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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 December 2014 (11.12.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/197336 A1

(51) International Patent Classification:
G10L 15/22 (2006.01) *H04M 1/27* (2006.01)

(21) International Application Number:
PCT/US2014/040401

(22) International Filing Date:
30 May 2014 (30.05.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/832,751 7 June 2013 (07.06.2013) US

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

(72) Inventor: **SINHA, Anoop, K.**; c/o Apple Inc., 1 Infinite Loop, MS 36-2PAT, 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))*

(54) Title: SYSTEM AND METHOD FOR DETECTING ERRORS IN INTERACTIONS WITH A VOICE-BASED DIGITAL ASSISTANT

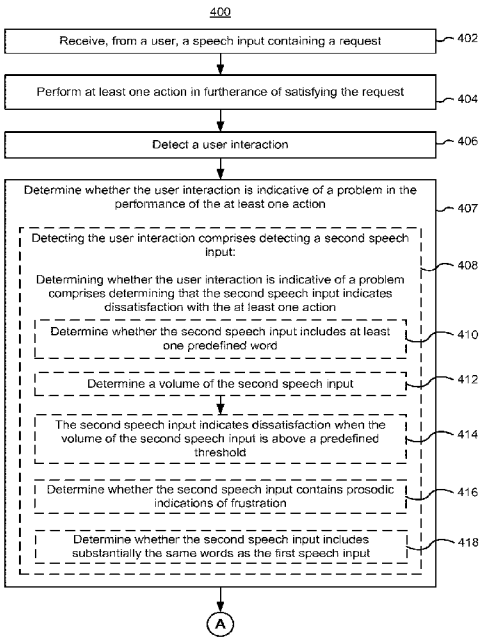


FIG. 4A

(57) Abstract: The method is performed at an electronic device with one or more processors and memory storing one or more programs for execution by the one or more processors. A speech input containing a request is received from a user. At least one action in furtherance of satisfying the request is performed. A user interaction is detected, such as a speech input to a digital assistant or a physical interaction with a device. It is determined whether the user interaction is indicative of a problem in the performing of the at least one action. Upon determining that the user interaction is indicative of a problem, information relating to the request is stored in a repository for error analysis.

WO 2014/197336 A1

SYSTEM AND METHOD FOR DETECTING ERRORS IN INTERACTIONS WITH A VOICE-BASED DIGITAL ASSISTANT

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims priority from U.S. Provisional Serial No. 61/832,751, filed on June 7, 2013, entitled SYSTEM AND METHOD FOR DETECTING ERRORS IN INTERACTIONS WITH A VOICE-BASED DIGITAL ASSISTANT, 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 systems and methods for detecting errors in speech interactions with a digital assistant.

BACKGROUND

[0003] Just like human personal assistants, digital assistants or virtual assistants can perform requested tasks and provide requested advice, information, or services. An assistant's ability to fulfill a user's request is dependent on the assistant's correct comprehension of the request or instructions. Recent advances in natural language processing have enabled users to interact with digital assistants using natural language, in spoken or textual forms, rather than employing a conventional user interface (e.g., menus or programmed commands). Such digital assistants can interpret the user's input to infer the user's intent, translate the inferred intent into actionable tasks and parameters, execute operations or deploy services to perform the tasks, and produce outputs that are intelligible to the user. Ideally, the outputs produced and the tasks performed by a digital assistant should fulfill the user's intent expressed during the natural language interaction between the user and the digital assistant. However, digital assistants will, from time to time, produce erroneous outputs and/or perform erroneous tasks in response to a user input, which can be irritating for users, and can make the digital assistant appear incompetent or unsophisticated.

[0004] Also, digital assistants that interact with users via speech inputs and outputs typically employ speech-to-text processing techniques to convert speech inputs to textual forms that can be further processed, and speech synthesis techniques to convert textual outputs to speech. In both cases, accurate conversion between speech and text is important to the usefulness of the digital

assistant. For example, if the words in a speech input are incorrectly identified by a speech-to-text process, the digital assistant may not be able to properly infer the user's intent, or may provide incorrect or unhelpful responses. Similarly, if the words in a speech output are incorrectly pronounced by the digital assistant, the user may have difficulty understanding the digital assistant. Incorrect pronunciations by the digital assistant also make the assistant appear incompetent or unsophisticated, and may reduce users' interest and confidence in the digital assistant.

[0005] In order to improve the quality of digital assistants, it is helpful to identify particular instances where errors have occurred, so that the source of the errors can be identified and addressed. However, it is difficult to identify errors made by a digital assistant, because there is often limited or no feedback about whether an error has occurred. Moreover, even if the occurrence of errors can be detected, it can be difficult to determine exactly what the error was or what part of an interaction or task performed by the digital assistant was perceived by the user to be in error.

[0006] Accordingly, there is a need for systems and methods to determine when errors occur in interactions with a digital assistant.

SUMMARY

[0007] The implementations described herein relate to determining when a digital assistant makes a mistake. Digital assistants are capable of performing many different types of actions in order to satisfy a user's request. For example, a user can issue a speech command such as "Call Jim Carpenter," and the digital assistant should initiate or otherwise facilitate a telephone call to a contact named Jim Carpenter. As another example, a user can issue a speech command requesting a restaurant reservation for a specified time and party size, and the digital assistant should determine the user's intent (e.g., that the user wants to make restaurant reservations), and perform the actions necessary to reserve a table in accordance with the user's request.

[0008] Because of the complexity of the tasks that modern digital assistants are capable of performing, there are many problems or errors that can prevent the digital assistant from achieving satisfactory results. For example, errors in speech-to-text processing can cause the digital assistant to incorrectly infer a user's intent—if the speech command "Call Jim Carpenter" is incorrectly understood as "Carl Jim Carpenter," the digital assistant is unlikely to successfully infer that the user intended to place a telephone call. Errors in natural language processing can also lead to errors, such as if a user asks for "dinner tables from Ikea," expecting the digital

assistant to search for inexpensive Swedish furniture, and the digital assistant infers that the user is requesting dinner reservations at a restaurant called Ikea.

[0009] Except in cases where the digital assistant fails to infer the user's intent altogether, the digital assistant is often unable to determine whether the actions it took or the responses it provided in response to a user request were correct. In particular, digital assistants often receive no explicit feedback as to whether any particular task was correctly identified and/or executed. Moreover, even though it may be possible to identify successful interactions based on users' acceptance of the digital assistant's actions and/or suggestions, a user's failure to accept the digital assistant's actions and/or suggestions alone is not necessarily indicative of any particular error. For example, if a digital assistant responds to a request to "Call Jim Carpenter" by providing a prompt allowing the user to initiate a call to a contact named Jim Carpenter, and the user does initiate the call, the digital assistant can safely infer that it successfully inferred the user's intent. However, a cancellation of the call or a failure to initiate the call could be because the digital assistant incorrectly inferred the user's intent (e.g., it provided a prompt to call the incorrect person), but it could also be because the user received a text message from Jim Carpenter just prior to placing the call, and no longer wished to call him. Or, the user was simply testing out the capabilities of the digital assistant. In these cases, the digital assistant cannot accurately infer that it made a mistake based on the user ignoring the digital assistant's suggestion.

