



- (51) **International Patent Classification:**
G06F 3/16 (2006.01) *G06F 9/445* (2006.01)
- (21) **International Application Number:**
PCT/US2014/026871
- (22) **International Filing Date:**
13 March 2014 (13.03.2014)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
61/785,853 14 March 2013 (14.03.2013) US
- (71) **Applicant:** APPLE INC. [US/US]; 1 Infinite Loop, Cupertino, CA 95014 (US).
- (72) **Inventor:** FLEIZACH, Christopher, B.; C/o 1 Infinite Loop, Mail Stop Ms 37-2FMW, Cupertino, CA 95014 (US).
- (74) **Agents:** EIDE, Christopher, B. 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,

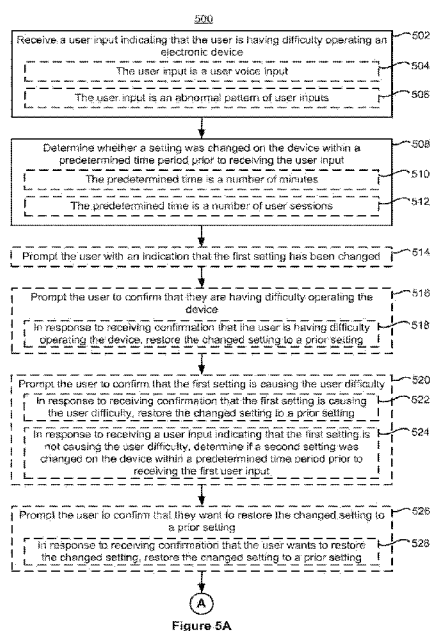
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, 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, 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))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(54) **Title:** VOICE CONTROL TO DIAGNOSE INADVERTENT ACTIVATION OF ACCESSIBILITY FEATURES



(57) **Abstract:** Methods and systems are provided for diagnosing inadvertent activation of user interface settings on an electronic device. The electronic device receives a user input indicating that the user is having difficulty operating the electronic device. The device then determines whether a setting was changed on the device within a predetermined time period prior to receiving the user input. When a first setting was changed within the predetermined time period prior to receiving the user input, the device restores the changed setting to a prior setting.

VOICE CONTROL TO DIAGNOSE INADVERTENT ACTIVATION OF ACCESSIBILITY FEATURES

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This Application claims the benefit of U.S. Provisional Application No. 61/785,853, filed on March 14, 2013, entitled VOICE CONTROL TO DIAGNOSE INADVERTENT ACTIVATION OF ACCESSIBILITY FEATURES, which is hereby incorporated by reference in its entirety for all purposes.

TECHNICAL FIELD

[0002] The disclosed implementations relate generally to digital assistants, and more specifically to a method and system for diagnosing an inadvertent activation of a user interface setting and restoring a prior user interface setting.

BACKGROUND

[0003] Accessibility features provide solutions that allow the elderly, young children, or people with disabilities to operate electronic devices. For example, voice-over technology provides advanced screen-reading technology allowing visually impaired individuals to hear a spoken description of what's onscreen and control their electronic device using only the keyboard. Screen magnification features also assist visually impaired individuals to operate electronic devices by magnifying the display or portions of the display up to or beyond 20-times normal resolution. Slow key technology assists individuals having physical or motor-skill challenges with typing by adding a delay between when a key is pressed and when the action is accepted, reducing the chance of mistake. Sticky key technology also assists individuals having physical or motor-skill challenges by allowing key combinations (e.g., chords) to be entered sequentially, rather than simultaneously.

[0004] While these features are important for users who know how to use them, they can be disorienting for users who do not know how to operate them properly. Often, these accessibility features completely change the normal modes of interaction familiar to users of the electronic device. For example, screen reading technology changes the function of the keyboard such that individual keys control a screen reader, rather than cause the input of text. Other features change the normal output of the device. For example, screen magnification technologies magnify portions or the entirety of a user interface display in fashions that may be unintelligible to a user not accustomed to the function.

[0005] Inadvertent activation often leaves users with few options other than to call customer service, as they do not know how to deactivate these features. Conventionally, there are no alternative modes of interacting with the electronic device that the user can use in all circumstances (e.g., independent of the status of user interface settings such as accessibility features) to turn off accidentally invoked accessibility features. Thus, it would be beneficial to provide methods and systems that can identify when a user is having difficulty operating an electronic device, caused by a change in a user interface setting, *e.g.*, activation of an accessibility feature, and allow the user to restore changed user interface settings.

SUMMARY

[0006] Many electronic devices allow a user to customize a user interface to match their individual needs and preferences. For example, accessibility features are available that allow people with disabilities to use electronic devices such as digital music players, smart phones, tablet computers, laptops, and desktop computers. These accessibility features change the way in which the user interacts with the device, for example, by magnifying the display of a graphical user interface, inactivating certain input elements, or providing audible feedback rather than, or in addition to, visual feedback. However, because these features alter the way that the user interacts with the electronic device, disabling these features may require user inputs that are not intuitive and/or very different from the way a user typically interacts with the device. Furthermore, users who inadvertently activate these features may not be familiar with how to interact with the device after activation, and may encounter difficulty performing otherwise routine tasks afterwards.

[0007] Accordingly, there is a need for electronic devices with faster and more intuitive methods for diagnosing inadvertent activation of a user interface setting, such as an accessibility feature, and for restoring a prior user interface setting. Such methods may complement or replace conventional methods for diagnosing inadvertent activation of a user interface setting and restoring a prior user interface setting. Such methods and systems reduce the cognitive burden on a user and produce a more efficient human-machine interface.

[0008] To satisfy these and other needs, methods and systems are provided here that utilize voice-based assistants as an alternative mode of interacting with an electronic device, regardless of active user interface settings. Recently, voice-based digital assistants, such as APPLE's SIRI, have been introduced into the marketplace to handle various tasks such as

web searching and navigation. One advantage of such voice-based digital assistants is that users can interact with a device in a hands-free manner without handling or even looking at the device. Hands-Free operation can be particularly beneficial when a person cannot, or should not, physically handle a device, such as when they are driving.

[0009] Here, voice-based digital assistants are used as a reliable method for interacting with an electronic device, regardless of the active user interface settings. Using the methods and systems provided herein, these voice-based digital assistants allow a user to ask what is wrong with their electronic device through spoken words. For example, a user can ask the voice-based assistant questions such as: “Why is my screen zoomed-in?” “Why is my computer talking to me?” “Why doesn’t my keyboard work?” The digital assistant then checks the status of user interface settings, such as accessibility features, that may be disrupting the normal operation of the electronic device, and restores a prior user interface environment that is more familiar to the user. Alternatively or additionally, the voice-based digital assistant can educate the user as to why their electronic device is functioning in a particular manner; how to avoid activating these features in the future; and how to disable these features when they are activated. Thus, even when user interface settings which completely change the normal modes of user-device interaction are active (e.g., accessibility features), users will have a familiar and reliable means for restoring familiar user interface settings.

[0010] The implementations described below provide systems and methods for diagnosing inadvertent activation of a user interface setting and restoring a prior user interface setting.

[0011] Some implementations provide a method performed at an electronic device for diagnosing inadvertent activation of a user interface setting on the electronic device. The method includes receiving a user input indicating that the user is having difficulty operating the electronic device and determining whether a setting was changed on the device within a predetermined time period prior to receiving the user input. The method also includes, upon a determination that a first setting was changed within the predetermined time period prior to receiving the user input, restoring the changed setting to a prior setting.

[0012] Some implementations provide an electronic device that includes a processing unit configured to detect a user input indicative that the user is having difficulty operating the electronic device. The processing unit is also configured to determine whether a setting was

changed on the device within a predetermined time period prior to the detection of the user input. The processor is further configured to, upon a determination that a first setting was changed within the predetermined time period prior to detection of the user input, restore the changed setting to a prior setting.

[0013] In accordance with some implementations, a computer-readable storage medium (e.g., a non-transitory computer readable storage medium) is provided, the computer-readable storage medium storing one or more programs for execution by one or more processors of an electronic device, the one or more programs including instructions for performing any of the methods described herein,

[0014] In accordance with some implementations, an electronic device (e.g., a mobile electronic device) is provided that comprises means for performing any of the methods described herein.

[0015] In accordance with some implementations, an electronic device (e.g., a portable electronic device) is provided that comprises a processing unit configured to perform any of the methods described herein.

[0016] In accordance with some implementations, an electronic device (e.g., a portable electronic device) is provided that comprises one or more processors and memory storing one or more programs for execution by the one or more processors, the one or more programs including instructions for performing any of the methods described herein.

[0017] In accordance with some implementations, an information processing apparatus for use in an electronic device is provided, the information processing apparatus comprising means for performing any of the methods described herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] Figure 1 is a block diagram illustrating an environment in which a digital assistant operates in accordance with some implementations.

[0019] Figure 2 is a block diagram illustrating a digital assistant client system in accordance with some implementations.

[0020] Figure 3A is a block diagram illustrating a standalone digital assistant system or a digital assistant server system in accordance with some implementations.

[0021] Figure 3B is a block diagram illustrating functions of the digital assistant shown in Figure 3A in accordance with some implementations.

[0022] Figure 3C is a network diagram illustrating a portion of an ontology in accordance with some implementations.

[0023] Figure 4 is a block diagram of an electronic device in accordance with some implementations.

[0024] Figures 5A-5B are flow diagrams illustrating a method of diagnosing inadvertent activation of a user interface setting and restoring a prior user interface setting in accordance with some implementations.

[0025] Figure 6 is a flow chart illustrating exemplary processing steps for facilitating diagnosis of inadvertent activation of a user interface setting and restoring a prior user interface setting, when initiated by a user, where x and y are predetermined thresholds, in accordance with some implementations.

[0026] Figure 7 is a flow chart illustrating exemplary processing steps for facilitating diagnosis of inadvertent activation of a user interface setting and restoring a prior user interface setting, when initiated by an electronic device, where x and y are predetermined thresholds, in accordance with some implementations.

[0027] Figure 8 is a screen-shot illustrating exemplary user-device interactions facilitating diagnosis of inadvertent activation of a user interface setting and restoring a prior user interface setting, when initiated by a user, in accordance with some implementations.

[0028] Figure 9 is a screen-shot illustrating exemplary user-device interactions facilitating diagnosis of inadvertent activation of a user interface setting and restoring a prior user interface setting, when initiated by an electronic device, in accordance with some implementations.

[0029] Figure 10 is a screen-shot illustrating exemplary user-device interactions facilitating diagnosis of inadvertent activation of a user interface setting and restoring a prior user interface setting, when initiated by an electronic device, in accordance with some implementations.

[0030] Like reference numerals refer to corresponding parts throughout the drawings.

DESCRIPTION OF IMPLEMENTATIONS

[0031] Figure 1 is a block diagram of an operating environment 100 of a digital assistant according to some implementations. The terms “digital assistant,” “virtual assistant,” “intelligent automated assistant,” “voice-based digital assistant,” or “automatic digital assistant,” refer to any information processing system that interprets natural language input in spoken and/or textual form to infer user intent (e.g., identify a task type that corresponds to the natural language input), and performs actions based on the inferred user intent (e.g., perform a task corresponding to the identified task type). For example, to act on a inferred user intent, the system can perform one or more of the following: identifying a task flow with steps and parameters designed to infer user intent (e.g., identifying a task type), 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 (e.g., sending a request to a service provider); and generating output responses to the user in an audible (e.g., speech) and/or visual form.

[0032] Specifically, once initiated, a digital assistant system is capable of accepting a user request at least partially in the form of a natural language command, request, statement, narrative, and/or inquiry. Typically, the user request seeks either an informational answer or performance of a task by the digital assistant system. A satisfactory response to the user request is generally either provision of the requested informational answer, performance of the requested task, or a combination of the two. For example, a user may ask the digital assistant system a question, such as “Where am I right now?” Based on the user’s current location, the digital assistant may answer, “You are in Central Park near the west gate.” The user may also request the performance of a task, for example, by stating “Please invite my friends to my girlfriend’s birthday party next week.” In response, the digital assistant may acknowledge the request by generating a voice output, “Yes, right away,” and then send a suitable calendar invite from the user’s email address to each of the user’s friends listed in the user’s electronic address book or contact list. 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.).

[0033] As shown in Figure 1, in some implementations, a digital assistant system is implemented according to a client-server model. The digital assistant system includes a

client-side portion (e.g., 102a and 102b) (hereafter “digital assistant (DA) client 102”) executed on a user device (e.g., 104a and 104b), and a server-side portion 106 (hereafter “digital assistant (DA) server 106”) executed on a server system 108. The DA client 102 communicates with the DA server 106 through one or more networks 110. The DA client 102 provides client-side functionalities such as user-facing input and output processing and communications with the DA server 106. The DA server 106 provides server-side functionalities for any number of DA clients 102 each residing on a respective user device 104 (also called a client device or electronic device).

[0034] In some implementations, the DA server 106 includes a client-facing I/O interface 112, one or more processing modules 114, data and models 116, an UO interface to external services 118, and, optionally, a settings history module 140. The client-facing I/O interface facilitates the client-facing input and output processing for the digital assistant server 106. The one or more processing modules 114 utilize the data and models 116 to determine the user’s intent based on natural language input and perform task execution based on the inferred user intent. Settings history module 140 creates tags in accordance with a user interface setting change on user device 104, and stores tags in association with a user interface setting change on user device 104.

[0035] In some implementations, the DA server 106 communicates with external services 120 (e.g., navigation service(s) 122-1, messaging service(s) 122-2, information service(s) 122-3, calendar service 122-4, telephony service 122-5, restoration service(s) 122-6, etc.) through the network(s) 110 for task completion or information acquisition. The I/O interface to the external services 118 facilitates such communications.

[0036] Examples of the user device 104 include, but are not limited to, a handheld computer, a personal digital assistant (PDA), a tablet computer, a laptop computer, a desktop computer, a cellular telephone, a smartphone, an enhanced general packet radio service (EGPRS) mobile phone, a media player, a navigation device, a game console, a television, a remote control, or a combination of any two or more of these data processing devices or any other suitable data processing devices. More details on the user device 104 are provided in reference to an exemplary user device 104 shown in Figure 2.

[0037] Examples of the communication network(s) 110 include local area networks (LAN) and wide area networks (WAN), e.g., the Internet. The communication network(s) 110 may be implemented using any known network protocol, including various wired or

wireless protocols, such as 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.

[0038] The server system 108 can be implemented on at least one data processing apparatus and/or a distributed network of computers. In some implementations, the server system 108 also employs various virtual devices and/or services of third-party service providers (e.g., third-party cloud service providers) to provide the underlying computing resources and/or infrastructure resources of the server system 108.

[0039] Although the digital assistant system shown in Figure 1 includes both a client side portion (e.g., the DA client 102) and a server-side portion (e.g., the DA server 106), in some implementations, a digital assistant system refers only to the server-side portion (e.g., the DA server 106). In some implementations, 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 example, in some implementations, the DA client 102 is a thin-client that provides only user-facing input and output processing functions, and delegates all other functionalities of the digital assistant to the DA server 106. In some other implementations, the DA client 102 is configured to perform or assist one or more functions of the DA server 106.

