• • •
Event Recognizer 280
Event Data 279
Application 236-1
Application
Application View 291
• • •
Application View 291
Event Handler 290
Data Updater 276
Object Updater 277
GUI Updater 278
• • •
Event Handler 290
• • •
Application Internal State 292

FIG. 2B

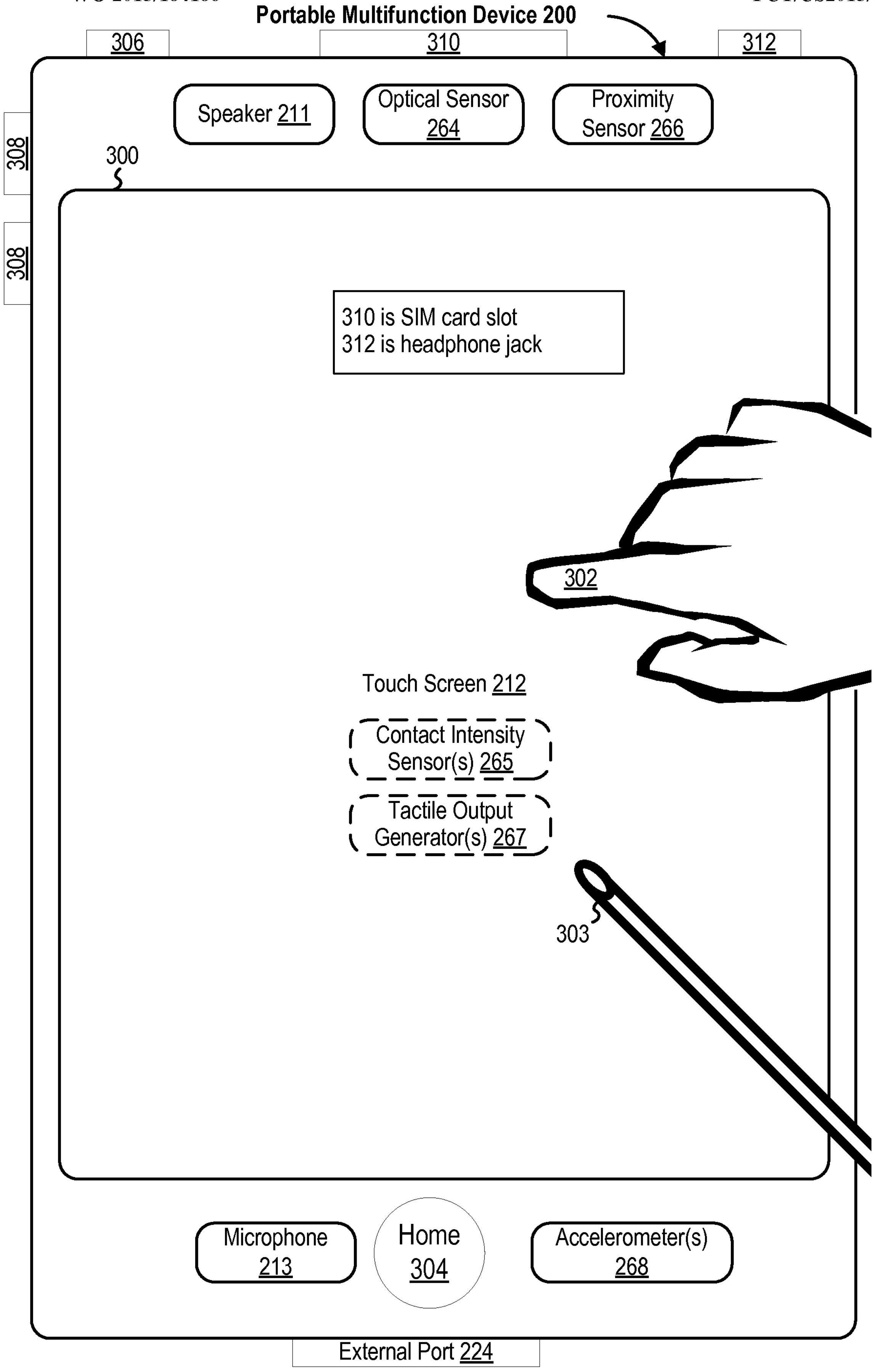
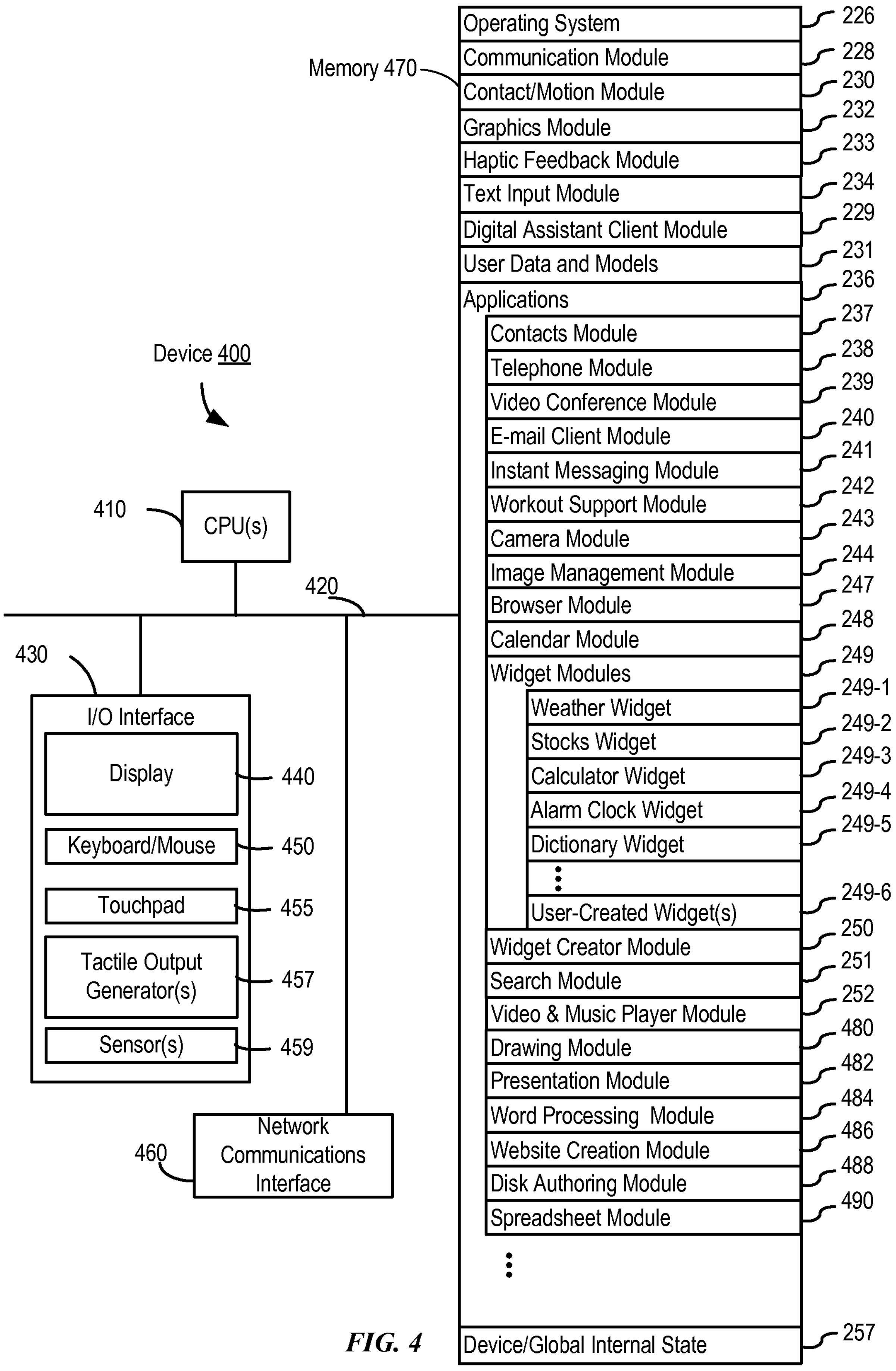

Portable Multifunction Device 200
306
310
312
308
308
Speaker 211
Optical Sensor 264
Proximity Sensor 266
300
310 is SIM card slot
312 is headphone jack
302
Touch Screen 212
Contact Intensity Sensor(s) 265
Tactile Output Generator(s) 267
303
Microphone 213
Home 304
Accelerometer(s) 268
External Port 224