[0010] However, certain interactions with the digital assistant (and/or with the device through which the user is interacting with the digital assistant) are better indicators of a problem in the performing of an action by the digital assistant. For example, a user may provide an input (e.g., a speech input) to the digital assistant such as "you got that wrong," or "what are you talking about," which clearly indicate an error. Users may even provide angry or insulting inputs to express their frustration with an error by the digital assistant, such as "forget you!" or "that was way off!" or "what are you talking about!?" As another example, a user may select an affordance (e.g., a touchscreen button) indicating that there was a problem with a response provided or action taken by the digital assistant. As yet another example, a user may manifest their frustration physically, such as by shaking a device being used to interact with the digital assistant (e.g., a smart phone), or slamming their hands on a keyboard.

[0011] In order to improve the digital assistant in response to these interactions, though, the digital assistant must be modified, tuned, or otherwise adjusted so that similar problems can be avoided in the future. Accordingly, once the digital assistant determines or detects a user interaction that is indicative of a problem in the performing of an action by the digital assistant,

information relating to the potentially problematic interaction in a repository is stored for error analysis. Error analysis can be performed automatically (e.g., with computer-implemented machine learning techniques), manually (e.g., by a technician reviewing the information and making adjustments to the digital assistant), or with a combination of automatic and manual techniques.

[0012] The implementations disclosed herein provide methods, systems, computer readable storage medium and user interfaces for a digital assistant to determine when a problem relating to the performing of an action by the digital assistant occurs, and taking actions designed to avoid additional instances of that problem from occurring.

[0013] According to some implementations, a method is performed at an electronic device with one or more processors and memory storing one or more programs for execution by the one or more processors. A speech input containing a request is received from a first user. At least one action in furtherance of satisfying the request is performed. A user interaction is detected. It is determined whether the user interaction is indicative of a problem in the performing of the at least one action. Upon determining that the user interaction is indicative of a problem, information relating to the request is stored in a repository for error analysis.

[0014] In some implementations, detecting the user interaction comprises detecting a second speech input, and determining whether the user interaction is indicative of a problem comprises determining that the second speech input indicates dissatisfaction with the at least one action. In some implementations, determining whether the second speech input indicates dissatisfaction includes determining whether the second speech input includes at least one predefined word.

[0015] In some implementations, determining whether the second speech input indicates dissatisfaction includes determining a volume of the second speech input. In some implementations, the second speech input indicates dissatisfaction when the volume of the second speech input is above a predefined threshold.

[0016] In some implementations, determining whether the second speech input indicates dissatisfaction includes determining whether the second speech input contains prosodic indications of frustration.

[0017] In some implementations, determining whether the second speech input indicates dissatisfaction includes determining whether the second speech input includes substantially the same words as the first speech input.

[0018] In some implementations, detecting the user interaction comprises detecting a second speech input and a third speech input, and determining whether the user interaction is indicative of a problem comprises determining that the second speech input and the third speech input

indicate dissatisfaction with the at least one action, wherein determining whether the second speech input indicates dissatisfaction includes determining that the second speech input and the third speech input each include substantially the same words as the first speech input.

[0019] In some implementations, the user interaction is any one or more of: a predefined motion of the device; a selection of an affordance; a termination of a dialog session with the intelligent automated assistant; and a rejection of a proposed task.

[0020] In some implementations, Upon determining that the user interaction is indicative of a problem, a first prompt requesting the user to confirm whether there was a problem in the performing of the at least one action is provided to the user. A confirmation or a disconfirmation of whether there was a problem in the performing of the at least one action is received from the user.

[0021] In some implementations, upon detecting a confirmation that there was a problem in the performing of the at least one action, providing a second prompt acknowledging that the problem occurred. In some implementations, the second prompt includes a request for the user to restate the speech input containing the request.

[0022] In some implementations, where the repository includes a plurality of entries from a plurality of users, the repository is analyzed to identify a set of entries, each entry of the set of entries having one or more similar characteristics indicative of an error. One or more of a speech-to-text module and a natural language processing module is adjusted based on the set of entries so as to reduce reproduction of the error.

[0023] In accordance with some embodiments, an electronic device includes one or more processors, memory, and one or more programs; the one or more programs are stored in the memory and configured to be executed by the one or more processors and the one or more programs include instructions for performing the operations of any of the methods described above. In accordance with some embodiments, a computer readable storage medium has stored therein instructions which when executed by an electronic device, cause the device to perform the operations of any of the methods described above. In accordance with some embodiments, an electronic device includes: means for performing the operations of any of the methods described above. In accordance with some embodiments, an information processing apparatus, for use in an electronic device, includes means for performing the operations of any of the methods described above.

[0024] The details of one or more implementations of the subject matter described in this specification are set forth in the accompanying drawings and the description below. Other

features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] Figure 1 is a block diagram illustrating an environment in which a digital assistant operates in accordance with some implementations.

[0026] Figure 2 is a block diagram illustrating a digital assistant client system in accordance with some implementations.

[0027] Figure 3A is a block diagram illustrating a digital assistant system or a server portion thereof in accordance with some implementations.

[0028] Figure 3B is a block diagram illustrating functions of the digital assistant shown in Figure 3A in accordance with some implementations.

[0029] Figure 3C is a diagram of a portion of an ontology in accordance with some implementations.

[0030] Figures 4A-4C are flow diagrams of an exemplary method for operating an intelligent automated assistant, in accordance with some implementations

[0031] Like reference numerals refer to corresponding parts throughout the drawings.

DESCRIPTION OF IMPLEMENTATIONS

[0032] 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," or "automatic digital assistant," refer to any information processing system that interprets natural language input in spoken and/or textual form to infer user intent, and performs actions based on the inferred user intent. For example, to act on a inferred user intent, the system can perform one or more of the following: identifying a task flow with steps and parameters designed to accomplish the inferred user intent, inputting specific requirements from the inferred user intent into the task flow; executing the task flow by invoking programs, methods, services, APIs, or the like; and generating output responses to the user in an audible (e.g. speech) and/or visual form.

[0033] Specifically, a digital assistant is capable of accepting a user request at least partially in the form of a natural language command, request, statement, narrative, and/or inquiry. Typically, the user request seeks either an informational answer or performance of a task by the digital assistant. A satisfactory response to the user request is 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 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, "Please invite my friends to my girlfriend's birthday party next week." In response, the digital assistant may acknowledge the request by saying "Yes, right away," and then send a suitable calendar invite on behalf of the user to each of the user' friends listed in the user's electronic address book. During performance of a requested task, the digital assistant sometimes interacts with the user in a continuous dialogue involving multiple exchanges of information over an extended period of time. There are numerous other ways of interacting with a digital assistant to request information or performance of various tasks. In addition to providing verbal responses and taking programmed actions, the digital assistant also provides responses in other visual or audio forms, e.g., as text, alerts, music, videos, animations, etc.