[0040] Figure 2 is a block diagram of a user device 104 in accordance with some implementations. The user device 104 includes a memory interface 202, one or more processors 204, and a peripherals interface 206. The various components in the user device 104 are coupled by one or more communication buses or signal lines. The user device 104 includes various sensors, subsystems, and peripheral devices that are coupled to the peripherals interface 206. The sensors, subsystems, and peripheral devices gather information and/or facilitate various functionalities of the user device 104.

[0041] For example, in some implementations, a motion sensor 210 (e.g., an accelerometer), a light sensor 212, a GPS receiver 213, a temperature sensor, and a proximity sensor 214 are coupled to the peripherals interface 206 to facilitate orientation, light, and proximity sensing functions. In some implementations, other sensors 216, such as a

biometric sensor, barometer, and the like, are connected to the peripherals interface 206, to facilitate related functionalities.

[0042] In some implementations, the user device 104 includes a camera subsystem 220 coupled to the peripherals interface 206. In some implementations, an optical sensor 222 of the camera subsystem 220 facilitates camera functions, such as taking photographs and recording video clips. In some implementations, the user device 104 includes one or more wired and/or wireless communication subsystems 224 provide communication functions. The communication subsystems 224 typically includes various communication ports, radio frequency receivers and transmitters, and/or optical (e.g., infrared) receivers and transmitters. In some implementations, the user device 104 includes an audio subsystem 226 coupled to one or more speakers 228 and one or more microphones 230 to facilitate voice-enabled functions, such as voice recognition, voice replication, digital recording, and telephony functions.

[0043] In some implementations, an I/O subsystem 240 is also coupled to the peripheral interfaces 206. In some implementations, the user device 104 includes a touch screen 246, and the I/O subsystem 240 includes a touch screen controller 242 coupled to the touch screen 246. When the user device 104 includes the touch screen 246 and the touch screen controller 242, the touch screen 246 and the touch screen controller 242 are typically configured to, for example, detect contact and movement or break thereof using any of a plurality of touch sensitivity technologies, such as capacitive, resistive, infrared, surface acoustic wave technologies, proximity sensor arrays, and the like. In some implementations, the user device 104 includes a display that does not include a touch-sensitive surface. In some implementations, the user device 104 includes a separate touch-sensitive surface. In some implementations, the user device 104 includes other input controller(s) 244. When the user device 104 includes the other input controller(s) 244, the other input controller(s) 244 are typically coupled to other input/control devices 248, such as one or more buttons (e.g., home button 810), rocker switches, thumb-wheel, infrared port, USB port, and/or a pointer device such as a stylus.

[0044] The memory interface 202 is coupled to memory 250. In some implementations, memory 250 includes a non-transitory computer readable medium, such as high-speed random access memory and/or non-volatile memory (e.g., one or more magnetic disk storage devices, one or more flash memory devices, one or more optical storage devices, and/or other non-volatile solid-state memory devices).

[0045] In some implementations, memory 250 stores an operating system 252, a communications module 254, a graphical user interface module 256, a sensor processing module 258, a phone module 260, and applications 262, and a subset or superset thereof. The operating system 252 includes instructions for handling basic system services and for performing hardware dependent tasks. The communications module 254 facilitates communicating with one or more additional devices, one or more computers and/or one or more servers. The graphical user interface module 256 facilitates graphic user interface processing. The sensor processing module 258 facilitates sensor-related processing and functions (e.g., processing voice input received with the one or more microphones 230). The phone module 260 facilitates phone-related processes and functions. The applications 262 facilitates various functionalities of user applications, such as electronic-messaging, web browsing, media processing, navigation, imaging and/or other processes and functions. In some implementations, the user device 104 stores in memory 250 one or more software applications 270-1 and 270-2 each associated with at least one of the external service providers.

[0046] As described above, in some implementations, memory 250 also stores client side digital assistant instructions (e.g., in a digital assistant client module 264) and various user data 266 (e.g., user-specific vocabulary data, preference data, and/or other data such as the user's electronic address book or contact list, to-do lists, shopping lists, etc.) to provide the client-side functionalities of the digital assistant.

[0047] In various implementations, the digital assistant client module 264 is capable of accepting voice input, text input, touch input, and/or gestural input through various user interfaces (e.g., the I/O subsystem 240) of the user device 104. The digital assistant client module 264 is also capable of providing output in audio, visual, and/or tactile forms. 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, the digital assistant client module 264 communicates with the digital assistant server (e.g., the digital assistant server 106, Figure 1) using the communication subsystems 224.

[0048] In some implementations, the digital assistant client module 264 utilizes various sensors, subsystems and peripheral devices to gather additional information from the surrounding environment of the user device 104 to establish a context associated with a user input. In some implementations, the digital assistant client module 264 provides the context

information or a subset thereof with the user input to the digital assistant server (e.g., the digital assistant server 106, Figure 1) to help infer the user's intent.

[0049] In some implementations, the context information that can accompany the user input includes sensor information, e.g., lighting, ambient noise, ambient temperature, images or videos of the surrounding environment, etc. In some implementations, the context information also includes 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 implementations, information related to the software state of the user device 106, e.g., running processes, installed programs, past and present network activities, background services, error logs, resources usage, etc., of the user device 104 is also provided to the digital assistant server (e.g., the digital assistant server 106, Figure 1) as context information associated with a user input.

[0050] In some implementations, the DA client module 264 selectively provides information (e.g., at least a portion of the user data 266) stored on the user device 104 in response to requests from the digital assistant server. In some implementations, the digital assistant client module 264 also elicits additional input from the user via a natural language dialogue or other user interfaces upon request by the digital assistant server 106 (Figure 1). The digital assistant client module 264 passes the additional input to the digital assistant server 106 to help the digital assistant server 106 in intent deduction and/or fulfillment of the user's intent expressed in the user request.

[0051] In some implementations, memory 250 may include additional instructions or fewer instructions. Furthermore, various functions of the user device 104 may be implemented in hardware and/or in firmware, including in one or more signal processing and/or application specific integrated circuits, and the user device 104, thus, need not include all modules and applications illustrated in Figure 2.

[0052] Figure 3A is a block diagram of an exemplary digital assistant system 300 (also referred to as the digital assistant) in accordance with some implementations. In some implementations, the digital assistant system 300 is implemented on a standalone computer system. In some implementations, the digital assistant system 300 is distributed across multiple computers. In some implementations, some of the modules and functions of the digital assistant are divided into a server portion and a client portion, where the client portion resides on a user device (e.g., the user device 104) and communicates with the server portion

(e.g., the server system 108) through one or more networks, e.g., as shown in Figure 1. In some implementations, the digital assistant system 300 is an implementation of the server system 108 (and/or the digital assistant server 106) shown in Figure 1. In some implementations, the digital assistant system 300 is implemented in a user device (e.g., the user device 104, Figure 1), thereby eliminating the need for a client-server system. It should be noted that the digital assistant system 300 is only one example of a digital assistant system, and that the digital assistant system 300 may 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 Figure 3A may be implemented in hardware, software, firmware, including one or more signal processing and/or application specific integrated circuits, or a combination of thereof.

[0053] The digital assistant system 300 includes memory 302, one or more processors 304, an input/output (I/O) interface 306, and a network communications interface 308. These components communicate with one another over one or more communication buses or signal lines 310.

[0054] In some implementations, memory 302 includes a non-transitory computer readable medium, such as high-speed random access memory and/or a non-volatile computer readable storage medium (e.g., one or more magnetic disk storage devices, one or more flash memory devices, one or more optical storage devices, and/or other non-volatile solid-state memory devices).

[0055] The I/O interface 306 couples input/output devices 316 of the digital assistant system 300, such as displays, a keyboards, touch screens, and microphones, to the user interface module 322. The I/O interface 306, in conjunction with the user interface module 322, receives user inputs (e.g., voice input, keyboard inputs, touch inputs, etc.) and processes them accordingly. In some implementations, when the digital assistant is implemented on a standalone user device, the digital assistant system 300 includes any of the components and I/O and communication interfaces described with respect to the user device 104 in Figure 2 (e.g., one or more microphones 230). In some implementations, the digital assistant system 300 represents the server portion of a digital assistant implementation, and interacts with the user through a client-side portion residing on a user device (e.g., the user device 104 shown in Figure 2).

[0056] In some implementations, the network communications interface 30K includes wired communication port(s) 312 and/or wireless transmission and reception circuitry 314. The wired communication port(s) receive and send communication signals via one or more wired interfaces, e.g., Ethernet, Universal Serial Bus (USB), FIREWIRE, etc. The wireless circuitry 314 typically receives and sends RF signals and/or optical signals from/to communications networks and other communications devices. The wireless communications may 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. The network communications interface 30K enables communication between the digital assistant system 300 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.

[0057] In some implementations, the non-transitory computer readable storage medium of memory 302 stores programs, modules, instructions, and data structures including all or a subset of: an operating system 318, a communications module 320, a user interface module 322, one or more applications 324, a digital assistant module 326, and settings 370. The one or more processors 304 execute these programs, modules, and instructions, and reads/writes from/to the data structures.

[0058] The operating system 318 (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 communications between various hardware, firmware, and software components.

[0059] The communications module 320 facilitates communications between the digital assistant system 300 with other devices over the network communications interface 308. For example, the communication module 320 may communicate with the communications module 254 of the device 104 shown in Figure 2. The communications module 320 also includes various software components for handling, data received by the wireless circuitry 314 and/or wired communications port 312.

[0060] In some implementations, the user interface module 322 receives commands and/or inputs from a user via the I/O interface 306 (e.g., from a keyboard, touch screen, and/or microphone), and provides user interface objects on a display.

[0061] The applications 324 include programs and/or modules that are configured to be executed by the one or more processors 304. For example, if the digital assistant system is implemented on a standalone user device, the applications 324 may include user applications, such as games, a calendar application, a navigation application, or an email application. If the digital assistant system 300 is implemented on a server Farm, the applications 324 may include resource management applications, diagnostic applications, or scheduling applications, for example.

[0062] Memory 302 also stores the digital assistant module (or the server portion of a digital assistant) 326. In some implementations, the digital assistant module 326 includes the following sub-modules, or a subset or superset thereof: an input/output processing module 328, a speech-to-text (STT) processing module 330, a natural language processing module 332, a dialogue flow processing module 334, a task flow processing module 336, a service processing module 338, and a diagnostic module 366. Each of these processing modules has access to one or more of the following data and models of the digital assistant 326, or a subset or superset thereof: ontology 360, vocabulary index 344, user data 348, categorization module 349, disambiguation module 350, task flow models 354, service models 356, user interface integration module 340, and settings integration module 368.

[0063] In some implementations, using the processing modules (e.g., the input/output processing module 328, the STT processing module 330, the natural language processing module 332, the dialogue flow processing module 334, the task flow processing module 336, the service processing module 338, and/or the user interface integration module 340), data, and models implemented in the digital assistant module 326, the digital assistant system 300 performs at least some of the following: 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, names, intentions, etc.); determining the task flow for fulfilling the inferred intent; and executing the task flow to fulfill the inferred intent. In some implementations, the digital assistant also takes appropriate actions when a satisfactory response was not or could not be provided to the user for various reasons.

[0064] In some implementations, as discussed below, the digital assistant system 300 identifies, From a natural language input, a user's intent to restore a user interface setting (e.g., to turn off an accessibility feature), and processes the natural language input so as to

restore an appropriate user interface setting (e.g., turn off an appropriate accessibility feature).

[0065] The settings 370 facilitate user customization of methods executed by processor 304, including methods that rely on global settings (e.g., system-wide volume, execution of sounds associated with common inputs, functionality of input receivers, display brightness, default language, or accessibility feature) and methods specific to an application (e.g., in-application volume, external content associated with an application, language associated with an application, or allowed functionalities of an application). In some implementations, the settings 370 include the following sub-modules, or a subset or superset thereof: application specific settings module 372, general user interface settings module 374, and restoration module 390. Each of these settings modules has access to one or more of the following data and models of the settings 370, or a subset or superset thereof: accessibility 376, language 378, volume 380, display 382, clock module 392, and settings history 394.

[0066] As shown in Figure 3B, in some implementations, the I/O processing module 328 interacts with the user through the I/O devices 316 in Figure 3A or with a user device (e.g., a user device 104 in Figure 1) through the network communications interface 308 in Figure 3A to obtain user input (e.g., a speech input) and to provide responses to the user input. The I/O processing module 328 optionally obtains context information associated with the user input from the user device, along with or shortly after the receipt of the user input. The context information includes user-specific data, vocabulary, and/or preferences relevant to the user input. In some implementations, the context information also includes software and hardware states (e.g., active settings) of the device (e.g., the user device 104 in Figure 1) 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 implementations, the I/O processing module 328 also sends follow-up questions to, and receives answers from, the user regarding the user request. In some implementations, when a user request is received by the I/O processing module 328 and the user request contains a speech input, the I/O processing module 328 forwards the speech input to the speech-to-text (STT) processing module 330 for speech-to-text conversions.

[0067] In some implementations, the speech-to-text processing module 330 receives speech input (e.g., a user utterance captured in a voice recording) through the I/O processing module 328. In some implementations, the speech-to-text processing module 330 uses various acoustic and language models to recognize the speech input as a sequence of

phonemes, and ultimately, a sequence of words or tokens written in one or more languages. The speech-to-text processing module 330 is implemented using any suitable speech recognition techniques, acoustic models, and language models, such as Hidden Markov Models, Dynamic Time Warping (DTW)-based speech recognition, and other statistical and/or analytical techniques. In some implementations, the speech-to-text processing can be performed at least partially by a third party service or on the user's device. Once the speech-to-text processing module 330 obtains the result of the speech-to-text processing (e.g., a sequence of words or tokens), it passes the result to the natural language processing module 332 for intent deduction.

[0068] The natural language processing module 332 ("natural language processor") of the digital assistant 326 takes the sequence of words or tokens ("token sequence") generated by the speech-to-text processing module 330, and attempts to associate the token sequence with one or more "actionable intents" recognized by the digital assistant. As used herein, an "actionable intent" represents a task that can be performed by the digital assistant 326 and/or the digital assistant system 300 (Figure 3A), and has an associated task flow implemented in the task flow models 354. The associated task flow is a series of programmed actions and steps that the digital assistant system 300 takes in order to perform the task. The scope of a digital assistant system's capabilities is dependent on the number and variety of task flows that have been implemented and stored in the task flow models 354, or in other words, on the number and variety of "actionable intents" that the digital assistant system 300 recognizes. The effectiveness of the digital assistant system 300, however, is also dependent on the digital assistant system's ability to infer the correct "actionable intent(s)" from the user request expressed in natural language.

[0069] In some implementations, in addition to the sequence of words or tokens obtained from the speech-to-text processing module 330, the natural language processor 332 also receives context information associated with the user request (e.g., from the I/O processing module 328). The natural language processor 332 optionally uses the context information to clarify, supplement, and/or further define the information contained in the token sequence received from the speech-to-text processing module 330. The context information includes, for example, user preferences, hardware and/or software states of the user device (e.g., active settings), 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.

[0070] In some implementations, the natural language processing is based on an ontology 360. The ontology 360 is a hierarchical structure containing a plurality of nodes, each node representing either an “actionable intent” or a “property” relevant to one or more of the “actionable intents” or other “properties.” As noted above, an “actionable intent” represents a task that the digital assistant system 300 is capable of performing (e.g., a task that is “actionable” or can be acted on). A “property” represents a parameter associated with an actionable intent or a sub-aspect of another property. A linkage between an actionable intent node and a property node in the ontology 360 defines how a parameter represented by the property node pertains to the task represented by the actionable intent node.