FIG. 3



Device 400
410
CPU(s)
420
430
I/O Interface
Display
440
Keyboard/Mouse
450
Touchpad
455
Tactile Output Generator(s)
457
Sensor(s)
459
460
Network Communications Interface
Memory 470
Operating System
226
Communication Module
228
Contact/Motion Module
230
Graphics Module
232
Haptic Feedback Module
233
Text Input Module
234
Digital Assistant Client Module
229
User Data and Models
231
Applications
236
Contacts Module
237
Telephone Module
238
Video Conference Module
239
E-mail Client Module
240
Instant Messaging Module
241
Workout Support Module
242
Camera Module
243
Image Management Module
244
Browser Module
247
Calendar Module
248
Widget Modules
249
Weather Widget
249-1
Stocks Widget
249-2
Calculator Widget
249-3
Alarm Clock Widget
249-4
Dictionary Widget
249-5
User-Created Widget(s)
249-6
Widget Creator Module
250
Search Module
251
Video & Music Player Module
252
Drawing Module
480
Presentation Module
482
Word Processing Module
484
Website Creation Module
486
Disk Authoring Module
488
Spreadsheet Module
490
Device/Global Internal State
257

FIG. 4

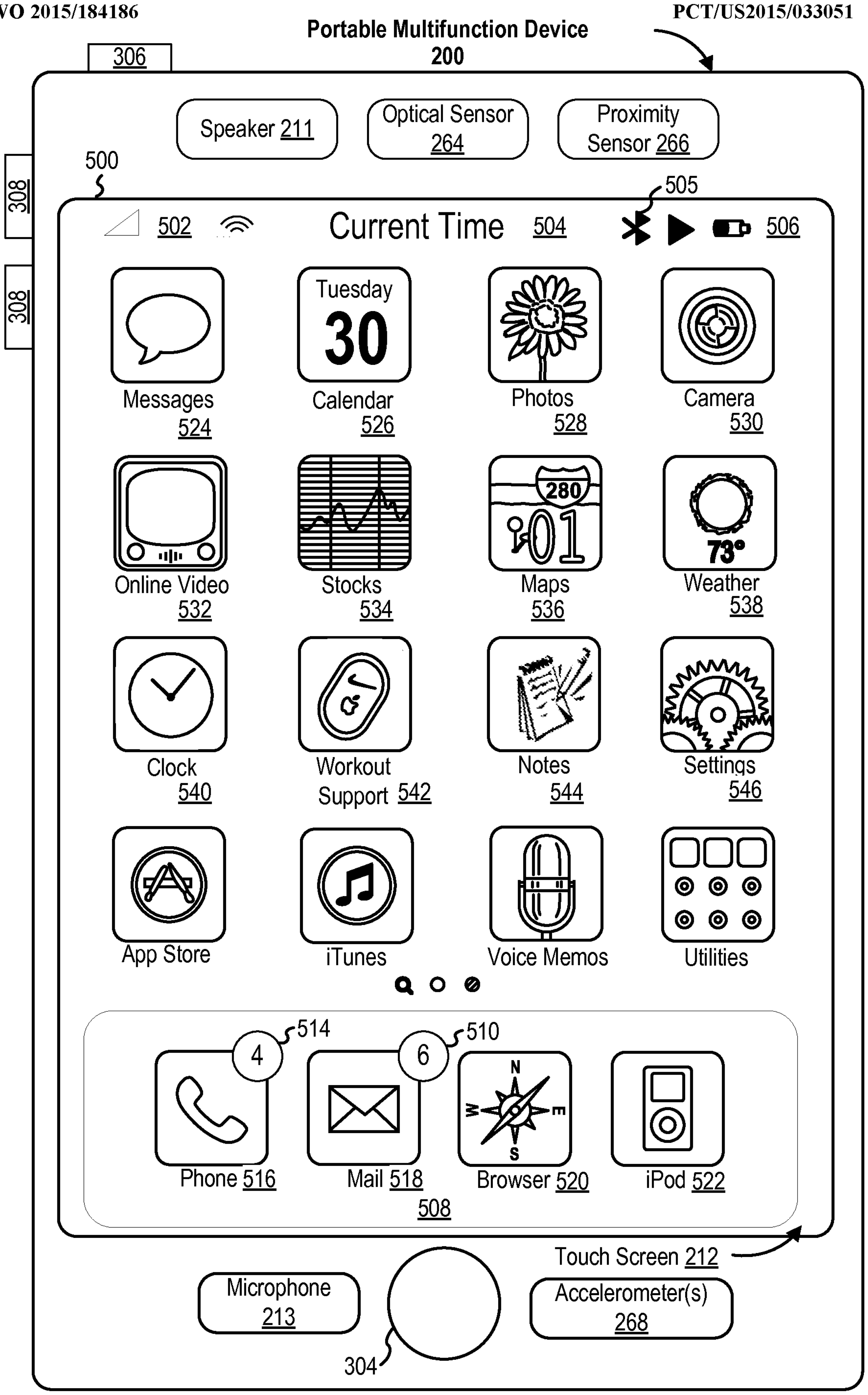

Portable Multifunction Device
200
306
308
308
Speaker 211
Optical Sensor 264
Proximity Sensor 266
500
505
502
Current Time
504
506
Tuesday
30
Messages 524
Calendar 526
Photos 528
Camera 530
Online Video 532
Stocks 534
280
01
Maps 536
73°
Weather 538
Clock 540
Workout Support 542
Notes 544
Settings 546
App Store
iTunes
Voice Memos
Utilities
4
514
6
510
N
W
E
S
Phone 516
Mail 518
Browser 520
iPod 522
508
Touch Screen 212
Microphone 213
Accelerometer(s) 268
304

FIG. 5A

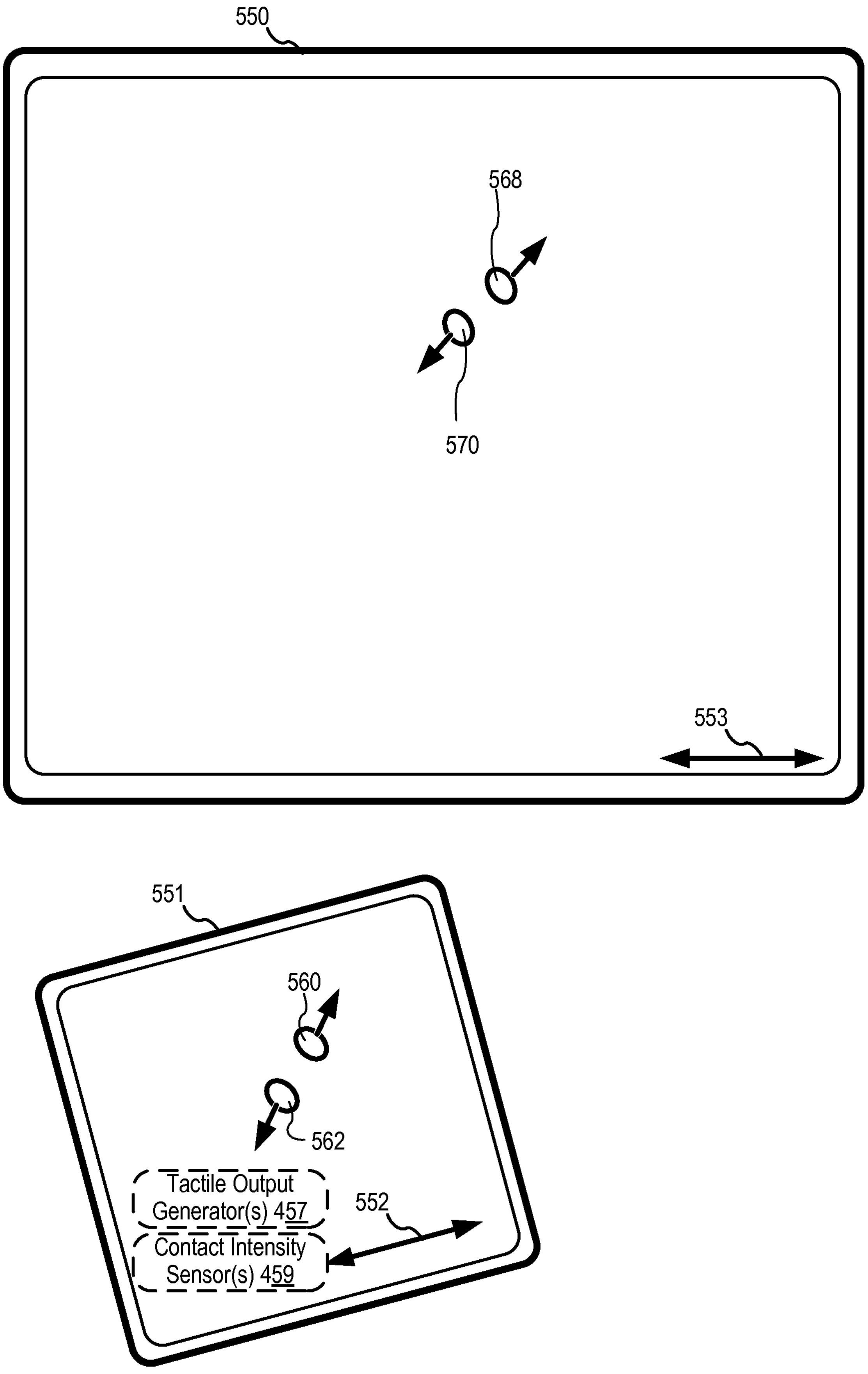

550
568
570
553
551
560
562
Tactile Output Generator(s) 457
Contact Intensity Sensor(s) 459
552