[0034] An example of a digital assistant is described in Applicant's U.S. Utility Application Serial No. 12/987,982 for "Intelligent Automated Assistant," filed January 10, 2011, the entire disclosure of which is incorporated herein by reference.

[0035] As shown in Figure 1, in some implementations, a digital assistant is implemented according to a client-server model. The digital assistant includes a client-side portion 102a, 102b (hereafter "DA client 102") executed on a user device 104a, 104b, and a server-side portion 106 (hereafter "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.

[0036] 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, and an I/O interface to external services 118. 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 inferred user intent. In some implementations, the DA-server 106 communicates with external services 120 through the network(s) 110 for task completion or information acquisition. The I/O interface to external services 118 facilitates such communications.

[0037] 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 smart phone, 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 other data processing devices. More details on the user device 104 are provided in reference to an exemplary user device 104 shown in Figure 2.

[0038] 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 e.g., 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.

[0039] The server system 108 is implemented on one or more standalone data processing apparatus 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.

[0040] Although the digital assistant 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, the functions of a digital assistant is 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 is a thin-client that provides only user-facing input and output processing functions, and delegates all other functionalities of the digital assistant to a backend server.

[0041] 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.

[0042] For example, a motion sensor 210, a light sensor 212, and a proximity sensor 214 are coupled to the peripherals interface 206 to facilitate orientation, light, and proximity sensing functions. One or more other sensors 216, such as a positioning system (e.g., GPS receiver), a temperature sensor, a biometric sensor, a gyro, a compass, an accelerometer, and the like, are also connected to the peripherals interface 206, to facilitate related functionalities.

[0043] In some implementations, a camera subsystem 220 and an optical sensor 222 are utilized to facilitate camera functions, such as taking photographs and recording video clips. Communication functions are facilitated through one or more wired and/or wireless communication subsystems 224, which can include various communication ports, radio frequency receivers and transmitters, and/or optical (e.g., infrared) receivers and transmitters. An audio subsystem 226 is coupled to speakers 228 and a microphone 230 to facilitate voice- enabled functions, such as voice recognition, voice replication, digital recording, and telephony functions.

[0044] In some implementations, an I/O subsystem 240 is also coupled to the peripherals interface 206. The I/O subsystem 240 includes a touch screen controller 242 and/or other input controller(s) 244. The touch-screen controller 242 is coupled to a touch screen 246. The touch screen 246 and the touch screen controller 242 can, 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. The other input controller(s) 244 can be coupled to other input/control devices 248, such as one or more buttons, rocker switches, thumb-wheel, infrared port, USB port, and/or a pointer device such as a stylus.

[0045] In some implementations, the memory interface 202 is coupled to memory 250. The memory 250 can include high-speed random access memory and/or non-volatile memory, such as one or more magnetic disk storage devices, one or more optical storage devices, and/or flash memory (e.g., NAND, NOR).

[0046] In some implementations, the memory 250 stores an operating system 252, a communication module 254, a user interface module 256, a sensor processing module 258, a phone module 260, and applications 262. The operating system 252 includes instructions for handling basic system services and for performing hardware dependent tasks. The communication module 254 facilitates communicating with one or more additional devices, one or more computers and/or one or more servers. The user interface module 256 facilitates graphic user interface processing and output processing using other output channels (e.g., speakers). The sensor processing module 258 facilitates sensor-related processing and functions. The phone module 260 facilitates phone-related processes and functions. The application module 262

facilitates various functionalities of user applications, such as electronic-messaging, web browsing, media processing, Navigation, imaging and/or other processes and functions.

[0047] As described in this specification, the 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, to-do lists, shopping lists, user-specified name pronunciations, etc.) to provide the client-side functionalities of the digital assistant.

[0048] In various implementations, the digital assistant client module 264 is capable of accepting voice input (e.g., speech input), text input, touch input, and/or gestural input through various user interfaces (e.g., the I/O subsystem 244) of the user device 104. The digital assistant client module 264 is also capable of providing output in audio (e.g., speech output), 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 using the communication subsystems 224.

[0049] In some implementations, the digital assistant client module 264 includes a speech synthesis module 265. The speech synthesis module 265 synthesizes speech outputs for presentation to the user. The speech synthesis module 265 synthesizes speech outputs based on text provided by the digital assistant. For example, the digital assistant generates text to provide as an output to a user, and the speech synthesis module 265 converts the text to an audible speech output. The speech synthesis module 265 uses any appropriate speech synthesis technique in order to generate speech outputs from text, including but not limited to concatenative synthesis, unit selection synthesis, diphone synthesis, domain-specific synthesis, formant synthesis, articulatory synthesis, hidden Markov model (HMM) based synthesis, and sinewave synthesis.

[0050] In some implementations, instead of (or in addition to) using the local speech synthesis module 265, speech synthesis is performed on a remote device (e.g., the server system 108), and the synthesized speech is sent to the user device 104 for output to the user. For example, this occurs in some implementations where outputs for a digital assistant are generated at a server system. And because server systems generally have more processing power or resources than a user device, it may be possible to obtain higher quality speech outputs than would be practical with client-side synthesis.

[0051] In some implementations, the digital assistant client module 264 utilizes the 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, the current user

interaction, and/or the current 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 to help infer the user's intent. In some implementations, the digital assistant also uses the context information to determine how to prepare and delivery outputs to the user.

[0052] In some implementations, the context information that accompanies the user input includes sensor information, e.g., lighting, ambient noise, ambient temperature, images or videos of the surrounding environment, etc. In some 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 are provided to the digital assistant server as context information associated with a user input.

[0053] In some implementations, the DA client module 264 selectively provides information (e.g., 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. 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.

[0054] In various implementations, the memory 250 includes 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.

[0055] Figure 3A is a block diagram of an example digital assistant system 300 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. 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 instructions for execution by one or more processors, firmware, including one or more signal processing and/or application specific integrated circuits, or a combination of thereof.

[0056] 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.

[0057] In some implementations, the 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, flash memory devices, or other non-volatile solid-state memory devices).

[0058] In some implementations, the I/O interface 306 couples input/output devices 316 of the digital assistant system 300, such as displays, 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, e.g., 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. 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).

[0059] In some implementations, the network communications interface 308 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 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 308 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.

[0060] In some implementations, memory 302, or the computer readable storage media 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, and a digital assistant module 326. The one or more processors 304 execute these programs, modules, and instructions, and reads/writes from/to the data structures.

[0061] The operating system 318 (e.g., Darwin, RTXC, LINUX, UNIX, OS X, WINDOWS, or an embedded operating system such as VxWorks) includes various software components and/or drivers for controlling and managing general system tasks (e.g., memory management, storage device control, power management, etc.) and facilitates communications between various hardware, firmware, and software components.