[0071] In some implementations, the ontology 360 is made up of actionable intent nodes and property nodes. Within the ontology 360, each actionable intent node is linked to one or more property nodes either directly or through one or more intermediate property nodes. Similarly, each property node is linked to one or more actionable intent nodes either directly or through one or more intermediate property nodes. For example, the ontology 360 shown in Figure 3C includes a “restaurant reservation” node, which is an actionable intent node. Property nodes “restaurant,” “date/time” (for the reservation), and “party size” are each directly linked to the “restaurant reservation” node (i.e., the actionable intent node). In addition, property nodes “cuisine,” “price range,” “phone number,” and “location” are sub-nodes of the property node “restaurant,” and are each linked to the “restaurant reservation” node (i.e., the actionable intent node) through the intermediate property node “restaurant.” For another example, the ontology 360 shown in Figure 3C also includes a “set reminder” node, which is another actionable intent node. Property nodes “date/time” (for the setting the reminder) and “subject” (for the reminder) are each linked to the “set reminder” node. Since the property “date/time” is relevant to both the task of making a restaurant reservation and the task of setting a reminder, the property node “date/time” is linked to both the “restaurant reservation” node and the “set reminder” node in the ontology 360.

[0072] An actionable intent node, along with its linked concept nodes, may be described as a “domain.” In the present discussion, each domain is associated with a respective actionable intent, and refers to the group of nodes (and the relationships there between) associated with the particular actionable intent. For example, the ontology 360 shown in Figure 3C’ includes an example of a restaurant reservation domain 362 and an example of a reminder domain 364 within the ontology 360. 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.” The reminder domain 364 includes the actionable intent node “set reminder,” and property nodes “subject” and “date/time.” In some implementations, the ontology 360 is made up of many domains. Each domain may share one or more property nodes with one or more other domains. For example, the “date/time” property node may be associated with many other domains (e.g., a scheduling domain, a travel reservation domain, a movie ticket domain, etc.), in addition to the restaurant reservation domain 362 and the reminder domain 364.

[0073] While Figure 3C illustrates two exemplary domains within the ontology 360, the ontology 360 may include other domains (or actionable intents), such as “initiate a phone call,” “find directions,” “schedule a meeting,” “send a message,” “provide an answer to a question,” “restore a user interface setting,” and so on. For example, a “send a message” domain is 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” may be further defined, for example, by the sub-property nodes such as “recipient name” and “message address.”

[0074] In some implementations, the ontology 360 includes all the domains (and hence actionable intents) that the digital assistant is capable of understanding and acting upon. In some implementations, the ontology 360 may be modified, such as by adding or removing domains or nodes, or by modifying relationships between the nodes within the ontology 360.

[0075] In some implementations, nodes associated with multiple related actionable intents may be clustered under a “super domain” in the ontology 360. For example, a “travel” super-domain may include a cluster of property nodes and actionable intent nodes related to travels. The actionable intent nodes related to travels may 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 “travels” super domain) may have many property nodes in common. For example, the actionable intent nodes for “airline reservation,” “hotel reservation,” “car rental,” “get directions,” “find points of interest” may share one or more of the property nodes “start location,” “destination,” “departure date/time,” “arrival date/time,” and “party size.”

[0076] In some implementations, each node in the ontology 360 is associated with a set of words and/or phrases that are relevant to the property or actionable intent represented by

the node. The respective set of words and/or phrases associated with each node is the so-called “vocabulary” associated with the node. The respective set of words and/or phrases associated with each node can be stored in the vocabulary index 344 (Figure 3B) in association with the property or actionable intent represented by the node. For example, returning to Figure 3B, the vocabulary associated with the node for the property of “restaurant” may include words such as “food,” “drinks,” “cuisine,” “hungry,” “cat,” “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” may include words and phrases such as “call,” “phone,” “dial,” “ring,” “call this number,” “make a call to,” and so on. The vocabulary index 344 optionally includes words and phrases in different languages.

[0077] In some implementations, the natural language processor 332 shown in Figure 3B receives the token sequence (e.g., a text string) from the speech-to-text processing module 330, and determines what nodes are implicated by the words in the token sequence. In some implementations, if a word or phrase in the token sequence is found to be associated with one or more nodes in the ontology 360 (via the vocabulary index 344), the word or phrase will “trigger” or “activate” those nodes. When multiple nodes are “triggered,” based on the quantity and/or relative importance of the activated nodes, the natural language processor 332 will select one of the actionable intents as the task (or task type) that the user intended the digital assistant to perform. In some implementations, the domain that has the most “triggered” nodes is selected. In some implementations, the domain having the highest confidence value (e.g., based on the relative importance of its various triggered nodes) is selected. In some implementations, the domain is selected based on a combination of the number and the importance of the triggered nodes. In some implementations, additional factors are considered in selecting the node as well, such as whether the digital assistant system 300 has previously correctly interpreted a similar request from a user.

[0078] In some implementations, the digital assistant system 300 also stores names of specific entities in the vocabulary index 344, so that when one of these names is detected in the user request, the natural language processor 332 will be able to recognize that the name refers to a specific instance of a property or sub-property in the ontology. In some implementations, the names of specific entities are names of businesses, restaurants, people, movies, and the like. In some implementations, the digital assistant system 300 can search and identify specific entity names from other data sources, such as the user’s address book or contact list, a movies database, a musicians database, and/or a restaurant database. In some

implementations, when the natural language processor 332 identifies that a word in the token sequence is a name of a specific entity (such as a name in the user's address book or contact list), that word is given additional significance in selecting the actionable intent within the ontology for the user request.

[0079] For example, when the words "Mr. Santo" are recognized from the user request, and the last name "Santo" is found in the vocabulary index 344 as one of the contacts in the user's contact list, then it is likely that the user request corresponds to a "send a message" or "initiate a phone call" domain. For another example, when the words "ABC Café" are found in the user request, and the term "ABC Café" is Found in the vocabulary index 344 as the name of a particular restaurant in the user's city, then it is likely that the user request corresponds to a "restaurant reservation" domain.

[0080] User data 348 includes user-specific information, such as user-specific vocabulary, user preferences, user address, user's default and secondary languages, user's contact list, and other short-term or long-term information for each user. The natural language processor 332 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," the natural language processor 332 is able to access user data 348 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.

[0081] In some implementations, natural language processor 332 includes categorization module 349. In some implementations, the categorization module 349 determines whether each of the one or more terms in a text string (e.g., corresponding to a speech input associated with a digital photograph) is one of an entity, an activity, or a location, as discussed in greater detail below. In some implementations, the categorization module 349 classifies each term of the one or more terms as one of an entity, an activity, or a location.

[0082] Once the natural language processor 332 identifies an actionable intent (or domain) based on the user request, the natural language processor 332 generates a structured query to represent the identified actionable intent. In some implementations, the structured query includes parameters for one or more nodes within the domain for the actionable intent, and at least some of the parameters are populated with the specific information and requirements specified in the user request. For example, the user may say "Make me a dinner

reservation at a sushi place at 7.” In this case, the natural language processor 332 may 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. Based on the information contained in the user’s utterance, the natural language processor 332 may 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} are not specified in the structured query based on the information currently available. In some implementations, the natural language processor 332 populates some parameters of the structured query with received context information. For example, if the user requested a sushi restaurant “near me,” the natural language processor 332 may populate a {location} parameter in the structured query with CPS coordinates from the user device 104.

[0083] In some implementations, the natural language processor 332 passes the structured query (including any completed parameters) to the task flow processing module 336 (“task flow processor”). The task flow processor 336 is configured to perform one or more of: receiving the structured query from the natural language processor 332, completing the structured query, and performing the actions required to “complete” the user’s ultimate request. In some implementations, the various procedures necessary to complete these tasks are provided in task flow models 354. In some implementations, the task flow models 354 include procedures for obtaining additional information from the user, and task flows for performing actions associated with the actionable intent.

[0084] As described above, in order to complete a structured query, the task flow processor 336 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, the task flow processor 336 invokes the dialogue processing module 334 (“dialogue processor”) to engage in a dialogue with the user. In some implementations, the dialogue processing module 334 determines how (and/or when) to ask the user for the additional information, and receives and processes the user responses. In some implementations, the questions are provided to and answers are received from the users through the I/O processing module 328. For example, the dialogue processing module 334

presents dialogue output to the user via audio and/or visual output, and receives input from the user via spoken or physical (e.g., touch gesture) responses. Continuing with the example above, when the task flow processor 336 invokes the dialogue processor 334 to determine the “party size” and “date” information for the structured query associated with the domain “restaurant reservation,” the dialogue processor 334 generates questions such as “For how many people?” and “On which day?” to pass to the user. Once answers are received From the user, the dialogue processing module 334 populates the structured query with the missing information, or passes the information to the task flow processor 336 to complete the missing information from the structured query.

[0085] In some cases, the task flow processor 336 may receive a structured query that has one or more ambiguous properties. For example, a structured query for the “send a message” domain may indicate that the intended recipient is “Bob,” and the user may have multiple contacts named “Bob.” The task flow processor 336 will request that the dialogue processor 334 disambiguate this property of the structured query. In turn, the dialogue processor 334 may ask the user “Which Bob?”, and display (or read) a list of contacts named “Bob” from which the user may choose.

[0086] In some implementations, dialogue processor 334 includes disambiguation module 350. In some implementations, disambiguation module 350 disambiguates one or more ambiguous terms (e.g., one or more ambiguous terms in a text string corresponding to a speech input associated with a user interface setting), In some implementations, disambiguation module 350 identifies that a first term of the one or more terms has multiple candidate meanings, prompts a user for additional information about the first term, receives the additional information from the user in response to the prompt and identifies the entity, activity, location, or user interface setting associated with the first term in accordance with the additional information.

[0087] In some implementations, disambiguation module 350 disambiguates pronouns. In such implementations, disambiguation module 350 identifies one of the one or more terms as a pronoun and determines a noun to which the pronoun refers. In some implementations, disambiguation module 350 determines a noun to which the pronoun refers by using a contact list associated with a user of the electronic device. Alternatively, or in addition, disambiguation module 350 determines a noun to which the pronoun refers as a name of an entity, an activity, or a location identified in a previous speech input associated with a previously tagged media file, (e.g., a digital photograph).

[0088] In some implementations, disambiguation module 350 accesses system history information (*e.g.*, settings history 394) of an electronic device (*e.g.*, user device 104) for determining a meaning of one or more terms. In some implementations, disambiguation module 350 identifies a term associated with user difficulty (*e.g.*, “why,” “help,” “trouble,” “difficulty,” “can’t,” or “how”) or user frustration (*e.g.*, “darn” or other words indicating frustration) and accesses settings history 394 to determine if a user interface setting has been changed recently, which may be causing the user difficulty. In some implementations, when more than one user interface setting has been changed, disambiguation module 350 prompts a user for additional information about their apparent difficulty. Disambiguation module 350 then uses the user’s response to the prompt to determine which, if any, of the setting changes are causing the difficulty.

[0089] In some implementations, disambiguation module 350 accesses information obtained from one or more sensors (*e.g.*, proximity sensor 214, light sensor 212, GPS receiver 213, temperature sensor 215, and motion sensor 210) of a handheld electronic device (*e.g.*, user device 104) for determining a meaning of one or more of the terms. In some implementations, disambiguation module 350 identifies two terms each associated with one of an entity, an activity, or a location. For example, a first of the two terms refers to a person, and a second of the two terms refers to a location. In some implementations, disambiguation module 350 identifies three terms each associated with one of an entity, an activity, or a location.

[0090] Once the task flow processor 336 has completed the structured query for an actionable intent, the task flow processor 336 proceeds to perform the ultimate task associated with the actionable intent. Accordingly, the task flow processor 336 executes the steps and instructions in the task flow model according to the specific parameters contained in the structured query. For example, the task flow model for the actionable intent of “restaurant reservation” may 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}, the task flow processor 336 may perform the steps of: (1) logging onto a server of the ABC Cafe or a restaurant reservation system that is configured to accept reservations for multiple restaurants, such as the ABC Cafe, (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. In another

example, described in greater detail below, the task flow processor 336 executes steps and instructions associated with restoring a user interface setting (e.g., turning off an accessibility feature), for example in conjunction with settings history module 140.

[0091] In some implementations, the task flow processor 336 employs the assistance of a service processing module 338 (“service processor”) to complete a task requested in the user input or to provide an informational answer requested in the user input. For example, the service processor 338 can act on behalf of the task flow processor 336 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, invoke or interact with third party services (e.g. a restaurant reservation portal, a social networking website or service, or a banking portal), or change a User interface setting (e.g., restore a user interface setting and/or turn off an accessibility feature). In some implementations, the protocols and application programming interfaces (API) required by each service can be specified by a respective service model among the service models 356. The service processor 338 accesses the appropriate service model for a service and generates requests for the service in accordance with the protocols and APIs required by the service according to the service model.

[0092] 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 parameters to the online reservation service. When requested by the task flow processor 336, the service processor 338 can establish a network connection with the online reservation service using the web address stored in the service models 356, 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.

[0093] In some implementations, the natural language processor 332, dialogue processor 334, and task flow processor 336 are used collectively and iteratively to infer and define the user’s intent, obtain information to further clarify and refine the user intent, and finally generate a response (e.g., provide an output to the user or complete a task) to fulfill the user’s intent.

[0094] In some implementations, after all of the tasks needed to fulfill the user’s request have been performed, the digital assistant 326 formulates a confirmation response, and sends the response back to the user through the I/O processing module 328. If the user request

seeks an informational answer, the confirmation response presents the requested information to the user. In some implementations, the digital assistant also requests the user to indicate whether the user is satisfied with the response produced by the digital assistant 326. In some implementations, where the user was experiencing difficulty, the digital assistant also educates the user on how to perform a task (*e.g.*, change a user interface setting or disable an accessibility feature) when Faced with a similar or the same situation.

[0095] In some implementations, the digital assistant 326 includes a diagnostics module 366. In some implementations, the diagnostics module acts in conjunction with the task flow processing module 336 to search for recent user interface setting changes (*e.g.*, in setting history 394) in response to a user input (*e.g.*, a request for help or abnormal pattern of user inputs).

[0096] In some implementations, the diagnostics module 366 includes a settings integration module 368 that acts in conjunction with the device settings 370 (*e.g.*, accessibility 376, language 378, volume 380, or display 382 settings) to diagnose an inadvertent activation of a user interface setting causing user difficulty, as explained in further detail with respect to Method 400 (Figures 4A-4B) below. For example, in some implementations, the diagnostics module 366 receives a natural language text string corresponding to a user's input, where the intent has been inferred that the user is having difficulties with the device. The diagnostics module 366 consults the restoration module 390 (*e.g.*, settings history 394) to determine if a setting change has been made within a predetermined period of time (*e.g.*, as tagged by a clock module 392). The diagnostics module 366 identifies a setting change in the setting history 394 associated with the user's difficulty and acts in conjunction with the restoration module 390 to restore a prior user-interface setting (*e.g.*, a user interface setting immediately preceding the identified setting change or a default user interface setting).