FIG. 5B

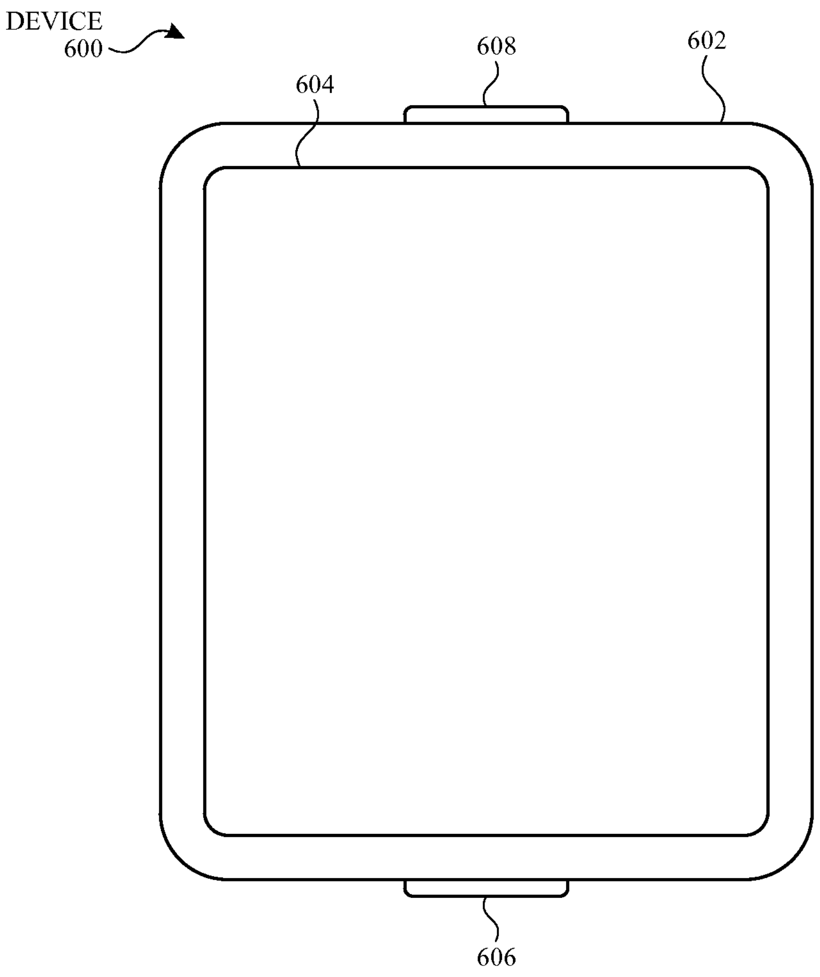
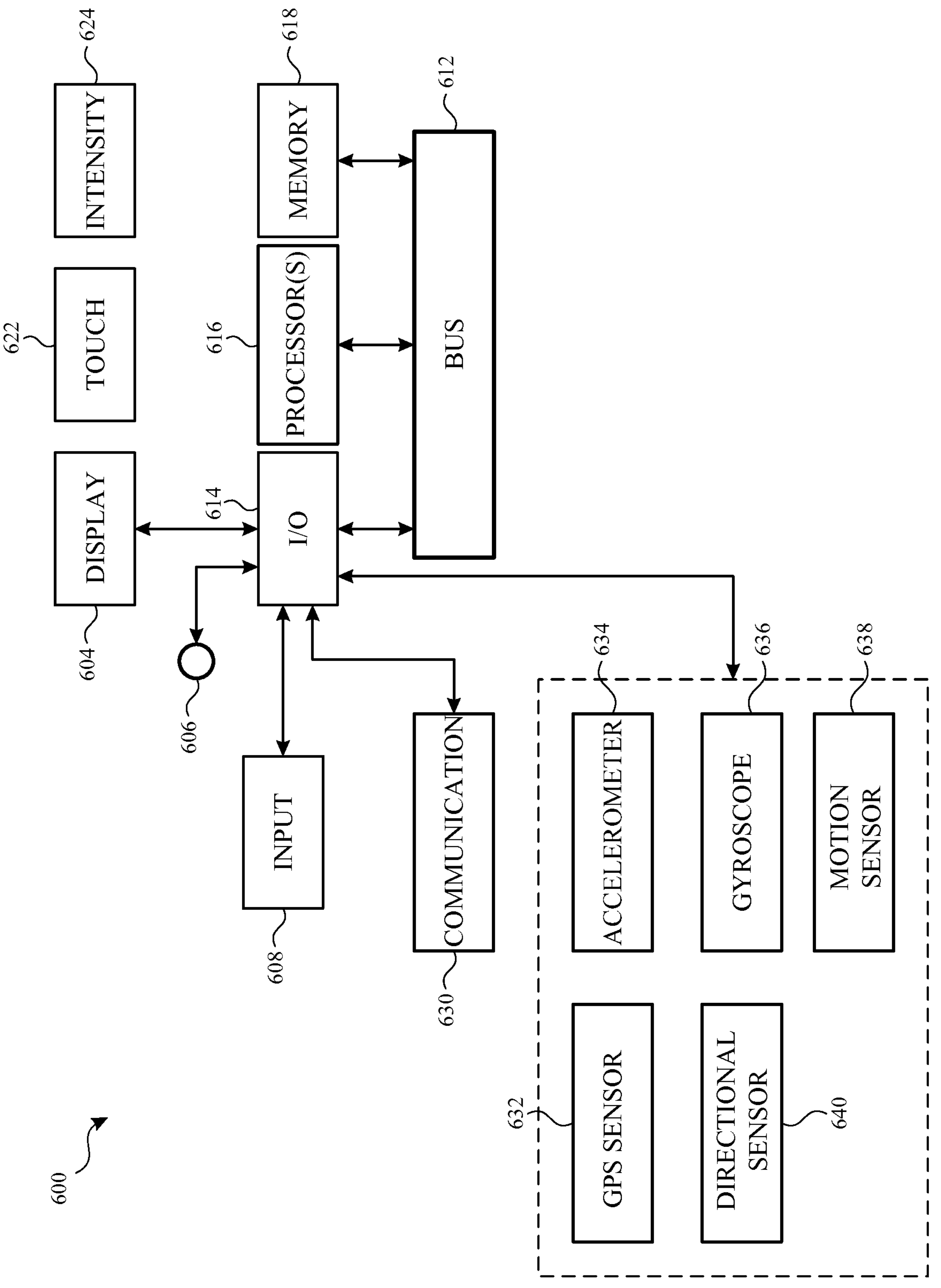


FIG. 6A



600
DISPLAY
604
TOUCH
622
INTENSITY
624
I/O
614
PROCESSOR(S)
616
MEMORY
618
BUS
612
606
INPUT
608
COMMUNICATION
630
GPS SENSOR
632
ACCELEROMETER
634
GYROSCOPE
636
MOTION SENSOR
638
DIRECTIONAL SENSOR
640