[0062] The communications module 320 facilitates communications between the digital assistant system 300 with other devices over the network communications interface 308. For example, the communications module 320 may communicate with the communication module 254 of the device 104 shown in Figure 2. The communications module 320 also includes various components for handling data received by the wireless circuitry 314 and/or wired communications port 312.

[0063] 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, pointing device, controller, and/or microphone), and generates user interface objects on a display. The user interface module 322 also prepares and delivers outputs (e.g., speech, sound, animation, text, icons, vibrations, haptic feedback, and light, etc.) to the user via the I/O interface 306 (e.g., through displays, audio channels, speakers, and touch-pads, etc.).

[0064] 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.

[0065] The 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, a speech interaction error detection module 339, an error analysis repository 340, and an error analysis module 342. Each of these 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, task flow models 354, and service models 356.

[0066] In some implementations, using the processing modules, data, and models implemented in the digital assistant module 326, the digital assistant 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, games, intentions, etc.); determining the task flow for fulfilling the inferred intent; and executing the task flow to fulfill the inferred intent.

[0067] In some implementations, as shown in Figure 3B, 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 (e.g., as speech outputs) 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 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. 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.

[0068] The speech-to-text processing module 330 (or speech recognizer) receives speech input (e.g., a user utterance captured in a voice recording) through the I/O processing module 328. In some implementations, the STT 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 STT processing module 330 can be 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 STT 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.

[0069] In some implementations, the STT processing module 330 includes and/or accesses a vocabulary of recognizable words, each associated with one or more candidate pronunciations of the word represented in a speech recognition phonetic alphabet. For example, the vocabulary may include the word "tomato" in association with the candidate pronunciations of "tuh-may-doe" and "tuh-mah-doe." In some implementations, the candidate pronunciations for words are determined based on the spelling of the word and one or more linguistic and/or phonetic rules. In some implementations, the candidate pronunciations are manually generated, e.g., based on known canonical pronunciations.

[0070] In some implementations, the candidate pronunciations are ranked based on the commonness of the candidate pronunciation. For example, the candidate pronunciation "tuh-may-doe" may be ranked higher than "tuh-mah-doe," because the former is a more commonly used pronunciation (e.g., among all users, for users in a particular geographical region, or for any other appropriate subset of users). In some implementations, one of the candidate pronunciations is selected as a predicted pronunciation (e.g., the most likely pronunciation).

[0071] When an utterance is received, the STT processing module 330 attempts to identify phonemes in the utterance (e.g., using an acoustic model), and then attempts to identify words that match the phonemes (e.g., using a language model). For example, if the STT processing module 330 first identifies the sequence of phonemes "tuh-may-doe" in an utterance, it then determines, based on the vocabulary index 344, that this sequence corresponds to the word "tomato."

[0072] In some implementations, the STT processing module 330 uses approximate matching techniques to determine words in an utterance. Thus, for example, the STT processing module 330 can determine that the sequence of phonemes "duh-may-doe" corresponds to the word "tomato," even if that particular sequence of phonemes is not one of the candidate phonemes for that word.

[0073] The natural language processing module 332 ("natural language processor") of the digital assistant 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. An "actionable intent" represents a

task that can be performed by the digital assistant, 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 takes in order to perform the task. The scope of a digital assistant's capabilities is dependent on the number and variety of task flows that have been implemented and stored in the task flow models 354, or in other words, on the number and variety of "actionable intents" that the digital assistant recognizes. The effectiveness of the digital assistant, however, is also dependent on the assistant's ability to infer the correct "actionable intent(s)" from the user request expressed in natural language.

[0074] In some implementations, in addition to the sequence of words or tokens obtained from the speech-to-text processing module 330, the natural language processing module 332 also receives context information associated with the user request, e.g., from the I/O processing module 328. The natural language processing module 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, sensor information collected before, during, or shortly after the user request, prior interactions (e.g., dialogue) between the digital assistant and the user, and the like. As described in this specification, context information is dynamic, and can change with time, location, content of the dialogue, and other factors.

[0075] In some implementations, the natural language processing is based on e.g., ontology 360. The ontology 360 is a hierarchical structure containing many nodes, each node representing either an "actionable intent" or a "property" relevant to one or more of the "actionable intents" or other "properties". As noted above, an "actionable intent" represents a task that the digital assistant is capable of performing, i.e., it is "actionable" or can be acted on. A "property" represents a parameter associated with an actionable intent or a sub-aspect of another property. A linkage between an actionable intent node and a property node in the ontology 360 defines how a parameter represented by the property node pertains to the task represented by the actionable intent node.

[0076] 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, as shown in Figure 3C, the ontology 360 may include a "restaurant reservation" node (i.e., an actionable intent node). Property nodes

"restaurant," "date/time" (for the reservation), and "party size" are each directly linked to the actionable intent node (i.e., the "restaurant reservation" node).

[0077] In addition, property nodes "cuisine," "price range," "phone number," and "location" are sub-nodes of the property node "restaurant," and are each linked to the "restaurant reservation" node (i.e., the actionable intent node) through the intermediate property node "restaurant." For another example, as shown in Figure 3C, the ontology 360 may also include a "set reminder" node (i.e., 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.

[0078] 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 different 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.

[0079] While Figure 3C illustrates two example domains within the ontology 360, other domains (or actionable intents) include, for example, "initiate a phone call," "find directions," "schedule a meeting," "send a message," and "provide an answer to a question," "read a list", "providing navigation instructions," "provide instructions for a task" and so on. A "send a message" domain is associated with a "send a message" actionable intent node, and 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."

[0080] 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 entire domains or nodes, or by modifying relationships between the nodes within the ontology 360.

[0081] 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."

[0082] 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 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," "eat," "pizza," "fast food," "meal," and so on. For another example, the vocabulary associated with the node for the actionable intent of "initiate a phone call" 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.

[0083] The natural language processing module 332 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. Based on the quantity and/or relative importance of the activated nodes, the natural language processing module 332 will select one of the actionable intents as the task 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 has previously correctly interpreted a similar request from a user.

[0084] In some implementations, the digital assistant 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 processing module 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 searches and identifies specific entity names from other data sources, such as the user's address book, a movies database, a musicians database, and/or a restaurant database. In some implementations, when the natural language processing module 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), that word is given additional significance in selecting the actionable intent within the ontology for the user request.

[0085] 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.

[0086] 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. In some implementations, the natural language processing module 332 uses the user-specific information to supplement the information contained in the user input to further define the user intent. For example, for a user request "invite my friends to my birthday party," the natural language processing module 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.

[0087] Other details of searching an ontology based on a token string is described in U.S. Utility Application Serial No. 12/341,743 for "Method and Apparatus for Searching Using An

Active Ontology," filed December 22, 2008, the entire disclosure of which is incorporated herein by reference.