[0097] In accordance with some implementations, Figure 4 shows a Functional block diagram of an electronic device 400 configured in accordance with the principles of the various described implementations. The functional blocks of the device are, optionally, implemented by hardware, software, or a combination of hardware and software to carry out the principles of the various described implementations. It is understood by persons of skill in the art that the functional blocks described in Figure 4 are, optionally, combined or separated into sub-blocks to implement the principles of the various described

implementations. Therefore, the description herein optionally supports any possible combination or separation or further definition of the functional blocks described herein.

[0098] As shown in Figure 4, an electronic device 400 includes a sound receiving unit configured to receive user utterances, optionally, an input pattern receiving unit 404 configured to receive patterns of inputs from a user, a voice prompting unit 406 configured to prompt the user for additional information, and a processing unit 408 coupled to the sound receiving unit 402, optional input pattern receiving unit 404, and voice prompting unit 406. In some implementations, the processing unit 408 includes a determining unit 410 and a setting restoring unit 412.

[0099] The electronic device 400 is configured to receive a user input (e.g., with the sound receiving unit 402 or input pattern receiving unit 404) indicating that the user is having difficulty operating the electronic device 400. The processing unit is configured to determine whether a setting was changed on the device within a predetermined time period prior to receiving the user input (e.g., with the determining unit 410). The processing unit 408 is further configured to, upon a determination that a first setting was changed within the predetermined time period prior to receiving the user input, restore the changed setting to a prior setting (e.g., with the setting restoring unit 412).

[0100] In some implementations, the user input is a user voice input (e.g., that is received by the sound receiving unit 402 of electronic device 400). In some implementations, the user input is an abnormal pattern of user inputs (e.g., that is received by the input pattern receiving unit 404).

[0101] In some implementations, the predetermined time is a number of minutes. In some implementations, the predetermined time is a number of previous user sessions.

[0102] In some implementations, the prior setting is a default setting. In some implementations, the prior setting is a setting that immediately preceded a setting change.

[0103] In some implementations, the electronic device 400 is configured to, prior to restoring the changed setting, prompt the user with an indication that the first setting has been changed (e.g., with the voice prompting unit).

[0104] In some implementations, the electronic device 400 is further configured to, prior to restoring the changed setting, prompt the user to confirm that they are having difficulty operating the device (e.g., with the voice prompting unit 406), and the processing unit 408 is further configured to, in response to receiving confirmation that the user is having difficulty

operating the device, restore the changed setting to a prior setting (e.g., with the setting restoring unit 412).

[0105] In some implementations, the electronic device 400 is further configured to, prior to restoring the changed setting, prompt the user to confirm that the first setting is causing the user difficulty (e.g., with the voice prompting unit 406), and the processing unit is further configured to, in response to receiving confirmation that the first setting is causing the user difficulty, restore the changed setting to a prior setting (e.g., using the setting restoring unit 412).

[0106] In some implementations, the processing unit 408 is further configured to, in response to receiving a user input indicating that the first setting is not causing the user difficulty, determine if a second setting was changed on the device within a predetermined time period prior to receiving the first user input (e.g., with the determining unit 410).

[0107] In some implementations, the electronic device 400 is further configured to, prior to restoring the changed setting, prompt the user to confirm that they want to restore the changed setting to a prior setting (e.g., with the voice prompting unit 406), and the processing unit 408 is further configured to, in response to receiving confirmation that the user wants to restore the changed setting, restore the changed setting to a prior setting (e.g., with the setting restoring unit 412).

[0108] In some implementations, the voice prompting unit 406 includes the use of a text-to-speech function.

[0109] In some implementations, the processing unit 408 is further configured to explain to the user how the display feature was enabled (e.g., with the voice prompting unit 406).

[0110] In some implementations, the processing unit 408 is further configured to explain to the user how to restore the changed setting to a prior setting (e.g., with the voice prompting unit 406).

[0111] The operations in the information processing methods described above are, optionally implemented by running one or more functional modules in information processing apparatus such as general purpose processors (e.g., as described above with respect to Figures 3A) or application specific chips.

[0112] Figures 5A-5B are flow diagrams illustrating a method 500 of diagnosing the inadvertent activation of a user interface setting and restoring a prior user interface setting

when the user is having difficulty operating an electronic device, in accordance with some implementations. The method 500 is performed at an electronic device (e.g., electronic device 104, Figures 1 and 2) with a digital assistant 326 and settings 370. Some operations of method 500 are, optionally, combined and/or the order of some operations is, optionally, changed. As described below, the method 500 provides an intuitive way to diagnose the inadvertent activation of a user interface setting and restore a prior user interface setting when a user is having difficulty operating an electronic device. The method reduces the cognitive burden on a user diagnosing the inadvertent activation of a user interface setting and restoring a prior user interface setting, thereby creating a more efficient human-machine interface.

[0113] In some implementations, an electronic device (e.g., electronic device 104) receives (502) a user input indicating that the user is having difficulty operating the electronic device (e.g., user help request 602 in Figure 6 or the detection of user difficulty 700 in Figure 7). For example, the user tells a digital assistant (e.g., digital assistant 326 in Figure 3A) that they are having difficulty operating the electronic device, or the electronic device detects an irregular pattern of user inputs that is not productive.

[0114] In some implementations, the user input indicating the user is having difficulty is an intentional input (e.g., request for help) made by the user. In some implementation, the user input is (504) a user voice input (e.g., user help request 602 in Figure 6 or user query 802 in Figure 8). For example, the user: asks a digital assistant (e.g., digital assistant 326 in Figure 3A) for help; asks a digital assistant a question related to the operation or user interface of the electronic device; and/or expresses their frustration to a digital assistant. For example, the user asks the digital assistant a question related to their operational difficulty e.g.,: “Why can’t I see the entire graphical user interface display?”; “Why doesn’t the notepad application launch when I contact the associated launch icon?”; or “Why are the colors displayed on my screen inverted’?” However, in some instances, the user does not use precise terms or syntax when asking the digital assistant for assistance, thus causing an ambiguous query, e.g.: “Why is everything so big on the screen?” (query 802 in Figure 8): “Why doesn’t my notepad work?”; or “Why are the colors so Funky’?” In some implementations, disambiguation module 350 identifies a term associated with user difficulty (e.g., “why,” “help,” “trouble,” “difficulty,” “can’t,” or “how”) or user frustration (e.g., “darn” or other words indicating frustration) as an indication that the user is experiencing difficulty operating the electronic device.

[0115] In some implementations, the user input indicating the user is having difficulty is an unintentional input made by the user, for example out of frustration or because they do not know what else to try.

[0116] In some implementations, the user input is (506) an abnormal pattern of user inputs. For example, the user turns the electronic device off and on more than a predetermined number of times over a given period of time (e.g., the user turns the device off and on more than three times in the span of 15 minutes) or turns the device on and off with an uncharacteristic user frequency (e.g., a user who usually turns the electronic device off once a day turns the device off and on four times in a single day). Or, for example, the user repeatedly makes an unproductive user input (e.g., a contact or gesture that is associated with the performance of a particular action on an electronic device operating under a first user interface setting, when the electronic device is operating under a second user interface setting in which the user input is not associated with the particular action), For example, a user repeatedly attempts to switch to a different application launch screen using a single-finger swipe gesture after a voice-over accessibility feature has been activated, requiring a three-finger swipe gesture to switch between application launch screens.

[0117] The electronic device (e.g., electronic device 104) determines (508) whether a setting was changed on the device within a predetermined time period prior to receiving the user input. For example, after detecting a qualifying user input indicating the user is having operational difficulties, the electronic device queries a data store (e.g., stored in internal memory 302 in Figure 3A and/or external restoration service(s) 122-6 in Figure 1) including records of user interface setting changes made on the device (e.g., settings history 394) against a predetermined time prior to detection of the user input (e.g., within the past hour, day, week).

[0118] As used herein, the term “predefined time” refers to either or both of a predetermined linear period of time (e.g., a number of minutes, hours, days, weeks, months, years) and a predetermined number of user sessions.

[0119] As used herein, the term “user session” refers to a continuous period of time over which a user interacts with an electronic device without interruption for more than a short period of time. For example, in some implementations, a user session is a period of time over which the user interacts (e.g., provides inputs, receives feedback, observes displayed content, etc.) with an electronic device without an interruption of at least 15 minutes. This, of course,

is merely illustrative of an exemplary implementation and is in no way meant to be limiting. For example, in some implementations, an interruption lasting as short as 30 seconds may end a user session, while in other implementations, an interruption lasting as long as a day may not end a user session.

[0120] In some implementations, the predetermined time is (510) a number of minutes (e.g., 5, 10, 15, 30, 60, 120, 240, 1440). In some implementations, the predetermined time is any linear measure of time, for example, minutes, days, weeks, months, or years.

[0121] In some implementations, the predetermined time is (512) a number of user sessions (e.g., 1, 2, 3, 4, 5, 10, 25, 50, or more user sessions).

[0122] In some implementations, the predetermined time is a combination of a number of minutes and a number of user sessions. For example, in some implementations, the predetermined time is the longer of a predetermined number of minutes and a predetermined number of user sessions (e.g., the timeframe for a qualifying change in a user setting is the past three days only if the user has completed at least three user sessions or vice versa). In some implementations, the predetermined time is the shorter of a predetermined number of minutes and a predetermined number of user sessions (e.g., the timeframe for a qualifying change in a user setting is the past three days unless a user has completed at least ten user-sessions in less than three days).

[0123] In some implementations, the device prompts (514) the user with an indication that a first setting has been changed. For example, the device audibly and/or visibly tells the user that a setting (e.g., a display setting such as the size of objects displayed on an associated display, the color scheme of objects displayed on an associated display or the language used on an associated display, an audio setting such as the master volume of the system or the volume associated with a particular audio content; or a setting that defines an action performed in response to a particular user input) has recently been changed (e.g., the a digital assistant, such as digital assistant 326, audibly says “Accessibility features were recently enabled.”)

[0124] In some implementations, the device prompts (516) the user to confirm that they are having difficulty operating the device (e.g., prompt “Are you having trouble?” 902 from device 104 in Figure 9). For example, a digital assistant (e.g., digital assistant 326) audibly and/or visually asks the user if they need assistance.

[0125] In some implementations, in response to receiving confirmation that the user is having difficulty operating the device (e.g., user response “Yes” 904 to prompt “Are you having trouble” 902 in Figure 9), the device restores (518) the changed setting to a prior setting. For example, user device 104 disables a voice-over accessibility feature in Figure 9 (e.g., as indicated by prompt “OK, I have disabled the voice-over feature.” 906) after the user confirms they are having difficulty (e.g., by response “Yes” 904).

[0126] In some implementations, where the device does not receive a confirmation that the user is having difficulty (e.g., the user responds that they are not having difficulty or the user does not respond at all), the device foregoes restoring the changed setting to a prior setting (e.g., “End” step 718 in Figure 7).

[0127] In some implementations, the device prompts (520) the user to confirm that the first setting is causing the user difficulty (e.g., prompt “Are you trying to use the notepad application?” 1006 in Figure 10). For example, a digital assistant (e.g., digital assistant 326) audibly and/or visually asks the user if a user interface setting determined to have been changed within a predefined time is causing the user difficulty.

[0128] In some implementations, in response to receiving confirmation (e.g., confirmation “Yes” 1008 in Figure 10) that the first setting (e.g., locking of a notepad application in Figure 10) is causing the user difficulty, the device restores (522) the changed setting to a prior setting (e.g., device 104 restores an unlocked setting for the notepad application in Figure 10, as indicated by prompt “OK, I have unlocked the notepad application.”)

[0129] In some implementations, in response to receiving a user input indicating that the first setting is not causing the user difficulty, the device determines (524) if a second setting was changed on the device within a predetermined time period prior to receiving the first user input. For example, after the device detects that the user is having difficulty (e.g., steps 602 and 702 in Figure 6 and 7, respectively), and identifies a first setting change occurred within a predetermined time period prior to the detection of user difficulty (e.g., steps 604 and 708 in Figures 6 and 7, respectively), and the user does not confirm that the identified setting change is causing difficulty (e.g., “No” at steps 608 and 712 in Figures 6 and 7, respectively), after optionally prompting the user for additional data (e.g., at steps 614 and 720 in Figures 6 and 7, respectively), the device determines if a second user interface setting change occurred

with the predetermined time prior to detection of the user difficulty (*e.g.*, at steps 604 and 708; $n = 2$).

[0130] In some implementations, where more than one user interface setting has been changed within the predetermined time, loops of steps 604-606-608-614 and steps 708-710-712-720 assist the device (*e.g.*, operating disambiguation module 350) in determining which setting changes are responsible for the user difficulty in an iterative fashion (*e.g.*, through a process of elimination in which the qualified user interface setting changes are queried to the user one by one, for example, starting with a user interface setting change that occurred most recently). In some implementations, where the device does not receive confirmation that the first setting change is causing user difficulty, when the device determines if a second user interface setting was changed with a predetermined time before detection of the user difficulty, the predetermined time is a second predetermined time longer than a first predetermined time (*e.g.*, the device looks farther back in settings history 394 to find a second change).

[0131] In some implementations, the device prompts (526) the user to confirm that they want to restore the changed setting to a prior setting (*e.g.*, prompt “Would you like to restore the default display settings?” 804 in Figure 8). For example, after the device detects user difficulty and identifies a user interface setting that was changed within the predetermined time prior to detection of the user difficulty, the device (*e.g.*, using digital assistant 326) audibly or visually asks the user if they want to change the setting, rather than automatically restoring a prior user interface setting, as in some implementations described herein.

[0132] In some implementations, in response to receiving confirmation that the user wants to restore the changed setting (*e.g.*, confirmation “Yes please” 806 in Figure 8), the device restores (528) the changed setting to a prior setting (*e.g.*, device 104 restores default settings in Figure 8 after confirmation 806, as indicated by prompt “OK, the settings have been restored” 808).

[0133] Upon a determination that a first setting was changed within the predetermined time period prior to receiving the user input, the device restores (530) the changed setting to a prior setting (*e.g.*, device 104: restores default display settings in Figure 8, as indicated by prompt 808; disables a voice-over feature in Figure 9, as indicated by prompt 906; and unlocks a notepad application in Figure 10, as indicated by prompt 1010).

[0134] In some implementations, the prior setting (e.g., the user interface setting being restored by the device when corresponding criteria are met) is a default setting (532). For example, the user interface setting is returned to, a “factory setting” or a default setting previously determined by the user (e.g., when first setting-up the electronic device or otherwise specifically configuring, a default user interface setting that may or may not override and/or replace a factory setting).

[0135] In some implementations, the prior setting is a setting that immediately preceded the setting change (534). In some implementations, rather than restoring a Factory or default user interface setting, as in accordance with some implementations described herein, the device (e.g., with diagnostics module 366, restoration module 390, or diagnostics module 366 in conjunction with restoration module 390) queries a data store (e.g., settings history 394) to determine the setting that immediately preceded the identified setting change and restores (e.g., using restoration module 390) the prior setting. For example, where a factory Font display setting is 10 Pt, a user customizes the user display by increasing the Font display setting to 14 pt at a first time. At a later time, the user inadvertently changes the font display setting to 48 pt, rendering the user display difficult to navigate. Upon detection that the user is experiencing difficulty with the increased font size, the device restores the font display setting to 14 pt.