FIG. 6B

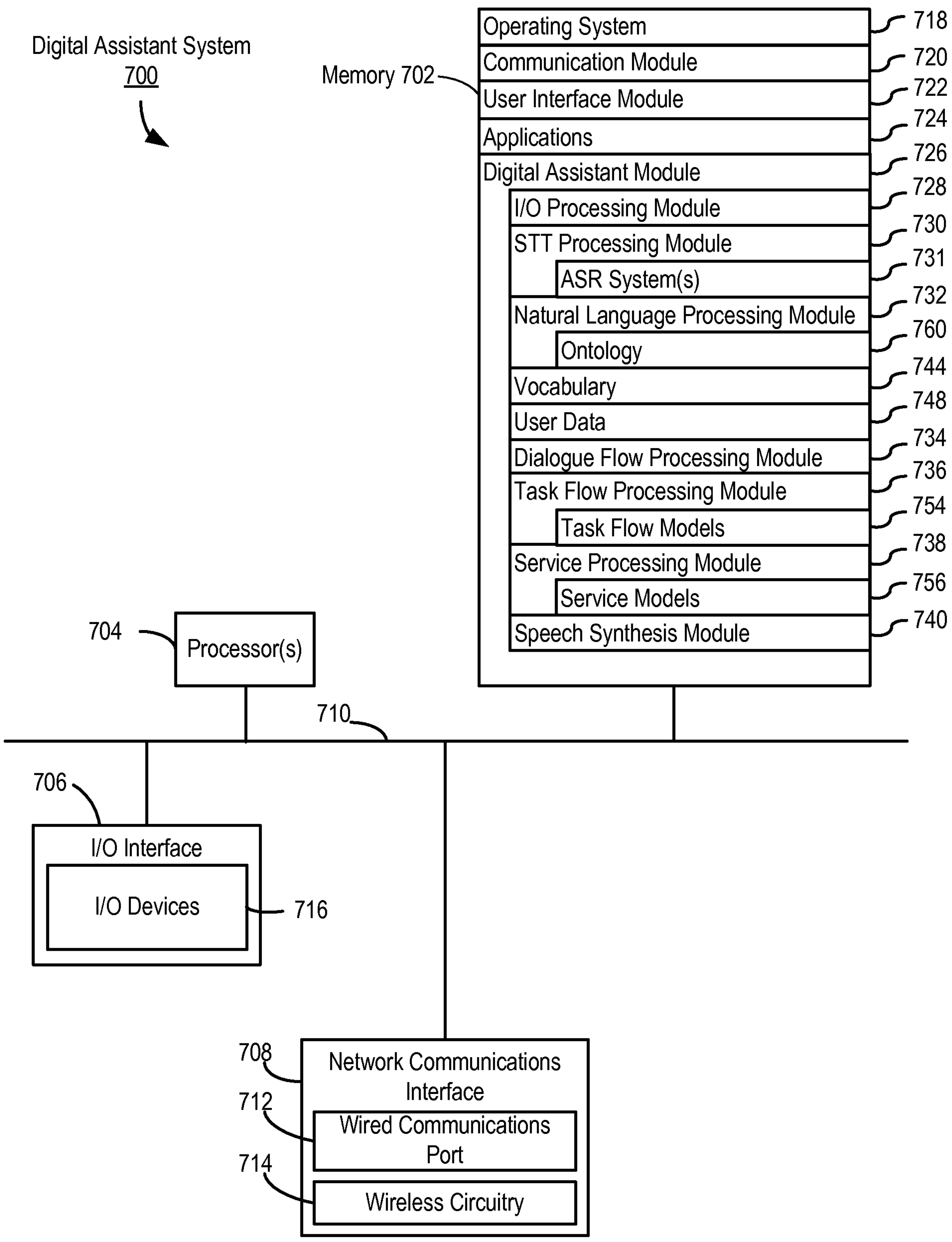

Digital Assistant System
700
Memory 702
Operating System
718
Communication Module
720
User Interface Module
722
Applications
724
Digital Assistant Module
726
I/O Processing Module
728
STT Processing Module
730
ASR System(s)
731
Natural Language Processing Module
732
Ontology
760
Vocabulary
744
User Data
748
Dialogue Flow Processing Module
734
Task Flow Processing Module
736
Task Flow Models
754
Service Processing Module
738
Service Models
756
Speech Synthesis Module
740
704
Processor(s)
710
706
I/O Interface
I/O Devices
716
708
Network Communications Interface
712
Wired Communications Port
714
Wireless Circuitry

FIG. 7A

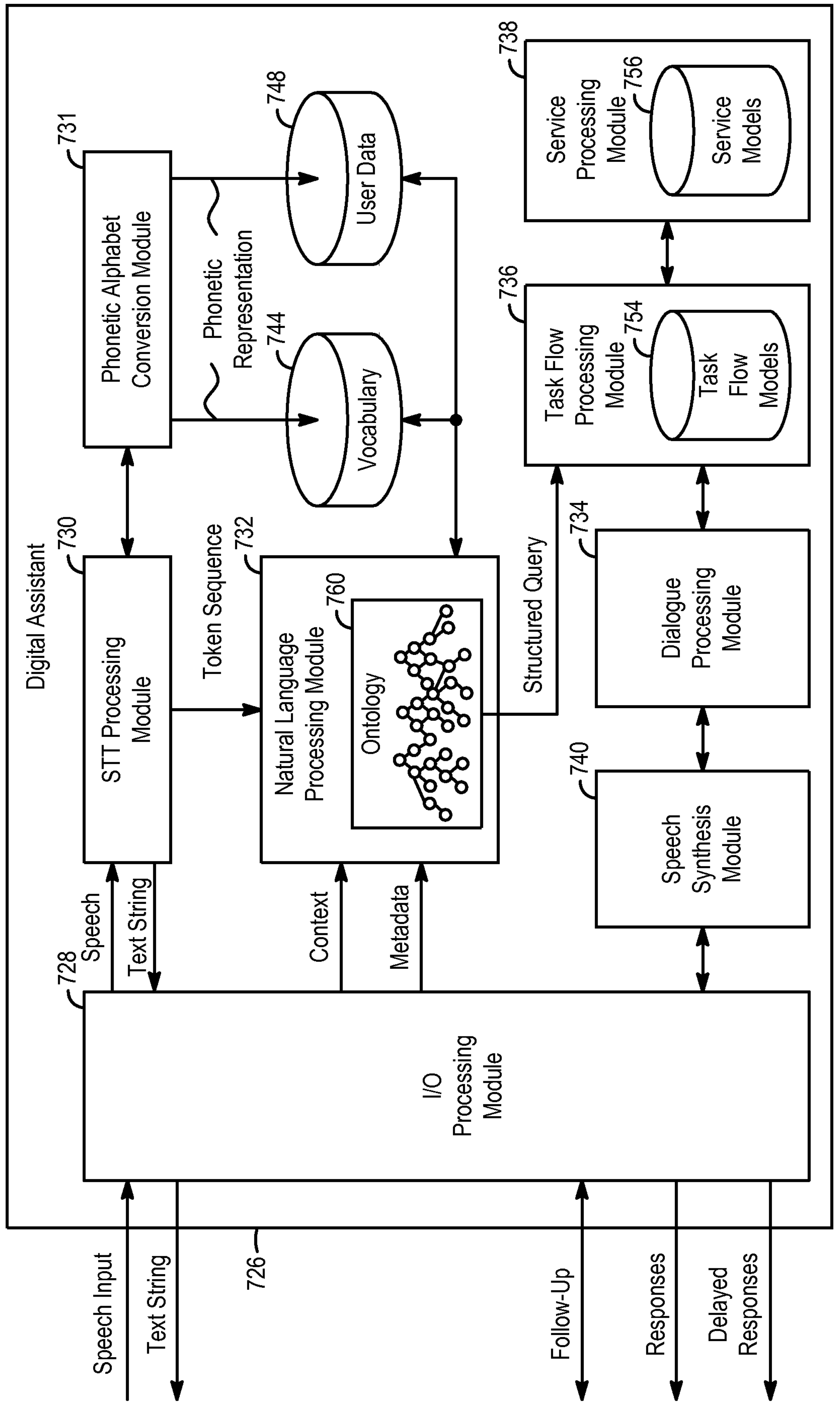

Digital Assistant
726
728
I/O Processing Module
730
STT Processing Module
731
Phonetic Alphabet Conversion Module
Phonetic Representation
744
Vocabulary
748
User Data
732
Natural Language Processing Module
760
Ontology
Token Sequence
Structured Query
734
Dialogue Processing Module
736
Task Flow Processing Module
754
Task Flow Models
738
Service Processing Module
756
Service Models
740
Speech Synthesis Module
Speech
Text String
Context
Metadata
Speech Input
Text String
Follow-Up
Responses
Delayed Responses