[0088] In some implementations, once the natural language processing module 332 identifies an actionable intent (or domain) based on the user request, the natural language processing module 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 processing module 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. In some implementations, based on the information contained in the user's utterance, the natural language processing module 332 generates a partial structured query for the restaurant reservation domain, where the partial structured query includes the parameters {Cuisine= "Sushi"} and {Time = "7pm"}. However, in this example, the user's utterance contains insufficient information to complete the structured query associated with the domain. Therefore, other necessary parameters such as {Party Size} and {Date} are not specified in the structured query based on the information currently available. In some implementations, the natural language processing module 332 populates some parameters of the structured query with received context information. For example, in some implementations, if the user requested a sushi restaurant "near me," the natural language processing module 332 populates a {location} parameter in the structured query with GPS coordinates from the user device 104.

[0089] In some implementations, the natural language processing module 332 passes the structured query (including any completed parameters) to the task flow processing module 336 ("task flow processor"). The task flow processing module 336 is configured to receive the structured query from the natural language processing module 332, complete the structured query, if necessary, and perform 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 include procedures for obtaining additional information from the user, and task flows for performing actions associated with the actionable intent.

[0090] As described above, in order to complete a structured query, the task flow processing module 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 processing module 336 invokes the dialogue flow processing module 334 to engage in a dialogue with the user. In some implementations, the dialogue flow processing module 334 determines how (and/or when) to ask the user for the additional information, and receives and processes the user responses. The questions are provided to and answers are received from the users through the I/O processing module 328. In some implementations, the dialogue flow 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., clicking) responses. Continuing with the example above, when the task flow processing module 336 invokes the dialogue flow processing module 334 to determine the "party size" and "date" information for the structured query associated with the domain "restaurant reservation," the dialogue flow processing module 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 flow processing module 334 can then populate the structured query with the missing information, or pass the information to the task flow processing module 336 to complete the missing information from the structured query.

[0091] In some cases, the task flow processing module 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 processing module 336 will request that the dialogue flow processing module 334 disambiguate this property of the structured query. In turn, the dialogue flow processing module 334 may ask the user "Which Bob?", and display (or read) a list of contacts named "Bob" from which the user may choose.

[0092] Once the task flow processing module 336 has completed the structured query for an actionable intent, the task flow processing module 336 proceeds to perform the ultimate task associated with the actionable intent. Accordingly, the task flow processing module 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 processing module 336 may perform the steps of: (1) logging onto a server of the ABC Café or a restaurant reservation system such as OPENTABLE®, (2) entering the date, time, and party size information in a form on the website, (3) submitting the form, and (4) making a calendar entry for the reservation in the user's calendar.

[0093] In some implementations, the task flow processing module 336 employs the assistance of a service processing module 338 ("service processing module") to complete a task requested in the user input or to provide an informational answer requested in the user input. For example, the service processing module 338 can act on behalf of the task flow processing module 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, and invoke or interact with third party services (e.g. a restaurant reservation portal, a social networking website, a banking portal, etc.). 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 processing module 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.

[0094] For example, if a restaurant has enabled an online reservation service, the restaurant can submit a service model specifying the necessary parameters for making a reservation and the APIs for communicating the values of the necessary parameter to the online reservation service. When requested by the task flow processing module 336, the service processing module 338 can establish a network connection with the online reservation service using the web address stored in the service model, and send the necessary parameters of the reservation (e.g., time, date, party size) to the online reservation interface in a format according to the API of the online reservation service.

[0095] In some implementations, the natural language processing module 332, dialogue flow processing module 334, and task flow processing module 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 (i.e., an output to the user, or the completion of a task) to fulfill the user's intent.

[0096] 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.

[0097] The error detection module 339 detects errors in interactions between a user and the digital assistant. In some implementations, to detect errors, the error detection module 339 monitors interactions between a user and the digital assistant, and/or between a user and a user

device. For example, the error detection module 339 monitors any of the following types of interactions, or a subset thereof: the content of a user 's speech inputs to the digital assistant (e.g., if a user says "you got that wrong," or "you are pronouncing that wrong," or if a user provides the same input multiple times within a short time), the prosody and/or mood of a user's spoken inputs (e.g., the volume, pace, rhythm, stress, and/or intonation of an input), affordance selections (e.g., if a user selects a lock-screen button or other affordance to cancel an action), movements of the device (e.g., shaking the device, setting the device down in a certain orientation, such as screen-down), termination of actions or suggested actions on the user device (e.g., cancelling a telephone call, email, text message, etc. after the digital assistant initiates or suggests it), initiation of an action shortly after a digital assistant fails to successfully infer an intent or adequately respond to a user, etc. In some implementations, the error detection module 339 monitors other types of interactions to detect errors as well. Additional details of how errors are detected by the error detection module 339 are discussed herein.

[0098] In order to detect such errors, in some implementations, the error detection module 339 communicates with or otherwise receives information from various modules and components of the digital assistant system 300 and/or the user device 104, such as the I/O processing module 328 (and/or the I/O devices 316), 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, the phone module 260, the sensor processing module 258, the I/O subsystem 240, and/or any of the sensors or I/O devices associated therewith. After detection of an error, information from these or other modules and components is stored in the error analysis repository 340 for further analysis. In some implementations, when an error is detected, the error detection module 339 prompts the user to confirm whether an error actually occurred, as discussed herein. Thus, the accuracy of the error detection module 339 is increased, as the user can quickly and easily provide definitive feedback to confirm or deny whether an error has actually occurred.

[0099] In some implementations, the error detection module 339 monitors actions taken by the user (e.g., on the user device 104) after the user cancels an action and/or dialogue with the digital assistant before successful completion of the action or dialogue. In particular, actions taken by the user after such a cancellation often indicates that the digital assistant did not accurately infer the user's intent (and/or did not correctly understand the user's speech input), as well as what the digital assistant should have done based on the user's input. As a specific example, a user may ask the digital assistant to "Call Philippe," and the digital assistant may respond by saying "Calling Phil" (e.g., because the digital assistant did not properly understand the speech input).

The user may quickly cancel the telephone call to Phil, and then proceed to manually initiate a telephone call with a contact named Philippe. Accordingly, the error detection module 339 detects that, because the telephone call to Phil was canceled, an error was made.

[0100] In some implementations, the error detection module 339 also learns correct actions to take in response to the post-cancellation interactions. For example, the digital assistant can learn that because the user ultimately initiated a telephone call with a contact named Philippe, the user's input "Call Philippe" referred to that contact, rather than the "Phil" that was initially identified by the digital assistant.

[0101] The error analysis repository 340 stores information to be analyzed to help improve the digital assistant system. In some implementations, information is stored in the error analysis repository 340 in response to a determination that an error or problem in the performing of an action has occurred. As discussed herein, there are various techniques for determining that an error or problem has occurred, any of which may be used (alone or in combination) to cause information to be stored in the error analysis repository 340.