[0136] In some implementations, the device explains (536) to the user how the setting was changed. For example, the device audibly or visually explains to user which steps were taken, or would be taken, to change the user interface setting that caused user difficulty, enabling the user to avoid inadvertently changing the setting again (e.g., so that the user doesn’t accidentally activate an undesired accessibility feature).

[0137] In some implementations, the device explains (538) to the user how to restore the changes setting to a prior setting (e.g., the device audibly or visually explains to the user how to disable an activated accessibility feature). For example, a digital assistant (e.g., digital assistant 326) audibly and/or visually walks the user through the steps necessary to restore a prior user interface setting under the changed user interface environment that caused the user difficulty. In some implementations, the device (e.g., using digital assistant 326) explains how to restore a prior user interface setting before restoring (530) the prior setting. In some implementations, the device (e.g., using digital assistant 326) explains how to restore a prior user interface setting while restoring (530) the prior setting (e.g., in a tutorial). In some

implementations, the device (e.g., using digital assistant 326) explains how to restore a prior user interface setting after restoring (530) the prior setting.

[0138] In some implementations of the methods, systems, and devices described above, prompting the user includes (540) the use of a text-to-speech function. In some implementations, prompting the user also includes the use of visually prompting the user.

[0139] In some implementations, one or more of the methods and systems described above for diagnosing an inadvertent activation of a user interface setting and restoring a prior user interface setting can be disabled by a user, e.g., through an application specific setting in application specific settings module 372. For example, it may be desirable to disable implementations where the device initiates the method upon receiving a user input indicating the user is having difficulty operating the device (e.g., method 700), when the device is operated by an elderly person, young child, or person with a disability who may appear to be having difficulty with an accessibility feature, but are actually not able to operate the device well in any setting.

[0140] It should be understood that the particular order in which the operations in Figures 5A-5B have been described is merely exemplary and is not intended to indicate that the described order is the only order in which the operations could be performed. One of ordinary skill in the art would recognize various ways to reorder the operations described herein.

Exemplary Methods

[0141] Now that an overview of the methods and systems for diagnosing inadvertent activation of a user interface setting, such as an accessibility feature, and for restoring a prior user interface setting have been presented, exemplary methods will be presented in conjunction with Figures 6-10. Figures 8-10 illustrate an exemplary electronic devices (e.g., user device 104) displaying graphical user interfaces (GUIs) 800, 900, and 1000, respectively, on touch screen 112. GUIs 800, 900, and 1000 include a dialog between the user (e.g., as indicated by text bubbles originating from the right side of touch screen 112) and a voice-based digital assistant (e.g., as indicated by text bubbles originating from the left side of touch screen 112).

[0142] Figures 6 and 8 illustrate exemplary method 600 for diagnosing inadvertent activation of a user interface setting (e.g., an accessibility feature) and for restoring a prior

user interface setting, in accordance with some implementations, where the user initiates the method.

[0143] In step 602, the user provides a request For help operating an electronic device (e.g., electronic device 104) to a voice-based digital assistant (e.g., digital assistant 326) associated with the electronic device. In some implementations, the request for help need not necessarily include the word “help” or any synonym thereof, any indication that the user is experiencing difficulty operating the device and that the difficulty may relate to a user-interface setting (e.g., a display, audio, or operational user interface setting) is sufficient to activate execution of method 600. For example, user input “Why is everything so big on the screen?” 802 is sufficient at least because it indicates the user is having difficulty operating the device (e.g., the user has to ask “why?”), and it indicates that the difficulty may relate to a user interface setting (e.g., the font and/or icon display size).

[0144] In some implementations, the voice-based digital assistant is accessible to the user through a default mechanism that cannot be disabled by a change to any user interface setting, including without limitation, accessibility Features, display settings, operational settings, volume settings, etc. In some implementations, the voice-based digital assistant is activated by a tactile input on the electronic device, e.g., by a double tap input on home button 810 of user device 104 in Figure 8. In some implementations, the voice-based digital assistant is activated by a voice input or signal, e.g., by use of a phrase recognized by the voice-based digital assistant such as “Hi Siri.”

[0145] In step 604, the electronic device determines whether a user interface setting was recently changed on the device (e.g., within a predetermined time prior to receiving the user input in step 602). In some implementations, the device queries a data store (e.g., settings history 394 stored locally in memory 302 of the device or stored externally such as by a restoration service 122-6) to determine if a user interface setting change was made within a predetermined time period (e.g., within a number of minutes and/or user sessions prior to the detection of the user difficulty).

[0146] In some implementations where the outcome of step 604 is “No,” the device prompts the user for additional information on their operational difficulty. Additional data received from the user in response is used by the device (e.g., using disambiguation module 350) to identify whether a second user interface setting was changed within a predetermined time of receiving the user input in step 602, in a second iteration of step 604. In some

implementations, where the number of iterations (n) of step 604 is less than a threshold number (x), additional inputs from the user indicating that an identified user interface setting change is not causing user difficulty in step 604 cause the device to prompt the user for additional information regarding their operational difficulty in an additional iteration of step 614. In some implementations, where the number of iterations (n) is equal to a threshold number (x), the device prompts the user to contact a customer service (e.g., by phone, email, text message, or SMS) in step 616, providing the user with an opportunity for additional technological support and avoiding an infinite loop. In some implementations, where the user responds affirmatively to the prompt provided in step 616, the device initiates contacting a customer service (e.g., by placing a phone call, sending an email, or sending a text message or SMS) in step 618. In some implementations, where the user responds negatively to the prompt provided in step 616, the device ends method 600 in step 620.

[0147] In some implementations, where the outcome of step 604 is “Yes,” an identified user interface setting was recently changed, the device asks the user to confirm that the identified user interface setting change is causing the user difficulty in step 606. In some implementations, asking for user confirmation includes asking if the user wants to restore a previous user interface setting. For example, prompt “Would you like to restore the default display settings?” 804. In some implementations, asking the user to confirm if an identified user interface setting is causing difficulty and asking the user to confirm if they wish to restore a prior user interface setting are performed in separate steps (e.g., they require separate confirmations from the user). In some implementations, only one of asking the user to confirm if an identified user interface setting is causing difficulty and asking the user to confirm if they wish to restore a prior user interface setting is performed.

[0148] In some implementations, the device receives confirmation that the identified user interface setting change is causing the user difficulty, and/or, in some implementations, confirmation to restore a prior user interface setting in step 608. For example, confirmation “Yes please” 806 from the user. In response, the device restores a prior user interface setting in step 610 (e.g., disables an accessibility feature). For example, user device 104 restores a default display setting in response to user response 806, as indicated by prompt “OK, the settings have been restored” 808, in Figure 8.

[0149] In some implementations, steps 606 and 608 are not performed, and in response to a determination that a first setting was changed within the predetermined time period prior to

receiving the user input in step, the device restores a prior user interface setting in step 610 (e.g., disables an accessibility feature).

[0150] In some implementations, before, during, or after restoring a prior user interface setting in step 610, the device educates the user on how to avoid changing the identified user interface setting again (e.g., by educating the user on how the user interface setting was changed) in step 612. In some implementations, before, during, or after restoring a prior user interface setting in step 610, the device additionally or alternatively educates the user on how to restore a prior user interface setting (e.g., how to disable an accessibility feature).

[0151] In some implementations, where the outcome of step 608 is “No,” the device prompts the user for additional information on their operational difficulty. Additional data received from the user in response is used by the device (e.g., using disambiguation module 350) to identify whether a second user interface setting was changed within a predetermined time of receiving the user input in step 602, in a second iteration of step 604. In some implementations, where the number of iterations (n) of step 608 is less than a threshold number (y), additional inputs from the user indicating that an identified user interface setting change is not causing user difficulty in step 604 cause the device to prompt the user for additional information regarding their operational difficulty in an additional iteration of step 614. In some implementations, where the number of iterations (n) is equal to a threshold number (y), the device prompts the user to contact a customer service (e.g., by phone, email, text message, or SMS) in step 616, providing the user with an opportunity for additional technological support and avoiding an infinite loop. In some implementations, where the user responds affirmatively to the prompt provided in step 616, the device initiates contacting a customer service (e.g., by placing a phone call, sending an email, or sending a text message or SMS) in step 618. In some implementations, where the user responds negatively to the prompt provided in step 616, the device ends method 600 in step 620.

[0152] Figures 7, 9, and 10 illustrate exemplary method 700 for diagnosing inadvertent activation of a user interface setting (e.g., an accessibility feature) and for restoring a prior user interface setting, in accordance with some implementations, where the electronic device initiates the method.

[0153] In step 702, an electronic device (e.g., electronic device 104) detects a pattern of user inputs indicating that a user is experiencing difficulty operating the electronic device. For example, in some implementations, the device detects an abnormal pattern of user inputs

(e.g., corresponding to the user: repeatedly turning the device off and on; repeatedly logging off and on to a workspace; repeatedly executing a touch or gesture not associated with an operation under a particular user interface environment; etc.) indicating the user is experiencing difficulty.

[0154] In some implementations, in response to the detection of an apparent user difficulty, the device asks the user if they are experiencing difficulty (e.g., using a voice-based digital assistant such as digital assistant 326) operating the electronic device in step 704. For example, user device provides audible and/or visible prompts “Are you having trouble?” 902 and 1002 in Figures 9 and 10, respectively (e.g., using voice-based digital assistant 326).

[0155] In some implementations, the device detects a user response indicating that they are experiencing difficulty operating the device (e.g., user responses “Yes” 904 and 1004 in Figures 9 and 10, respectively). In some implementations, where the device does not receive a user response confirming the operational difficulty in step 706, the device ends method 700 in step 718.

[0156] In step 708, the electronic device determines whether a user interface setting was recently changed on the device (e.g., within a predetermined time prior to receiving the user input in step 702). In some implementations, the device queries a data store (e.g., settings history 394 stored locally in memory 302 of the device or stored externally such as by a restoration service 122-6) to determine if a user interface setting change was made within a predetermined time period (e.g., within a number of minutes and/or user sessions prior to the detection of the user difficulty).

[0157] In some implementations where the outcome of step 708 is “No,” the device prompts the user For additional information on their operational difficulty. Additional data received from the user in response is used by the device (e.g., using disambiguation module 350) to identify whether a second user interface setting was changed within a predetermined time of receiving the user input in step 702, in a second iteration of step 708. In some implementations, where the number of iterations (n) of step 708 is less than a threshold number (x), additional inputs from the user indicating that an identified user interface setting change is not causing user difficulty in step 708 cause the device to prompt the user for additional information regarding their operational difficulty in an additional iteration of step 720. In some implementations, where the number of iterations (n) is equal to a threshold

number (x), the device prompts the user to contact a customer service (e.g., by phone, email, text message, or SMS) in step 722, providing the user with an opportunity for additional technological support and avoiding an infinite loop. In some implementations, where the user responds affirmatively to the prompt provided in step 722, the device initiates contacting a customer service (e.g., by placing a phone call, sending an email, or sending a text message or SMS) in step 724. In some implementations, where the user responds negatively to the prompt provided in step 722, the device ends method 700 in step 726.

[0158] In some implementations, where the outcome of step 708 is “Yes,” an identified user interface setting was recently changed, the device asks the user to confirm that the identified user interface setting change is causing the user difficulty in step 710. For example, prompt “Are you trying to use the notepad application?” 1006. In some implementations, asking for user confirmation includes asking if the user wants to restore previous user interface setting. In some implementations, asking the user to confirm if an identified user interface setting is causing difficulty and asking the user to confirm if they wish to restore a prior user interface setting are performed in separate steps (e.g., they require separate confirmations from the user). In some implementations, only one of asking the user to confirm if an identified user interface setting is causing difficulty and asking the user to confirm if they wish to restore a prior user interface setting is performed.

[0159] In some implementations, the device receives confirmation that the identified user interface setting change is causing the user difficulty, and/or, in some implementations, confirmation to restore a prior user interface setting in step 712. For example, confirmation “Yes” 1008 from the user. In response, the device restores a prior user interface setting in step 714 (e.g., disables an accessibility feature). For example, user device 104 unlocks an application in response to user response 1008, as indicated by prompt “OK, I have unlocked the notepad application” 1010, in Figure 10.

[0160] In some implementations, steps 710 and 712 are not performed, and in response to a determination that a first setting was changed within the predetermined time period prior to receiving the user input in step, the device restores a prior user interface setting in step 714 (e.g., disables an accessibility feature). For example, user device 104 disables an accessibility feature, as indicated by prompt “OK, I have disabled the voice-over feature” 906, in Figure 9.

[0161] In some implementations, before, during, or after restoring a prior user interface setting in step 714, the device educates the user on how to avoid changing the identified user interface setting again (e.g., by educating the user on how the user interface setting was changed) in step 716. In some implementations, before, during, or after restoring a prior user interface setting in step 714, the device additionally or alternatively educates the user on how to restore a prior user interface setting (e.g., how to disable an accessibility feature) in step 716.

[0162] In some implementations, where the outcome of step 712 is “No,” the device prompts the user for additional information on their operational difficulty. Additional data received from the user in response is used by the device (e.g., using disambiguation module 350) to identify whether a second user interface setting was changed within a predetermined time of receiving the user input in step 702, in a second iteration of step 708. In some implementations, where the number of iterations (n) of step 712 is less than a threshold number (y), additional inputs from the user indicating that an identified user interface setting change is not causing user difficulty in step 712 cause the device to prompt the user for additional information regarding their operational difficulty in an additional iteration of step 720. In some implementations, where the number of iterations (n) is equal to a threshold number (y), the device prompts the user to contact a customer service (e.g., by phone, email, text message, or SMS) in step 722, providing the user with an opportunity for additional technological support and avoiding an infinite loop. In some implementations, where the user responds affirmatively to the prompt provided in step 722, the device initiates contacting a customer service (e.g., by placing a phone call, sending an email, or sending a text message or SMS) in step 724. In some implementations, where the user responds negatively to the prompt provided in step 722, the device ends method 700 in step 726.

[0163] The foregoing description, for purpose of explanation, has been described with reference to specific implementations. However, the illustrative discussions above are not intended to be exhaustive or to limit the disclosed implementations to the precise Forms disclosed. Many modifications and variations are possible in view of the above teachings. The implementations were chosen and described in order to best explain the principles and practical applications of the disclosed ideas, to thereby citable others skilled in the art to best utilize them with various modifications as are suited to the particular use contemplated.

[0164] It will be understood that, although the terms “first,” “second,” etc. may be used herein to describe various elements, these elements should not be limited by these terms.

These terms are only used to distinguish one element from another. For example, a first sound detector could be termed a second sound detector, and, similarly, a second sound detector could be termed a first sound detector, without changing the meaning of the description, so long as all occurrences of the “first sound detector” are renamed consistently and all occurrences of the “second sound detector” are renamed consistently. The first sound detector and the second sound detector are both sound detectors, but they are not the same sound detector.

[0165] The terminology used herein is for the purpose of describing particular implementations only and is not intended to be limiting of the claims. As used in the description of the implementations 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 “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.