FIG. 7B

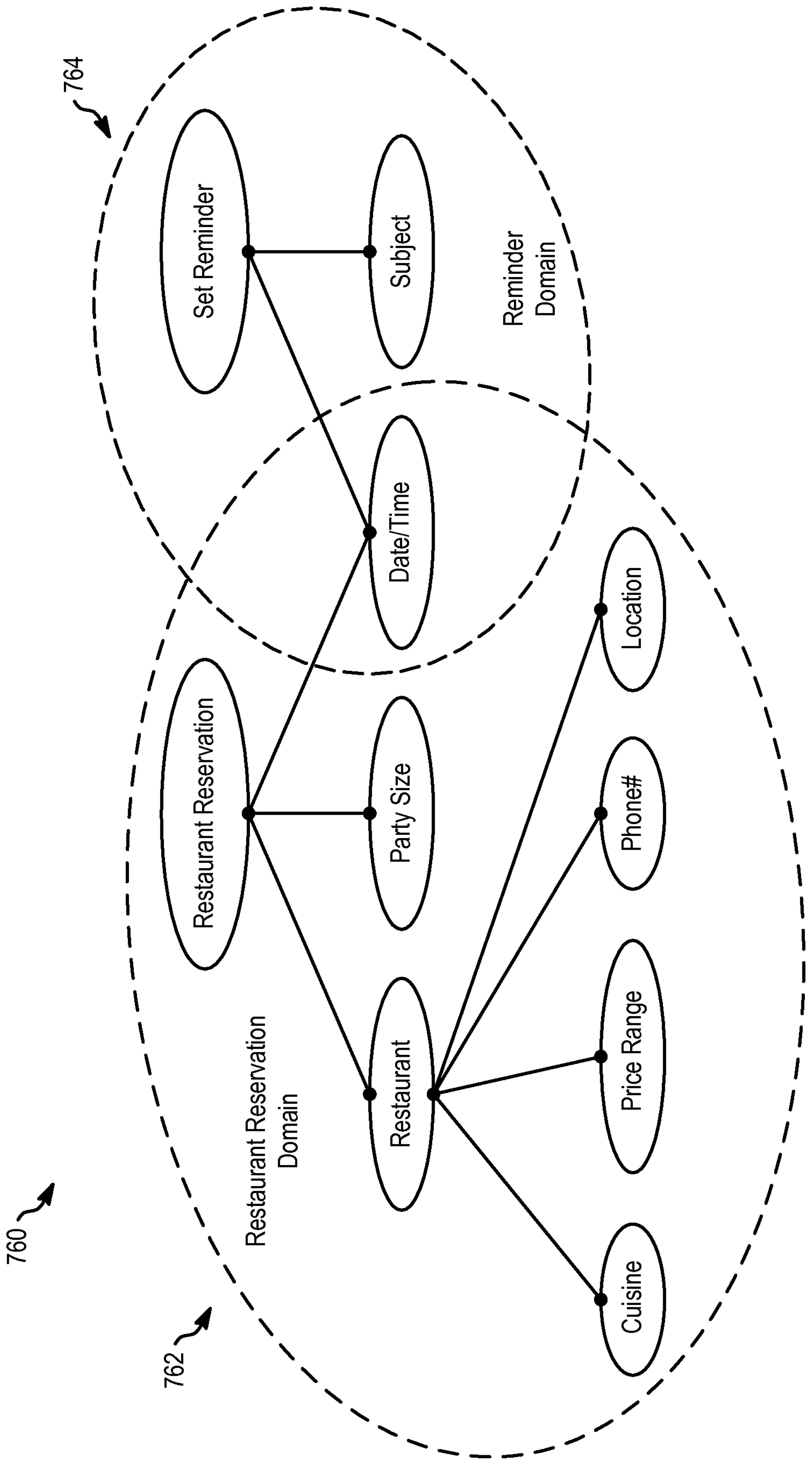

764
Set Reminder
Subject
Reminder Domain
Date/Time
Restaurant Reservation
Party Size
Location
Phone#
Price Range
Restaurant
Cuisine
Restaurant Reservation Domain
760
762

FIG. 7C

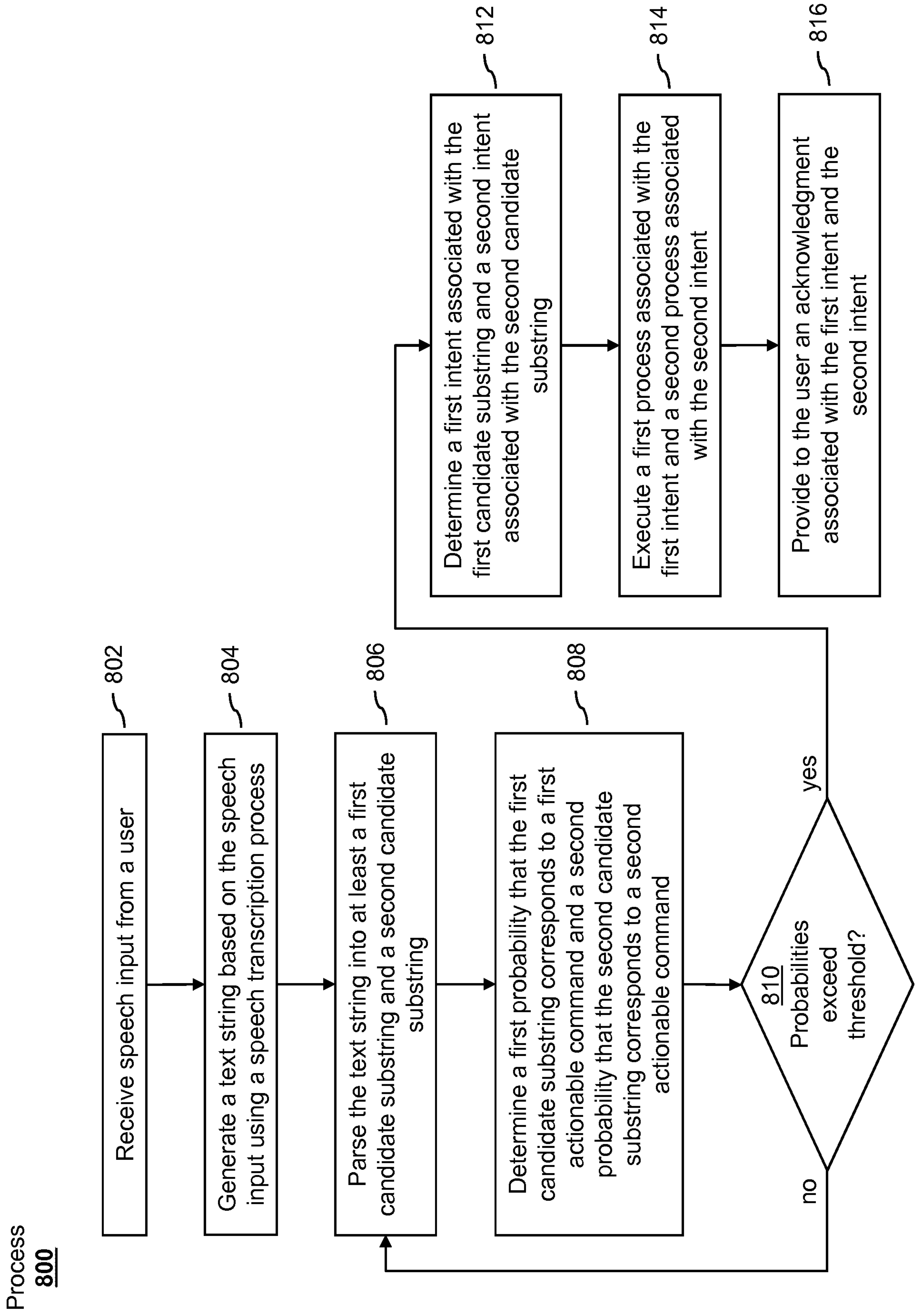

Process
800
Receive speech input from a user
802
Generate a text string based on the speech input using a speech transcription process
804
Parse the text string into at least a first candidate substring and a second candidate substring
806
Determine a first probability that the first candidate substring corresponds to a first actionable command and a second probability that the second candidate substring corresponds to a second actionable command
808
810
Probabilities exceed threshold?
yes
no
Determine a first intent associated with the first candidate substring and a second intent associated with the second candidate substring
812
Execute a first process associated with the first intent and a second process associated with the second intent
814
Provide to the user an acknowledgment associated with the first intent and the second intent
816