[0102] The repository 340 stores information about user interactions with the digital assistant, such as a transcript of a user's inputs and the digital assistant's outputs, a record of actions taken by the digital assistant (e.g., a record of a call that the digital assistant initiated in response to a speech input, or any other action), a record of user interactions with a user device (e.g., button/touchscreen selections, accelerometer data, etc.), etc. In some implementations, such information is stored in the error analysis repository 340 only after it is determined that an error has likely occurred (e.g., because the user shouted at the device or selected an affordance indicating an error).

[0103] In some implementations, a record of each user interaction with the digital assistant (e.g., including speech input/output transcripts, records of button selections, accelerometer data, etc.) is automatically stored in the error analysis repository 340. Then, if no interaction indicative of an error is detected during or within a predetermined duration after the interaction, the record of the interaction is removed from the repository. If an interaction indicative of an error is detected, however, at least a part of the record of the interaction is stored for later analysis.

[0104] In some implementations, the error analysis repository 340 includes user interaction information from a plurality of users of digital assistants. Accordingly, the error analysis repository 340 can be used to identify systemic errors and/or problems, as well as or in addition to errors and/or problems that are specific to individual users (e.g., because of accents or grammatical idiosyncrasies of a particular user).

[0105] The error analysis module 342 analyzes the information in the error analysis repository 340 to identify individual errors and/or patterns of errors by the digital assistant. For example, the error analysis module 342 may use one or more machine learning techniques to process the data in the error analysis repository 340 to identify patterns of interactions that are indicative of a problem. As one specific example, the error analysis module 342 identifies instances where a particular speech input was detected from multiple individual users (e.g., "Tickets to are go," resulting from a misunderstanding of the movie name "Argo"), and where the digital assistant is unable to infer the user's intent (e.g., because the request, as transcribed, is nonsensical). If this pattern of interactions occurs frequently enough in the error analysis repository 340, the error analysis module 342 can determine that an adjustment to one or more components of the digital assistant should be made in order to avoid similar errors in the future. In some implementations, the error analysis module 342 also automatically (e.g., without human intervention) adjusts one or more attributes or processes of the digital assistant (e.g., an acoustic or language model of the speech-to-text processing module 330, etc.) in response to detecting the pattern in the error analysis repository 340. In some implementations, the error analysis module 342 facilitates manual adjustment or tuning of one or more attributes or processes of the digital assistant, for example, by providing a report of the error and/or pattern of interactions that appear to indicate a systemic error to a human operator. The human operator may then review the information and take appropriate actions to rectify the error.

[0106] More details on the digital assistant can be found in the U.S. Utility Application No. 12/987,982, entitled "Intelligent Automated Assistant", filed January 18, 2010, U.S. Utility Application No. 61/493,201, entitled "Generating and Processing Data Items That Represent Tasks to Perform", filed June 3, 2011, the entire disclosures of which are incorporated herein by reference.

[0107] In most scenarios, when the digital assistant receives a user input from a user, the digital assistant attempts to provide an appropriate response to the user input with as little delay as possible. For example, suppose the user requests certain information (e.g., current traffic information) by providing a speech input (e.g., "How does the traffic look right now?"). Right after the digital assistant receives and processes the speech input, the digital assistant optionally provides a speech output (e.g., "Looking up traffic information ...") acknowledging receipt of the user request. After the digital assistant obtains the requested information in response to the user request, the digital assistant proceeds to provide the requested information to the user without further delay. For example, in response to the user's traffic information request, the digital assistant may provide a series of one or more discrete speech outputs separated by brief pauses

(e.g., "There are 2 accidents on the road. <Pause> One accident is on 101 north bound near Whipple Avenue. <Pause> And a second accident is on 85 north near 280."), immediately after the speech outputs are generated.

[0108] For the purpose of this specification, the initial acknowledgement of the user request and the series of one or more discrete speech outputs provided in response to the user request are all considered sub-responses of a complete response to the user request. In other words, the digital assistant initiates an information provision process for the user request upon receipt of the user request, and during the information provision process, the digital assistant prepares and provides each sub-response of the complete response to the user request without requiring further prompts from the user.

[0109] Sometimes, additional information or clarification (e.g., route information) is required before the requested information can be obtained. In such scenarios, the digital assistant outputs a question (e.g., "Where are you going?") to the user asking for the additional information or clarification. In some implementations, the question provided by the digital assistant is considered a complete response to the user request because the digital assistant will not take further actions or provide any additional response to the user request until a new input is received from the user. In some implementations, once the user provides the additional information or clarification, the digital assistant initiates a new information provision process for a "new" user request established based on the original user request and the additional user input.

[0110] In some implementations, the digital assistant initiates a new information provision process upon receipt of each new user input, and each existing information provision process terminates either (1) when all of the sub-responses of a complete response to the user request have been provided to the user or (2) when the digital assistant provides a request for additional information or clarification to the user regarding a previous user request that started the existing information provision process.

[0111] In general, after a user request for information or performance of a task is received by the digital assistant, it is desirable that the digital assistant provides a response (e.g., either an output containing the requested information, an acknowledgement of a requested task, or an output to request a clarification) as promptly as possible. Real-time responsiveness of the digital assistant is one of the key factors in evaluating performance of the digital assistant. In such cases, a response is prepared as quickly as possible, and a default delivery time for the response is a time immediately after the response is prepared.

[0112] Sometimes, however, after an initial sub-response provided immediately after receipt of the user input, the digital assistant provides the remaining one or more sub- responses one at a

time over an extended period of time. In some implementations, the information provision process for a user request is stretched out over an extended period of time that is longer than the sum of the time required to provide each sub-response individually. For example, in some implementations, short pauses (i.e., brief periods of silence) are inserted between an adjacent pair of sub-responses (e.g., a pair of consecutive speech outputs) when they are delivered to the user through an audio-output channel.

[0113] In some implementations, a sub-response is held in abeyance after it is prepared and is delivered only when a predetermined condition has been met. In some implementations, the predetermined condition is met when a predetermined trigger time has been reached according to a system clock and/or when a predetermined trigger event has occurred. For example, if the user says to the digital assistant "set me a timer for 5 minutes," the digital assistant initiates an information provision process upon receipt of the user request. During the information provision process, the digital assistant provides a first sub-response (e.g., "OK, timer started.") right away, and does not provide a second and final sub-response (e.g., "OK, five minutes are up") until 5 minutes later. In such cases, the default delivery time for the first sub-response is a time immediately after the first sub-response is prepared, and the default delivery time for the second, final sub-response is a time immediately after the occurrence of the trigger event (e.g., the elapse of 5 minutes from the start of the timer). The information provision process is terminated when the digital assistant finishes providing the final sub-response to the user. In various implementations, the second sub-response is prepared any time (e.g., right after the first sub-response is prepared, or until shortly before the default delivery time for the second sub-response) before the default delivery time for the second sub-response.