[0166] As used herein, the term “if” may be construed to mean “when” or “upon” or “in response to determining” or “in accordance with a determination” or “in response to detecting,” that a stated condition precedent is true, depending on the context. Similarly, the phrase “if it is determined [that a stated condition precedent is true]” or “if [a stated condition precedent is true]” or “when [a stated condition precedent is true]” may be construed to mean “upon determining” or “upon a determination that” or “in response to determining” or “in accordance with a determination” or “upon detecting” or “in response to detecting” that the stated condition precedent is true, depending on the context.

What is claimed is:

1. A method for diagnosing inadvertent activation of a user interface setting, the method comprising:
 - at an electronic device comprising a processor and memory storing instructions for execution by the processor:
 - receiving a user input indicating that the user is having difficulty operating the electronic device;
 - determining whether a setting was changed on the device within a predetermined time period prior to receiving the user input; and
 - upon a determination that a first setting was changed within the predetermined time period prior to receiving the user input, restoring the changed setting to a prior setting.
2. The method of claim 1, wherein the user input is a user voice input.
3. The method of claim 1 or 2, wherein the user input is an abnormal pattern of user inputs.
4. The method according to any one of claims 1-3, wherein the predetermined time is a number of minutes.
5. The method according to any one of claims 1-3, wherein the predetermined time is a number of user sessions.
6. The method according to any one of claims 1-5, wherein the prior setting is a default setting.
7. The method according to any one of claims 1-5, wherein the prior setting is a setting that immediately preceded the setting change.
8. The method according to any one of claims 1-7, further comprising:
 - prior to restoring the changed setting, prompting the user with an indication that the first setting has been changed.
9. The method according to any one of claims 1-8, further comprising:

prior to restoring the changed setting, prompting the user to confirm that they are having difficulty operating the device; and

in response to receiving confirmation that the user is having difficulty operating the device, restoring the changed setting to a prior setting.

10. The method according to any one of claims 1-9, further comprising:

prior to restoring the changed setting, prompting the user to confirm that the first setting is causing the user difficulty; and

in response to receiving confirmation that the first setting is causing the user difficulty, restoring the changed setting to a prior setting.

11. The method of claim 10, further comprising:

in response to receiving a user input indicating that the first setting is not causing the user difficulty, determining if a second setting was changed on the device within a predetermined time period prior to receiving the first user input.

12. The method according to any one of claims 1-11, further comprising:

prior to restoring the changed setting, prompting the user to confirm that they want to restore the changed setting to a prior setting; and

in response to receiving confirmation that the user wants to restore the changed setting, restoring the changed setting to a prior setting.

13. The method according to any one of claims 8-12, wherein prompting the user includes the use of a text-to-speech function.

14. The method according to any one of claims 1-13, further comprising:

explaining to the user how the display feature was enabled.

15. The method according to any one of claims 1-14, further comprising:

explaining to the user how to restore the changed setting to a prior setting.

16. An electronic device, comprising:

a processor;

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 for performing any of the methods of claims 1-15.

17. A computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by an electronic device, cause the device to perform any of the methods of claims 1-15.