FIG. 8

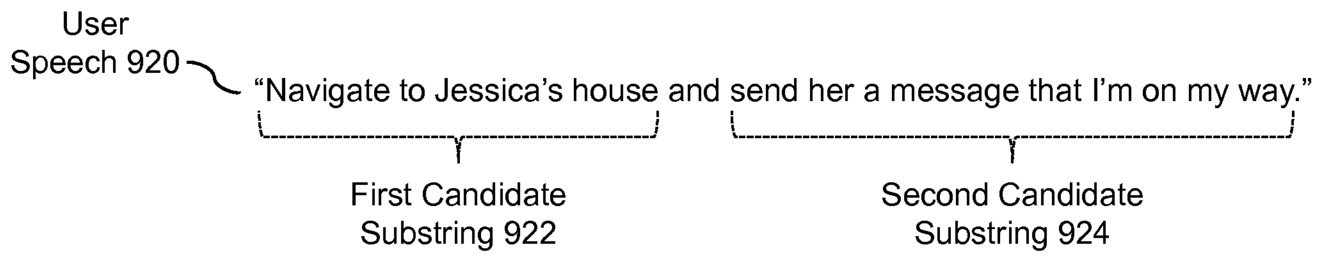

User
Speech 920
"Navigate to Jessica's house and send her a message that I'm on my way."
First Candidate
Substring 922
Second Candidate
Substring 924

FIG. 9

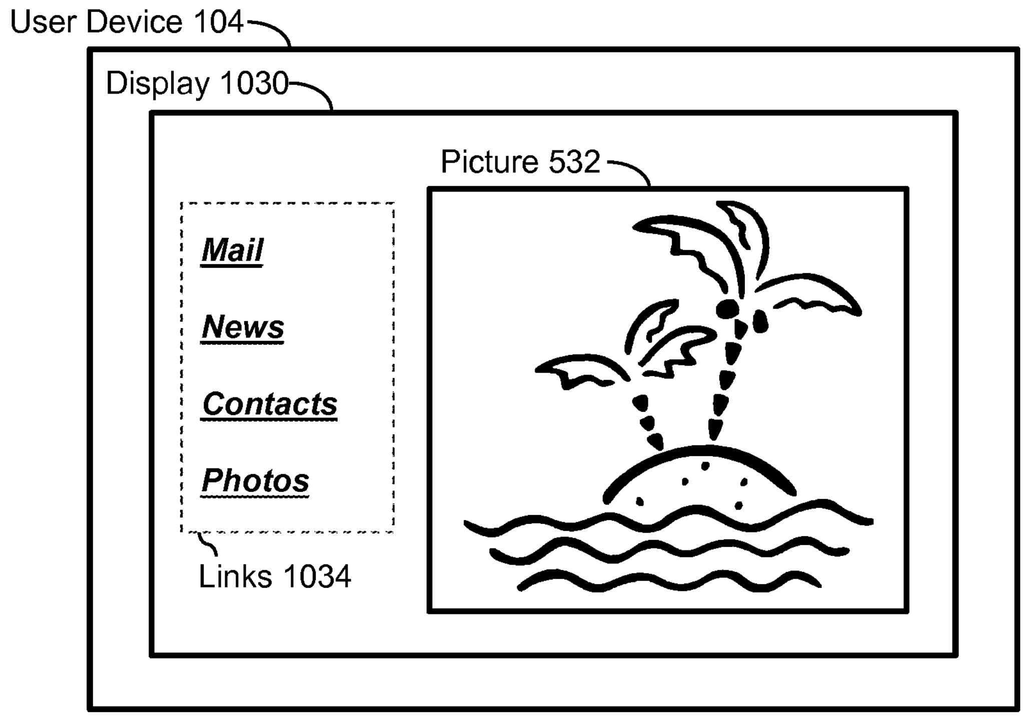
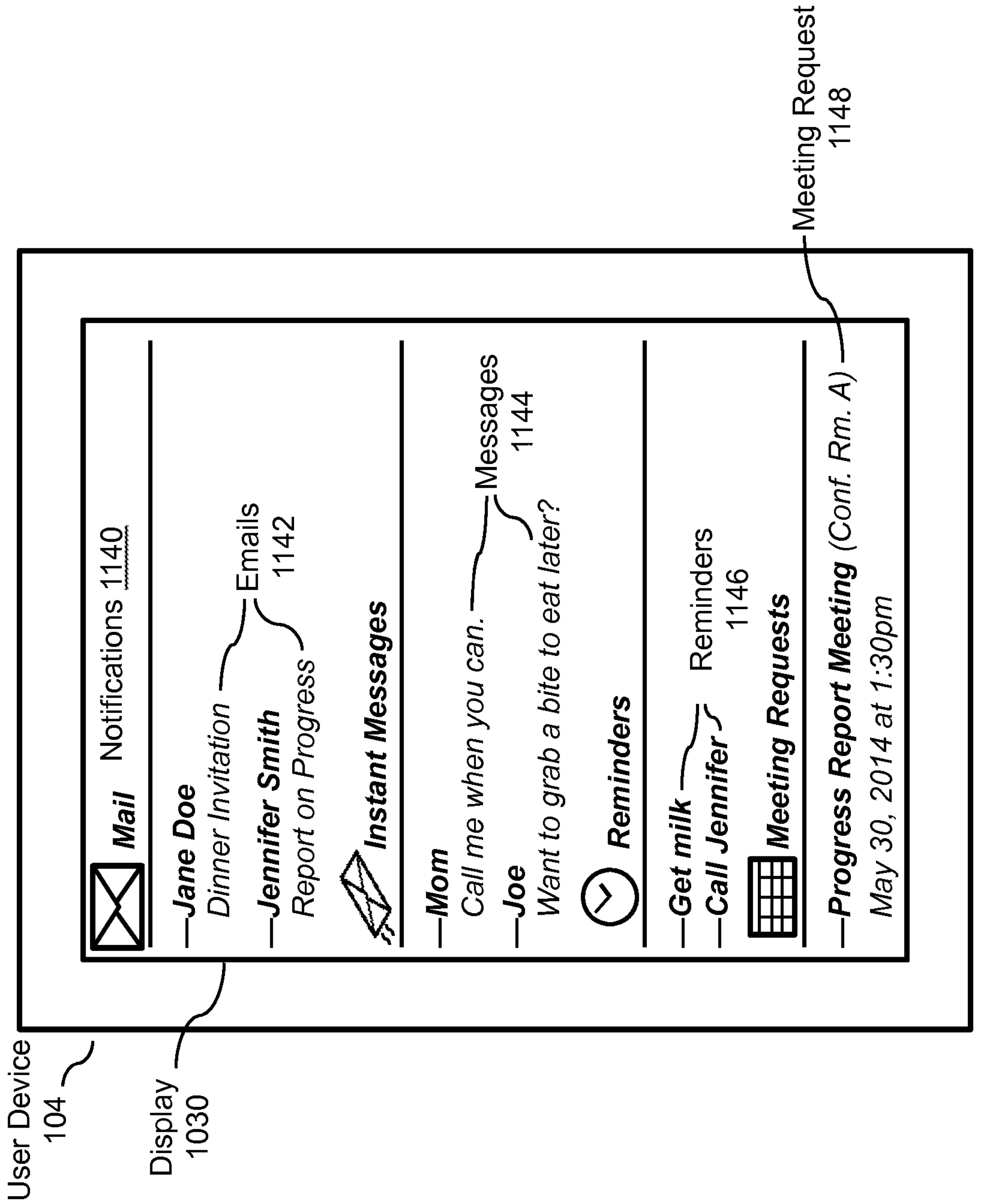

User Device 104
Display 1030
Picture 532
Mail
News
Contacts
Photos
Links 1034

FIG.
10


Notifications 1140

Mail

—Jane Doe
Dinner Invitation

—Jennifer Smith
Report on Progress

Instant Messages

—Mom
Call me when you can.

—Joe
Want to grab a bite to eat later?