[0114] Figures 4A-4C are flow diagrams of an exemplary method 400 implemented by a digital assistant. In some implementations, the method 400 is performed at an electronic device with one or more processors and memory storing one or more programs for execution by the one or more processors. For example, in some implementations, the method 400 is performed at the user device 104 or the server system 108. In some implementations, the method 400 is performed by the digital assistant system 300 (Figure 3A), which, as noted above, may be implemented on a standalone computer system (e.g., either the user device 104 or the server system 108) or distributed across multiple computers (e.g., the user device 104, the server system 108, and/or additional or alternative devices or systems). While the following discussion describes the method 400 as being performed by a digital assistant (e.g., the digital assistant system 300), the method is not limited to performance by any particular device or combination of devices. Moreover, the

individual steps of the method may be distributed among the one or more computers, systems, or devices in any appropriate manner.

[0115] The digital assistant receives, from a user, a speech input containing a request (402). In some implementations, the speech input corresponds to a user utterance recorded and/or received by the user device 104. In some implementations, the speech input is received in the course of, or as part of, an interaction with the digital assistant. The request may be any request, including a request that indicates a task that the digital assistant can perform (e.g., making and/or facilitating restaurant reservations, initiating telephone calls and text messages, etc.), a request for a response (e.g., an answer to a question, such as "how far is Earth from the sun?"), and the like.

[0116] The digital assistant performs at least one action in furtherance of satisfying the request (404). In some implementations, the at least one action includes displaying a transcription of the speech input (e.g., as generated by the speech-to-text processing module 330, Figure 3A) on a display of a user device.

[0117] In some implementations, the at least one action includes generating a speech output that summarizes or describes the intent inferred by the digital assistant from the speech input (e.g., using the natural language processing module 332). For example, the digital assistant may output the phrase "Searching the web for information about pine trees" in response to the speech input "web search for pine trees."

[0118] In some implementations, the at least one action includes performing one or more tasks in a task flow that is intended to fulfill the user's intent (e.g., providing a prompt to initiate a telephone call, pre-populating a text message to a specified recipient, making restaurant reservations through an online portal, etc.).

[0119] The digital assistant detects a user interaction (406). The digital assistant determines whether the user interaction is indicative of a problem in the performing of the at least one action (407). Examples of user interactions that can be indicative of a problem in the performing of the at least one action, and how they are used by the digital assistant, are described in greater detail herein.

[0120] In some implementations, detecting the user interaction comprises detecting a second speech input, and determining whether the user interaction is indicative of a problem comprises determining that the second speech input indicates dissatisfaction with the at least one action (408).

[0121] A user's dissatisfaction can be determined in several possible ways. For example, users who are dissatisfied with the digital assistant may be likely to voice their displeasure to the digital assistant using some common words or phrases. Thus, in some implementations, determining

whether the second speech input indicates dissatisfaction includes determining whether the second speech input includes at least one predefined word (e.g., of a plurality of possible predefined words) (410). In some implementations, predefined words are selected based on a prediction (e.g., human or machine generated) as to what words or phrases users may say to the digital assistant when something goes wrong. In some implementations, predefined words are selected based on actual words and/or phrases that have been included in speech inputs to a digital assistant after an error has been detected (e.g., the predefined words are crowd-sourced). Specific examples of predefined words include the following words: error, wrong, incorrect, misunderstand, bad, etc. The predefined words may also include multi-word phrases, such as: "what was that?", "that was way off", "what are you talking about?", or even playful insults or rebukes to the digital assistant, such as "forget you!" or "you stink." In some implementations, the predefined words include spoken sounds that indicate frustration, such as an exasperated "ugh!"

[0122] Another way to determine a user's dissatisfaction is by evaluating the volume of the second speech input, because users may raise their voices in the second speech input, either out of frustration or in an attempt to provide a clearer or louder speech input for the digital assistant to process. Thus, in some implementations, determining whether the second speech input indicates dissatisfaction includes determining a volume of the second speech input (412). In some implementations, the second speech input indicates dissatisfaction when the volume of the second speech input is above a predefined threshold (414).

[0123] In some implementations, the threshold is an upper level of a normal speaking volume (e.g., 65 decibels, 70 decibels, 75 decibels, or any other appropriate level).

[0124] In some implementations, the predetermined threshold is based on the first speech input, so that if the second speech input is louder than the first speech input (e.g., by a predetermined amount), the digital assistant determines that the second speech input indicates dissatisfaction. This allows for a flexible approach to determining whether a second speech input was louder because of user frustration, or simply because the user is in a noisy environment or has a louder speaking voice than other users. In some implementations, the predetermined threshold is based on the average volume of the first speech input, the maximum volume of the first speech input, or the average or maximum volume of the first speech input plus an additional volume margin (e.g., 2, 3, or 5 decibels, or any other appropriate margin).

[0125] Yet another way to determine a user's dissatisfaction is by detecting a mood or emotion of the second speech input apart from the meanings of the words in the input. For example, prosodic features of the second speech input (e.g., the rhythm, pace, stress, intonation, volume, etc., of the speech input) can be used to determine whether the user is frustrated or annoyed.

Accordingly, in some implementations, determining whether the second speech input indicates dissatisfaction includes determining whether the second speech input contains prosodic indications of frustration (416). Examples of characteristic speaking styles that can be detected in order to indicate frustration include hyperarticulation (e.g., exaggerated pronunciation of words and/or syllables), exaggerated pauses, increased volume or pitch, and the like. Detecting annoyance and frustration based on prosody is described in greater detail, for example, in U.S. Pat. No. 7,912,720, "System and Method for Building Emotional Machines," filed on July 20, 2005, which is hereby incorporated by reference in its entirety.

[0126] Users may also indicate dissatisfaction by repeating the same speech input multiple times in an effort to make the digital assistant understand his or her words or intent. Accordingly, detecting the same input from a user multiple times within a short period of time and/or within the same dialog with the digital assistant can indicate that the user is not being properly understood, or that the digital assistant is not properly identifying the user's intent from the speech input. Thus, in some implementations, determining whether the second speech input indicates dissatisfaction includes determining whether the second speech input includes substantially the same words as the first speech input (418). In some implementations, the words in the first and second speech input must be identical in order for the digital assistant to detect dissatisfaction based on the speech inputs. In some implementations, the words in the first and second speech input may be somewhat different from one another.

[0127] In some implementations, detecting the user interaction comprises detecting a second speech input and a third speech input, and determining whether the user interaction is indicative of a problem comprises determining that the second speech input and the third speech input indicate dissatisfaction with the at least one action, wherein determining whether the second speech input indicates dissatisfaction includes determining that the second speech input and the third speech input each include substantially the same words as the first speech input (420). Thus, the digital assistant may detect dissatisfaction after the user repeats substantially the same input (e.g., using the same or substantially the same words) three times. In some implementations, the three inputs must be received within a predetermined time period in order for the digital assistant to infer that they indicate dissatisfaction. In some implementations, the predetermined time period is 30 seconds, 1 minute, 1.5 minutes, 2 minutes, or any other appropriate time period. In some implementations, the three inputs must be received within the same dialog session with the digital assistant (e.g., without the user leaving a user interface environment of the digital assistant on a user device).