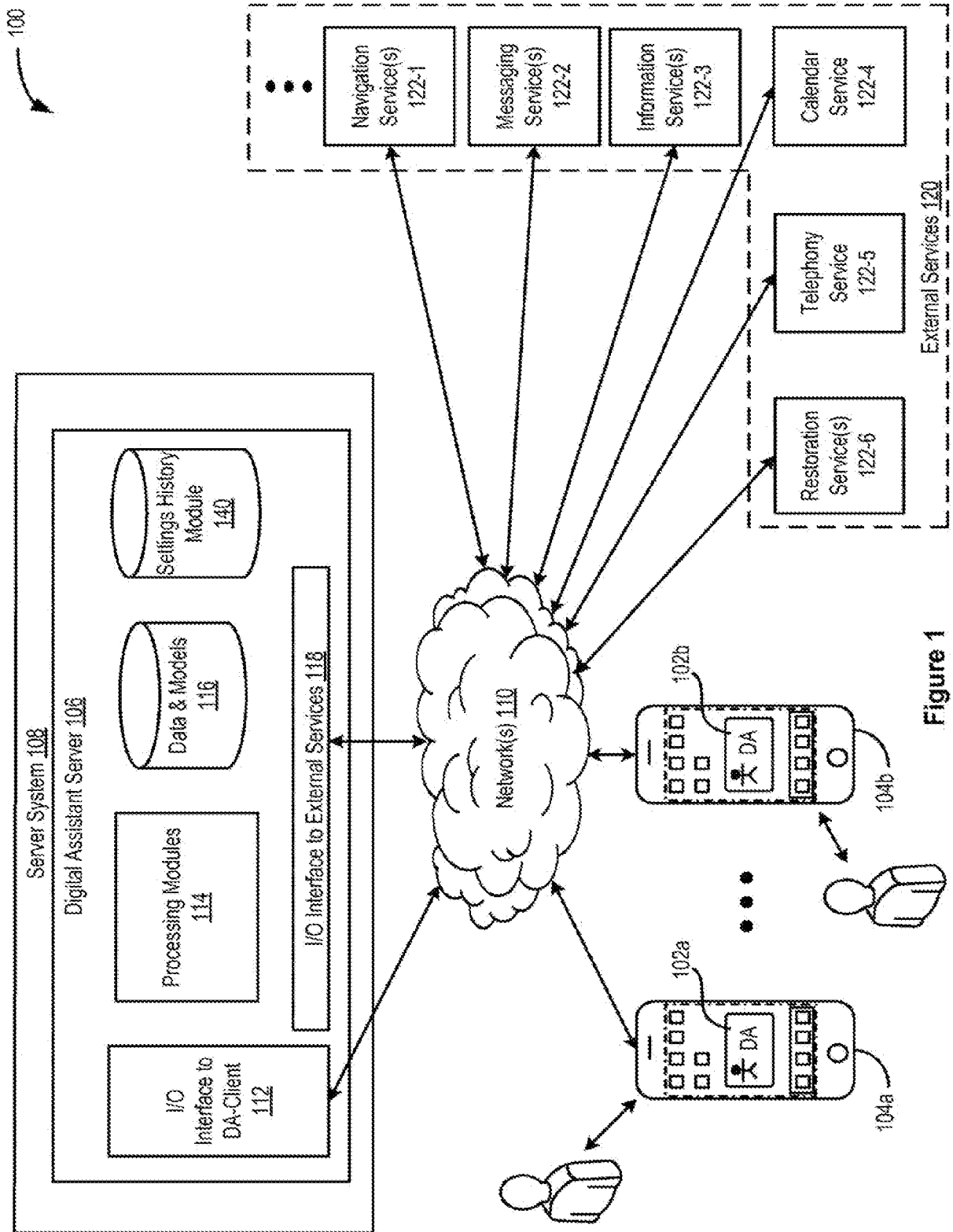


Figure 1

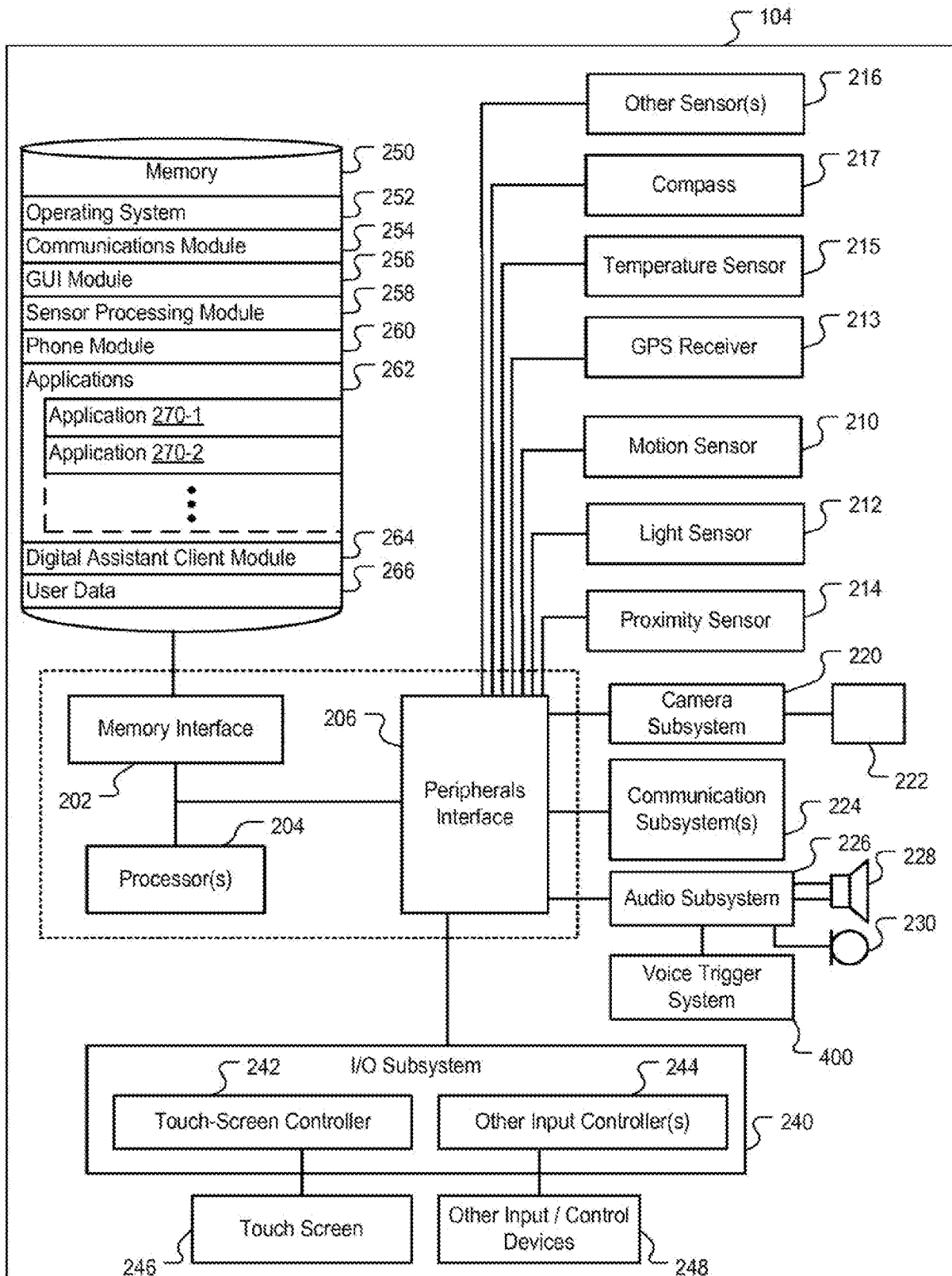


Figure 2

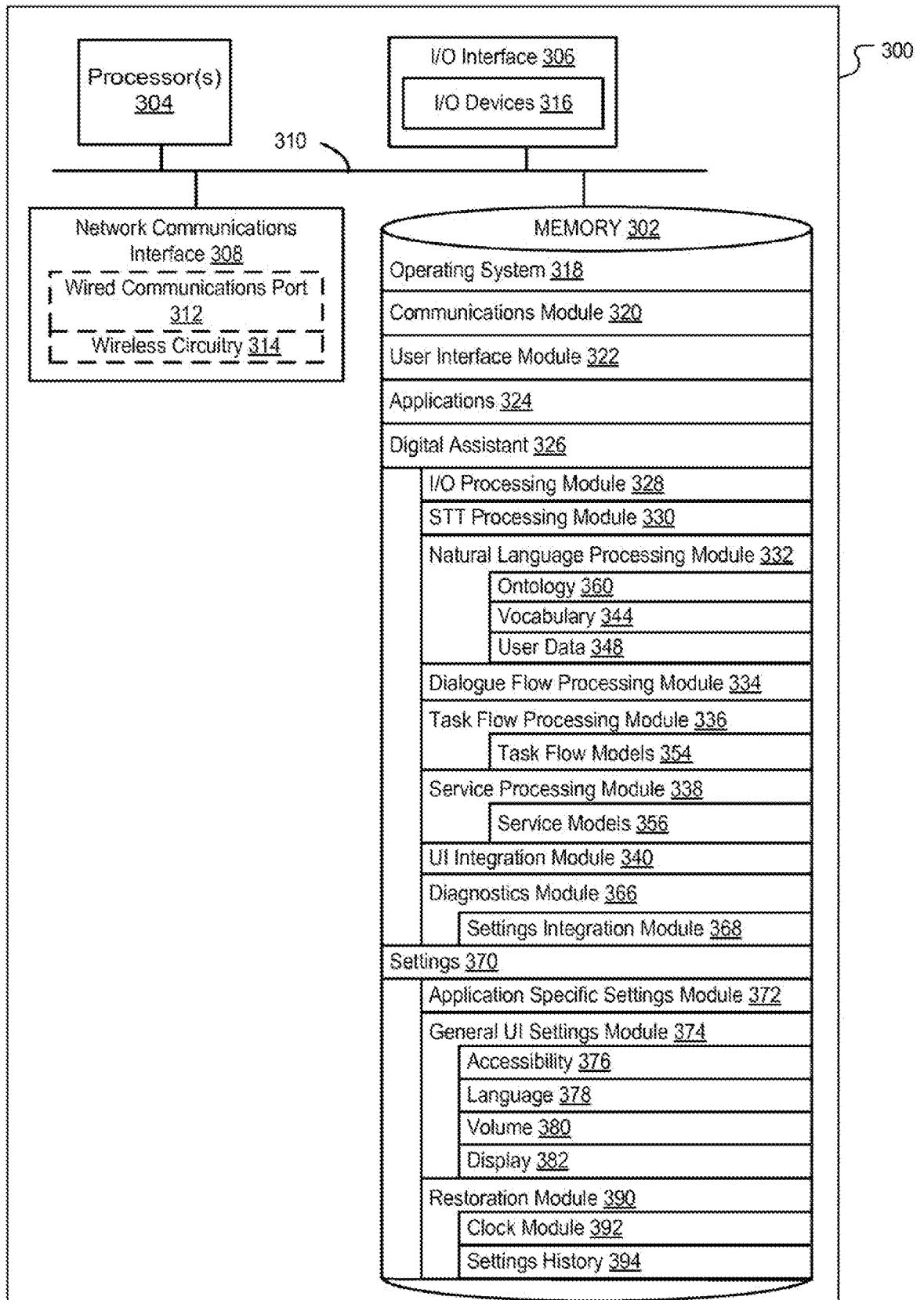


Figure 3A

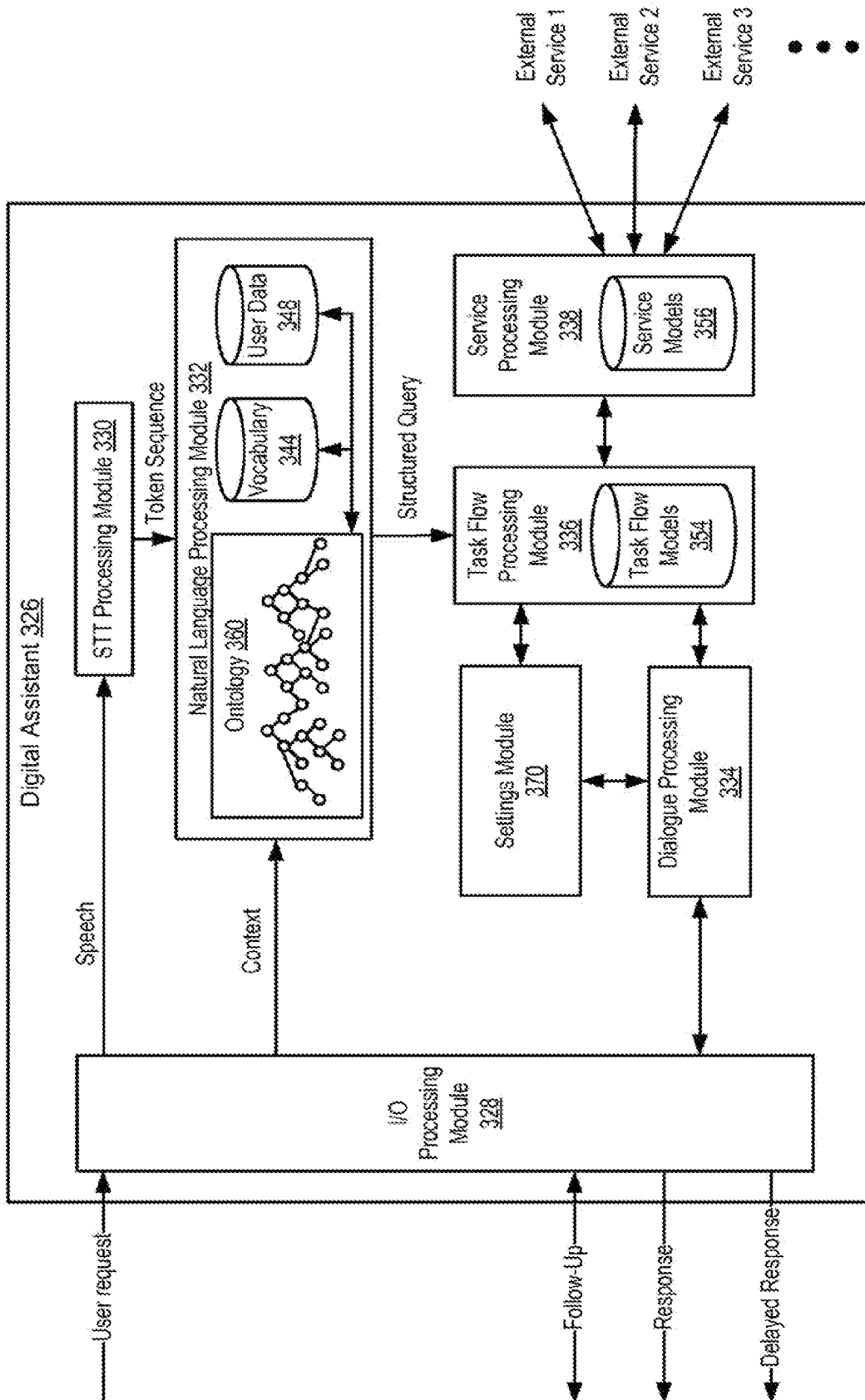


Figure 3B

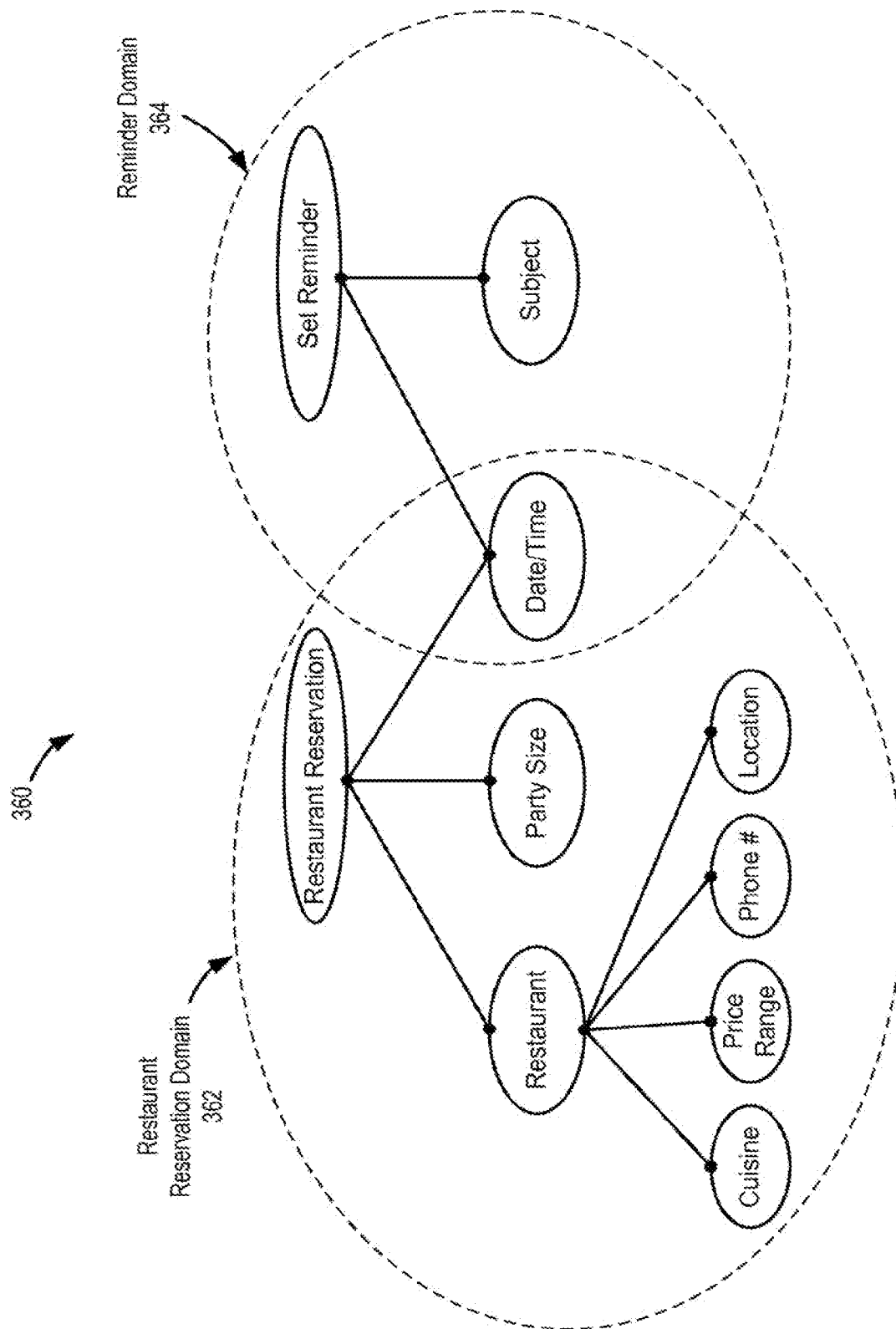


Figure 3C

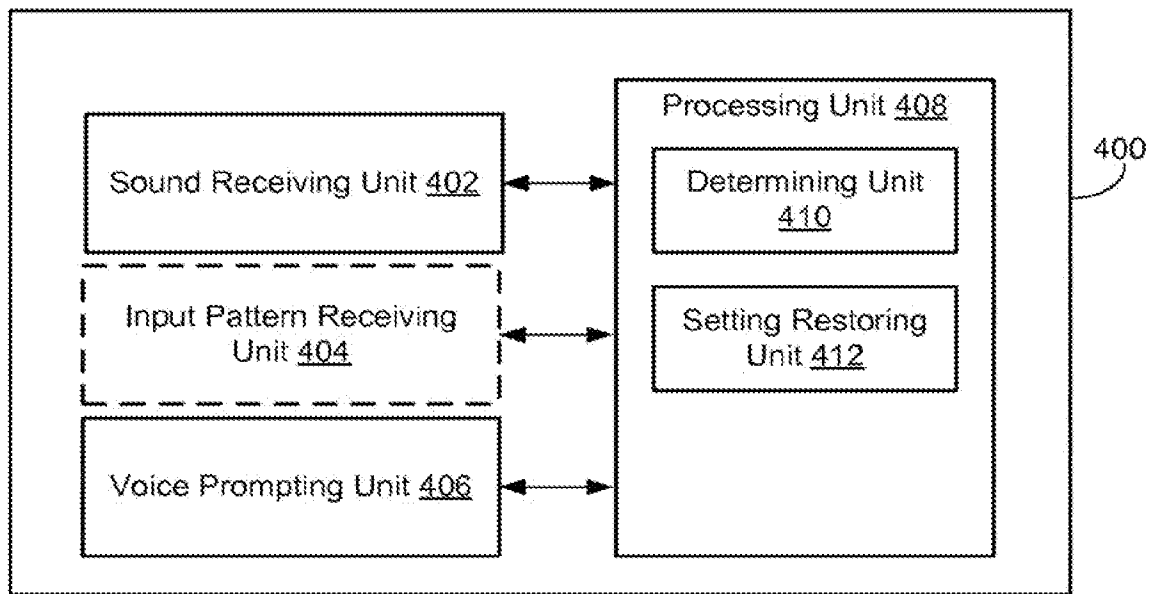


Figure 4

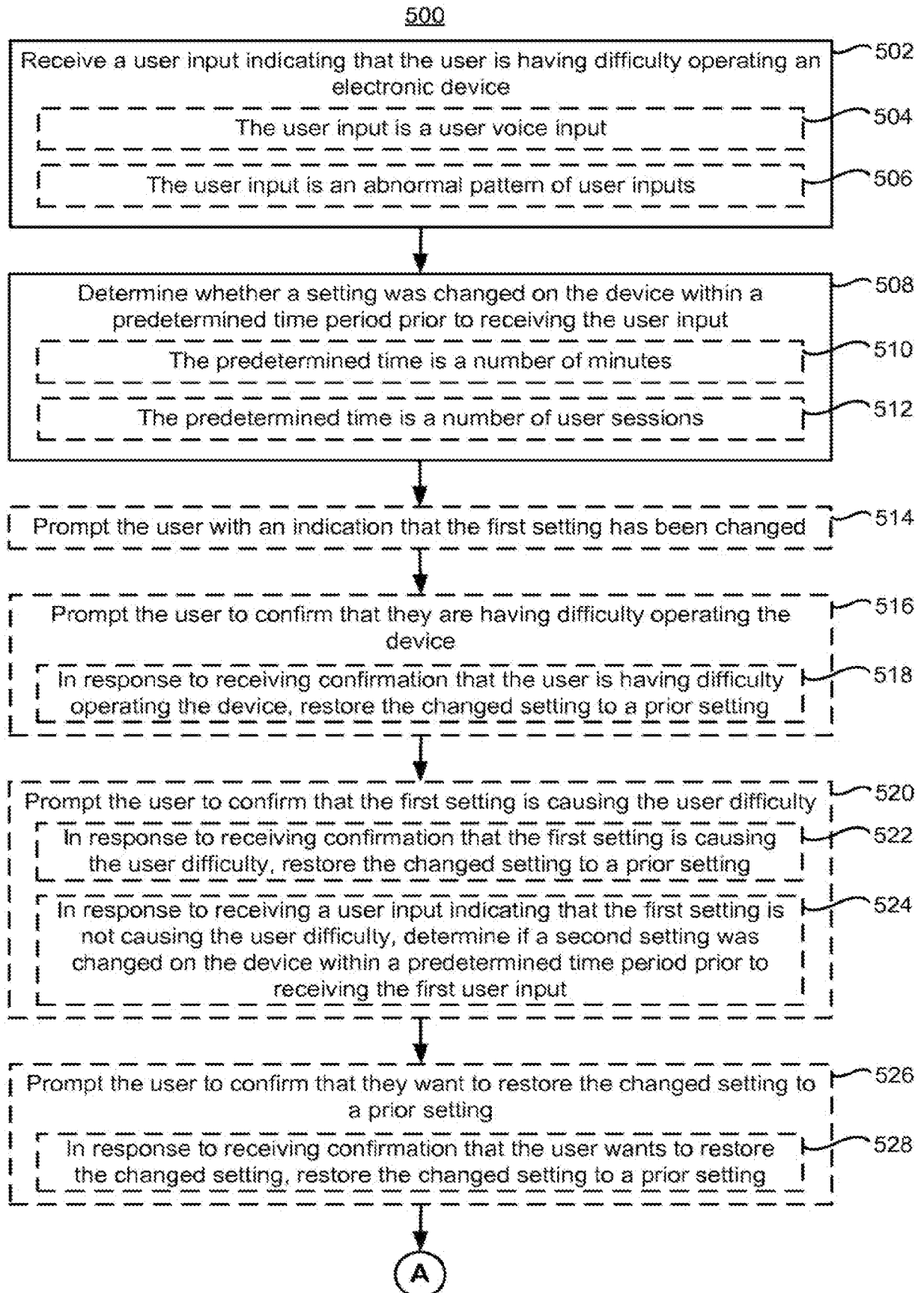


Figure 5A

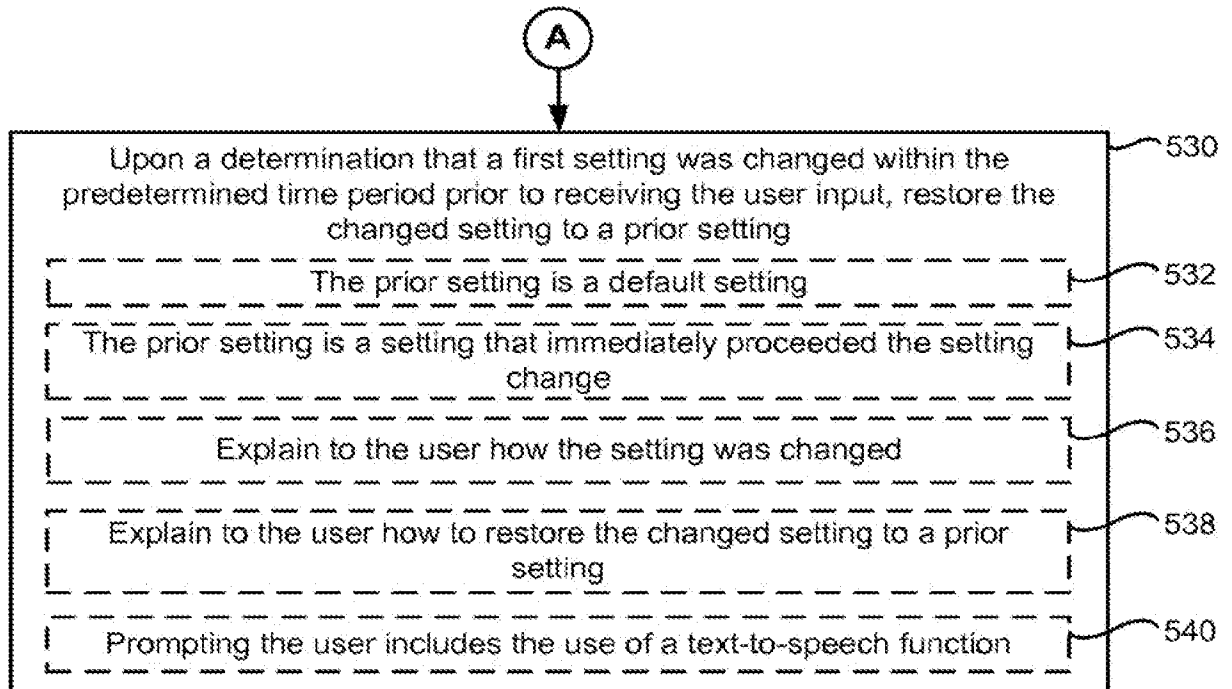


Figure 5B

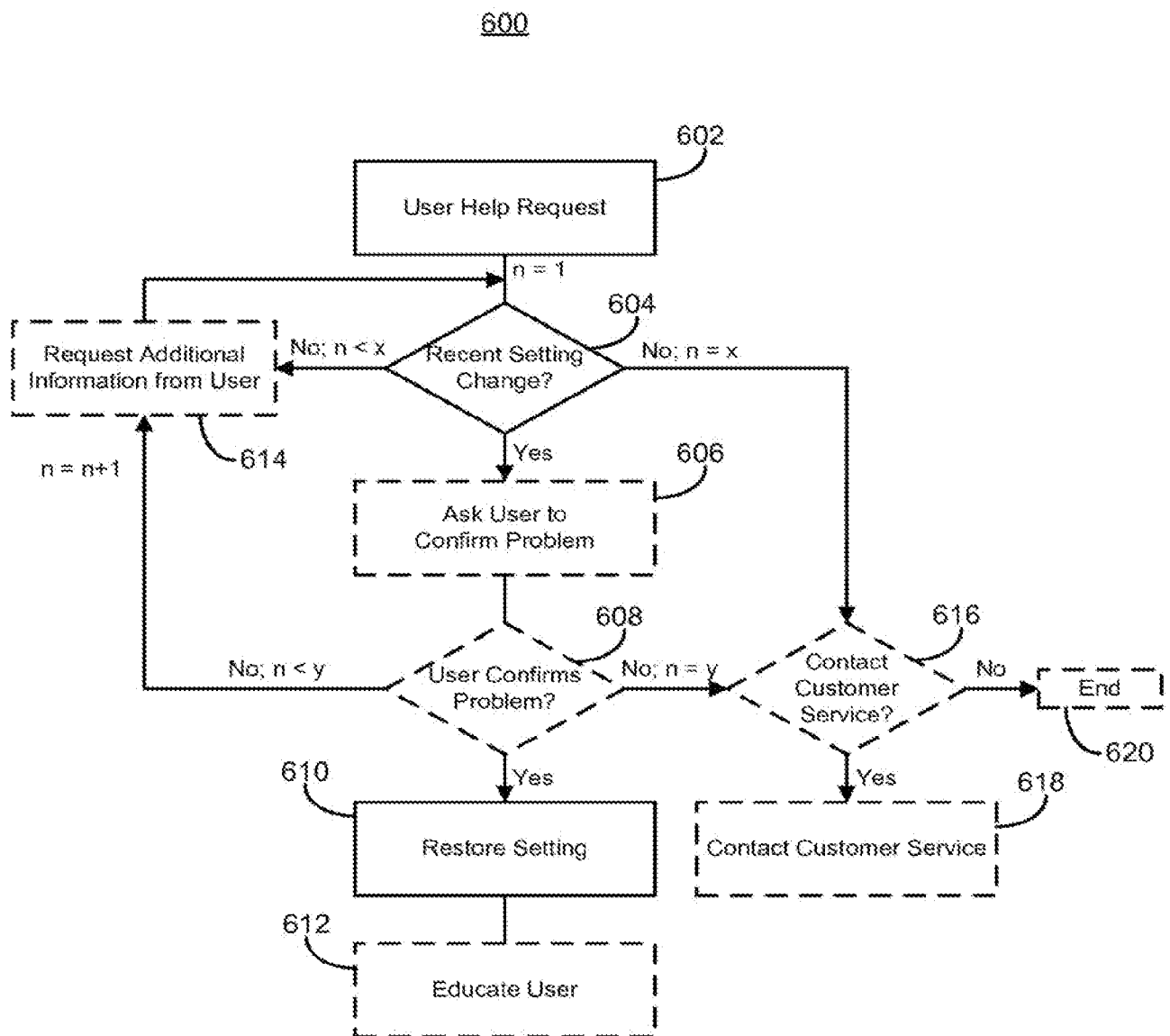


Figure 6

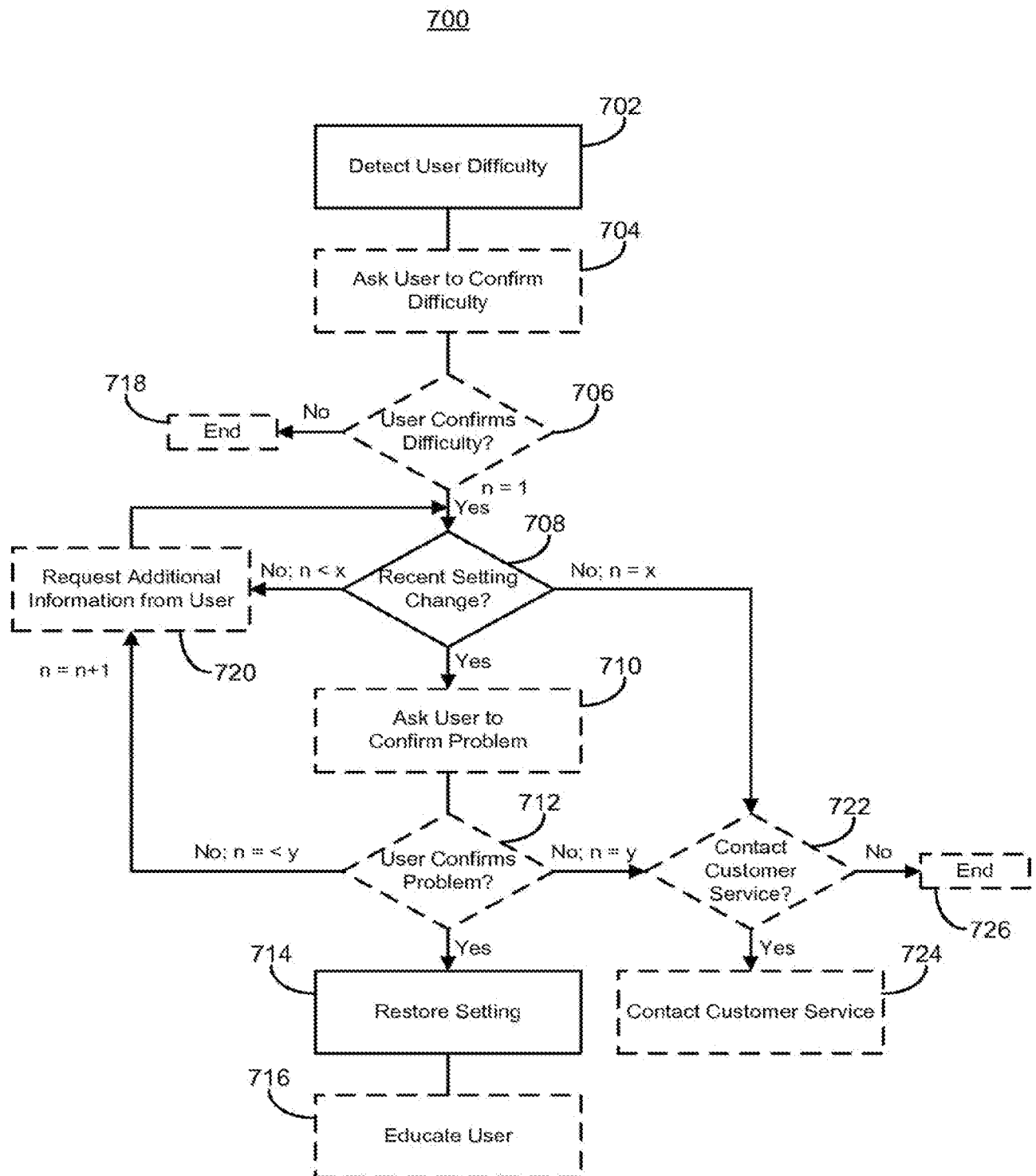


Figure 7

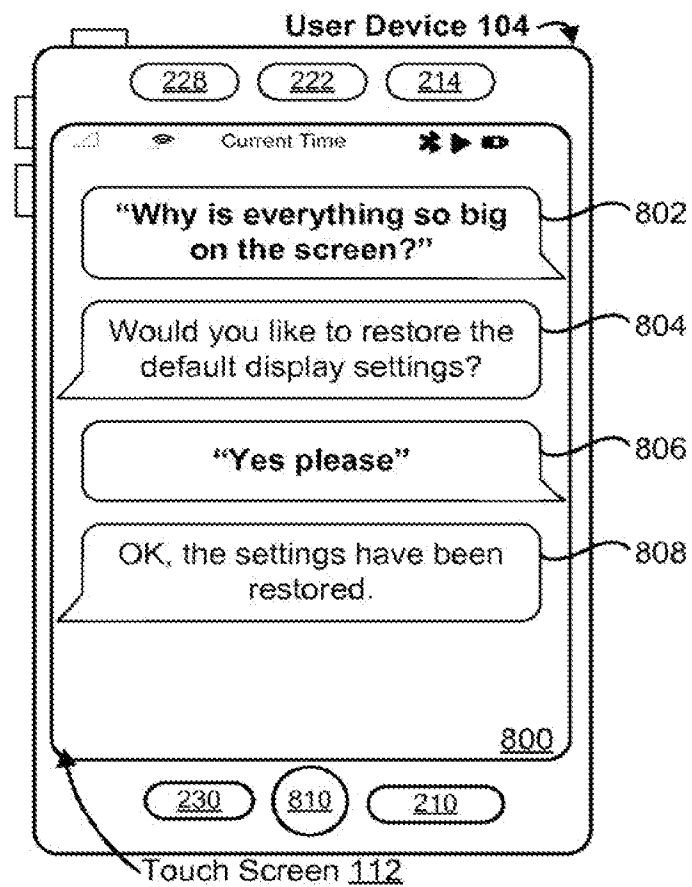


Figure 8

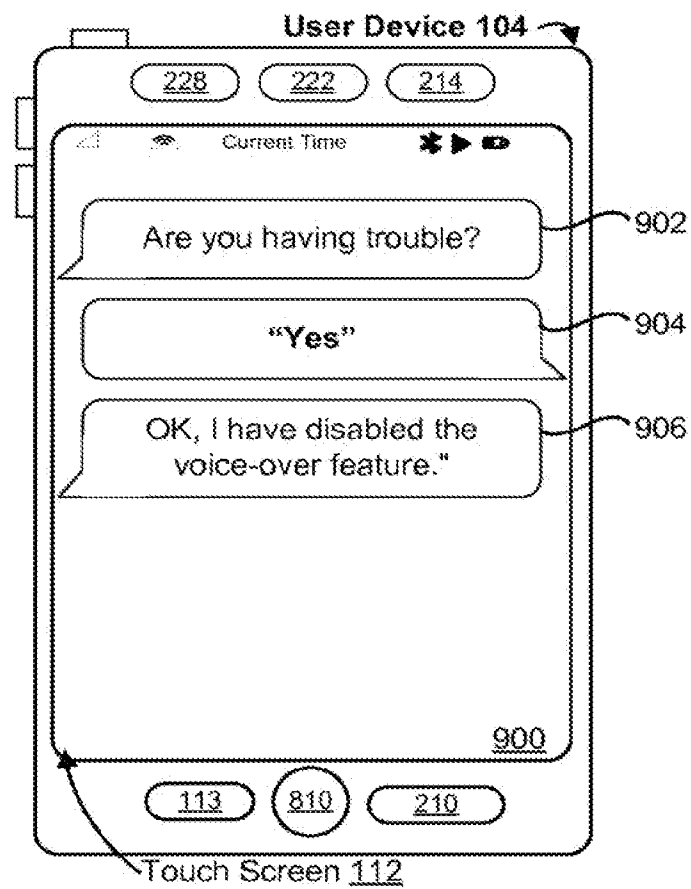


Figure 9

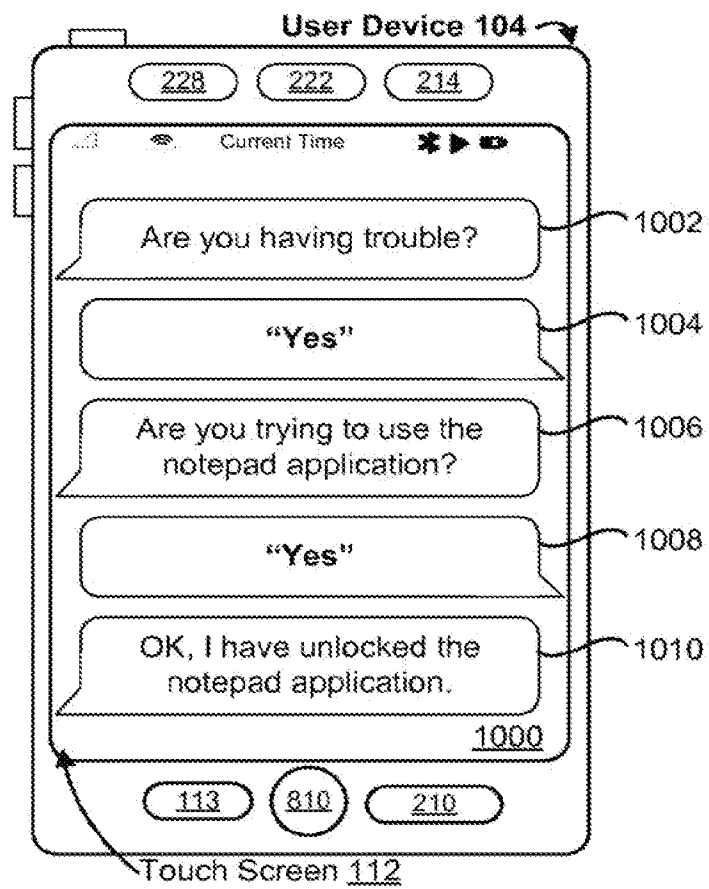


Figure 10

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2014/026871

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F3/16 G06F9/445
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F G10L H04M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2005/108642 A1 (SINCLAIR ROBERT E II [US] SINCLAIR II ROBERT E [US]) 19 May 2005 (2005-05-19) paragraph [0048] - paragraph [0069] figures 1,4	1-17
A	WO 2012/155079 A2 (JOHNSON CONTROLS TECH CO [US]; FAY WILLIAM [US]; DOUTHITT BRIAN L [US]) 15 November 2012 (2012-11-15) figure 3 paragraph [0050] - paragraph [0053]	1-17
A	US 2004/260718 A1 (FEDOROV VLADIMIR D [US]) 23 December 2004 (2004-12-23) paragraph [0025] - paragraph [0041]	1-17



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

16 July 2014

Date of mailing of the international search report

23/07/2014

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Alliot, Sylvain

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2014/026871

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 2005108642	A1	19-05-2005	CN 1826636 A	30-08-2006
			EP 1665235 A2	07-06-2006
			JP 2007511833 A	10-05-2007
			KR 20060114280 A	06-11-2006
			US 2005108642 A1	19-05-2005
			WO 2005057327 A2	23-06-2005

WO 2012155079	A2	15-11-2012	EP 2707872 A2	19-03-2014
			WO 2012155079 A2	15-11-2012

US 2004260718	A1	23-12-2004	NONE	