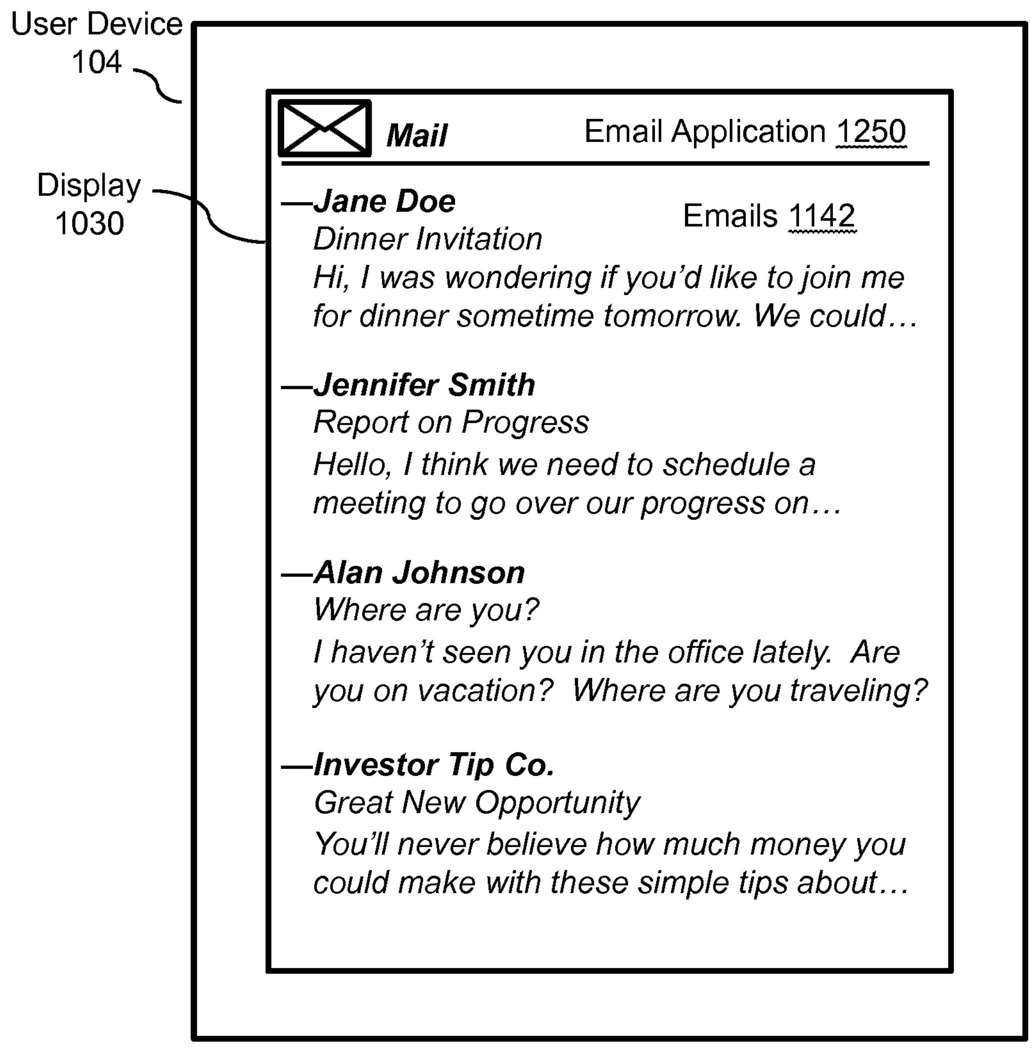
Reminders

—Get milk
—Call Jennifer

Meeting Requests

—Progress Report Meeting (Conf. Rm. A)
May 30, 2014 at 1:30pm

FIG. 11



—**Jane Doe**
Dinner Invitation
Hi, I was wondering if you'd like to join me for dinner sometime tomorrow. We could…

—**Jennifer Smith**
Report on Progress
Hello, I think we need to schedule a meeting to go over our progress on…

—**Alan Johnson**
Where are you?
I haven't seen you in the office lately. Are you on vacation? Where are you traveling?

—**Investor Tip Co.**
Great New Opportunity
You'll never believe how much money you could make with these simple tips about…

FIG. 12

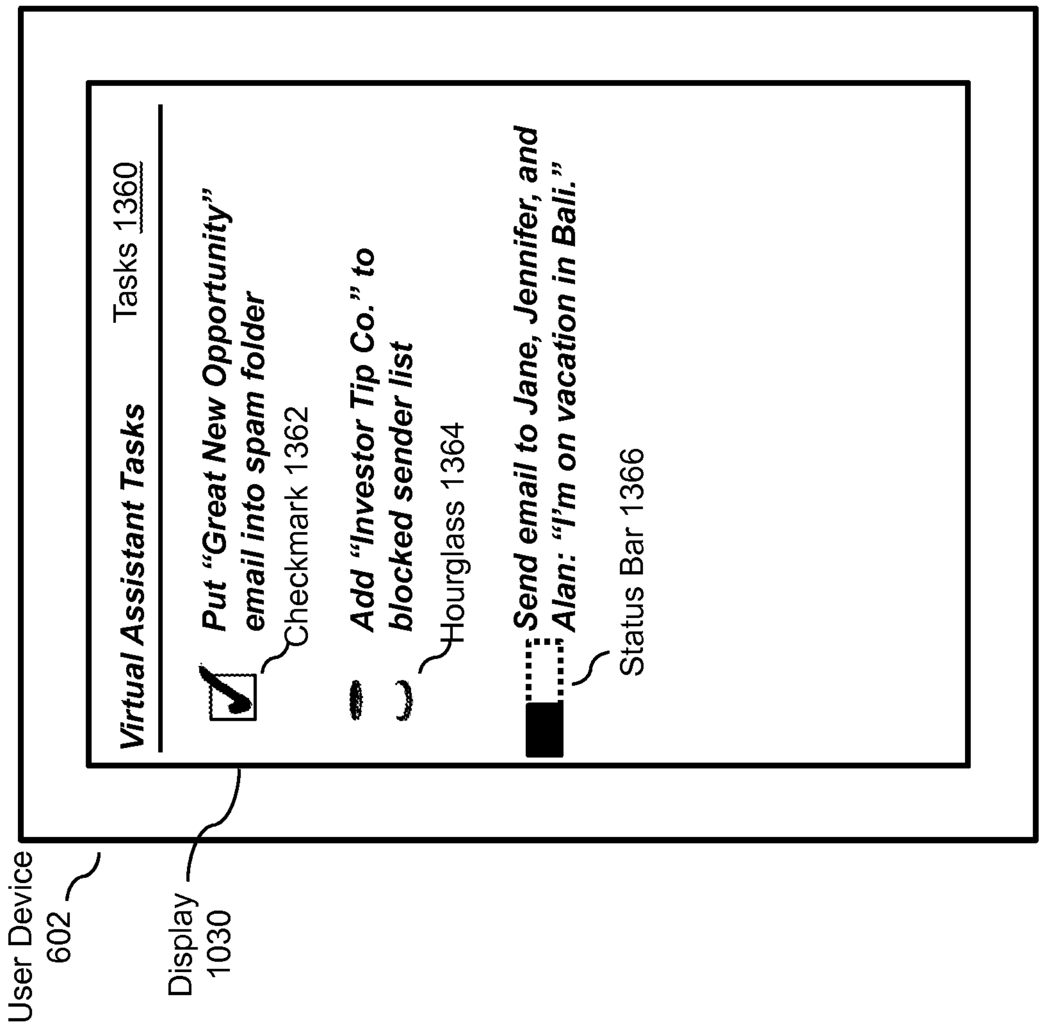

Virtual Assistant Tasks
Tasks 1360
Put "Great New Opportunity" email into spam folder
Checkmark 1362
Add "Investor Tip Co." to blocked sender list
Hourglass 1364
Send email to Jane, Jennifer, and Alan: "I'm on vacation in Bali."
Status Bar 1366
User Device 602
Display 1030

FIG. 13

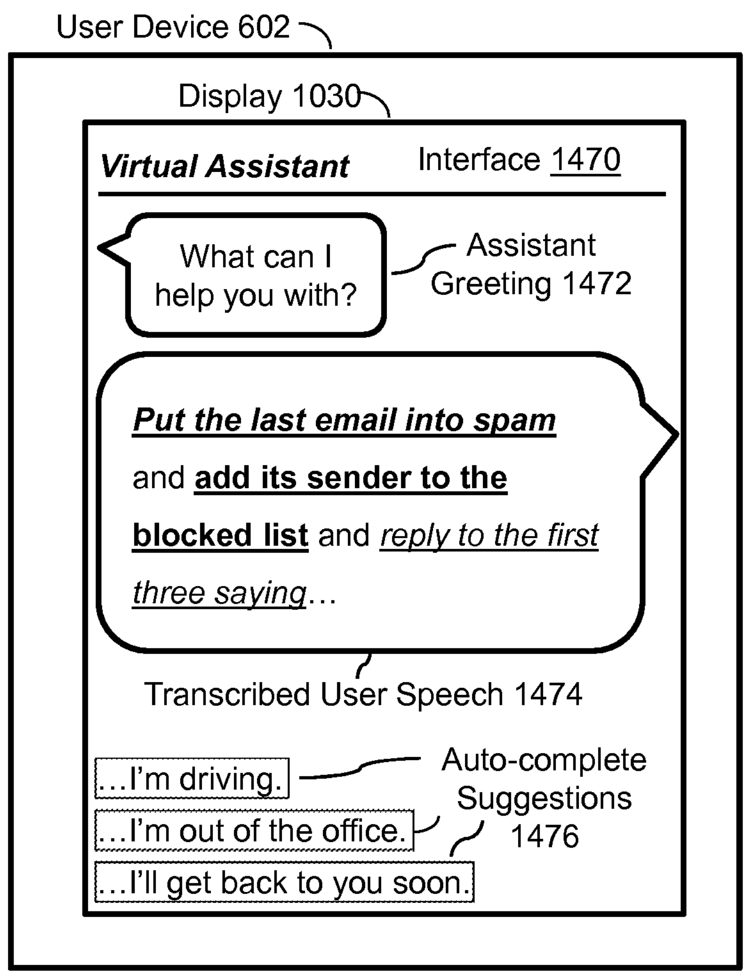

User Device 602
Display 1030
Virtual Assistant
Interface 1470
What can I help you with?
Assistant Greeting 1472
Put the last email into spam and add its sender to the blocked list and reply to the first three saying…
Transcribed User Speech 1474
…I'm driving.
…I'm out of the office.
…I'll get back to you soon.
Auto-complete Suggestions 1476

FIG. 14A

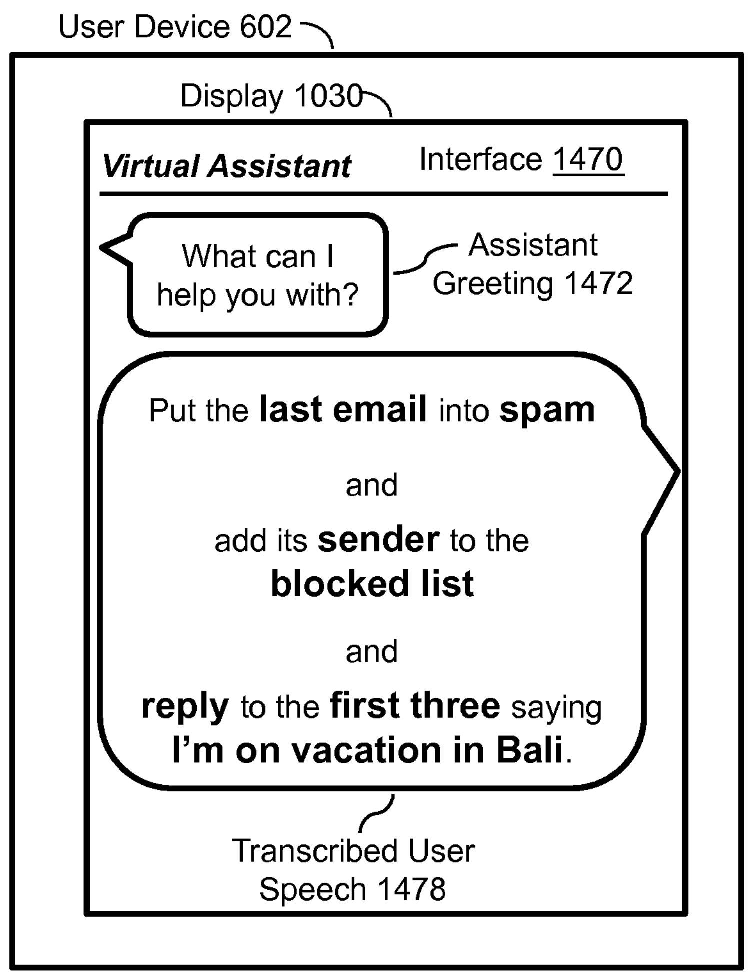

User Device 602
Display 1030
Virtual Assistant
Interface 1470
What can I help you with?
Assistant Greeting 1472
Put the last email into spam and add its sender to the blocked list and reply to the first three saying I'm on vacation in Bali.
Transcribed User Speech 1478

FIG. 14B

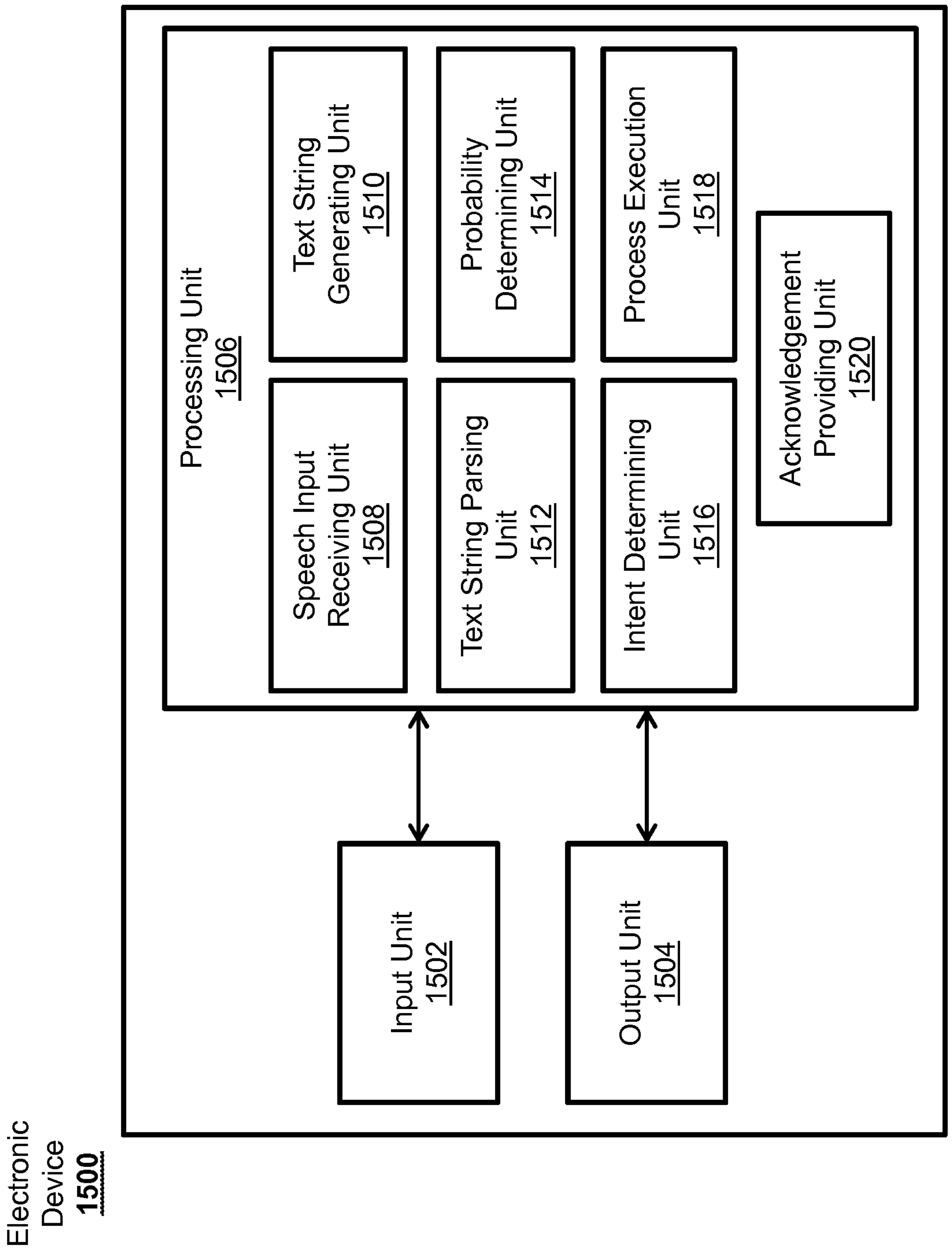

Electronic Device 1500
Input Unit 1502
Output Unit 1504
Processing Unit 1506
Speech Input Receiving Unit 1508
Text String Generating Unit 1510
Text String Parsing Unit 1512
Probability Determining Unit 1514
Intent Determining Unit 1516
Process Execution Unit 1518
Acknowledgement Providing Unit 1520

FIG. 15