[0128] In some implementations, more than three inputs that include substantially the same words as the first speech input must be received within the predetermined time period (e.g., 4, 5, 6, or more inputs).

[0129] Other user interactions with a user device can also indicate a problem in the performing of at least one action. In some implementations, the user interaction (detected at step (406)) is a predefined motion of the device (422). For example, in some implementations, the predefined motion is a shaking motion (e.g., as detected by the motion sensor 210, Figure 2). In some implementations, the predefined motion is a shaking motion having a certain motion profile (e.g., a certain speed, frequency, and/or magnitude of movement).

[0130] In some implementations, the user interaction (detected at step (406)) is a selection of an affordance (424). For example, when a user is interacting with a digital assistant on a user device (e.g., a computer, smart phone, etc.), an affordance is provided that, when selected, indicates a problem in the performing of an action by the digital assistant. In some implementations, the affordance is a physical button. In some implementations, the affordance is a touchscreen element. In some implementations, the touchscreen element is associated with text, such as "report a mistake," thus informing the user that selection of the affordance will indicate to the digital assistant that the user is dissatisfied with one of the assistant's actions and/or responses, or that the digital assistant has made a mistake or otherwise malfunctioned.

[0131] In some implementations, after selecting the affordance, the digital assistant receives a speech input providing additional details about the problem. For example, after detecting a selection of the affordance, the digital assistant prompts the user to provide the speech input, such as by saying "Sorry about that – can you please describe what went' wrong?", and indicates when the user should speak the explanation (e.g., with an audible tone and/or a visual indication that the digital assistant is listening).

[0132] The speech input is then stored (e.g., in the error analysis repository 340, Figure 3A) in addition to other information relating to the interaction between the user and the digital assistant, as described above. In some implementations, the error analysis module 342 performs speech processing on the speech input in order to identify additional details about the problem, which are then used by the error analysis module 342 (and/or a human operator) to adjust one or more aspects of the digital assistant system to help prevent future similar errors. As a specific example, the user may record a speech input such as "you didn't understand what I was saying" or "I was trying to get directions to McDonalds." This information can be very useful in determining what part of the interaction the user perceived as a problem or was otherwise dissatisfied with.

[0133] Sometimes, if a user becomes aware that the digital assistant is not going to properly satisfy the user's intent, the user will simply terminate the dialog with the assistant and perform the intended action manually (or simply forgo the action altogether). Thus, in some implementations, the user interaction (detected at step (406)) is a termination of a dialog session with the intelligent automated assistant (426). In some implementations, the termination of the dialog session occurs prior to satisfying the user's intent. For example, if a user issues a speech input to "Call Jim Carpenter," and the user cancels or terminates the dialog session with the intelligent automated assistant after receiving a prompt to initiate a call to "Tim Carpenter," the digital assistant will determine that there was a problem with the interaction.

[0134] In some implementations, the user interaction (detected at step (406)) corresponds to a rejection of a proposed task (428). For example, if the digital assistant prompts the user to accept or cancel a proposed task (e.g., to initiate a telephone call, send a text or email message, confirm restaurant reservations, etc.), rejection of the task may indicate that the user was dissatisfied with the proposed task, and that the digital assistant may have made an error. In some implementations, the proposed task is any task that the digital assistant does not execute without prior approval by a user. For example, in some implementations, the digital assistant does not initiate communications until the user has confirmed that the communication should be initiated (e.g., the request to "Call Jim Carpenter," will be followed with a prompt such as "I found Jim Carpenter in your contacts. Shall I call him?"). Accordingly, rejection of this task (e.g., a response of "No!" or a selection of a "cancel" button) corresponds to a rejection of the proposed task to initiate a call to a contact named Jim Carpenter.

[0135] In some implementations, upon determining that the user interaction is indicative of a problem (in step (407)), the digital assistant provides a first prompt requesting the user to confirm whether there was a problem in the performing of the at least one action (430). In some implementations, the first prompt is displayed on a touchscreen display of a user device. In some implementations, the digital assistant receives, from the user, a confirmation or a disconfirmation of whether there was a problem in the performing of the at least one action (432).

[0136] In some implementations, if the user confirms that there as a problem in the performing of the at least one action, the digital assistant provides a second prompt acknowledging that the problem occurred (434). By acknowledging the problem to the user, the digital assistant appears more responsive to the user's frustrations and difficulties, thus improving the user experience.

[0137] In some implementations, the second prompt includes a request for the user to restate the speech input containing the request (436). The restated speech input may be recorded and stored in a repository in association with other information relating to the request. This recording

can then be used by the digital assistant and/or a human operator to further identify what may have caused the problem and/or error, or otherwise led to the dissatisfaction of the user. This may also be beneficially employed where the speech input containing the request (received at step (402)) is not recorded or stored when it is received.

[0138] Upon determining that the user interaction is indicative of a problem, the digital assistant stores information relating to the request in a repository (e.g., the error analysis repository 340, Figure 3A) for error analysis (438). In some implementations, as described above, the digital assistant performs error analysis automatically, for example, by applying machine learning (or supervised machine learning) techniques to identify what caused the error and change or tune one or more aspects of the digital assistant to prevent the error from recurring.

[0139] Also noted above, the repository can include entries from many users, so that errors that appear for multiple users can be more easily detected, and global changes and/or adjustments to the digital assistant can be made for all users (or a particular subset of users), instead of just for one user at a time. Accordingly, in some implementations, the repository includes a plurality of entries from a plurality of users, and the digital assistant analyzes the repository to identify a set of entries, each entry of the set of entries having one or more similar characteristics indicative of an error (440). The digital assistant then adjusts one or more of a speech-to-text module and a natural language processing module based on the set of entries so as to reduce reproduction of the error (442). In some implementations, adjusting the speech-to-text module includes adjusting an acoustic model, a language model, or both. In some implementations, adjusting the natural language processing module includes adjusting an ontology.

[0140] In some implementations, adjusting one or more of a speech-to-text module and a natural language processing module includes adding words to a vocabulary (e.g., the vocabulary index 344). For example, the digital assistant may identify a set of entries where the movie title "Argo" was incorrectly detected as the phrase "are go" by a speech-to-text processor. In this case, the digital assistant (and/or a human operator) can update a vocabulary to include the name "Argo" as a movie title. The digital assistant may also or instead update a natural language processing module to associate the phrase "are go" with a movie named "Argo," such that even if a speech input is improperly transcribed as including the phrase "are go," the natural language processing module will identify that the user intended the movie title instead.

[0141] The operations described above with reference to Figures 4A-4C are, optionally, implemented by components depicted in Figure 2 and/or Figures 3A-3B. Similarly, it would be clear to a person having ordinary skill in the art how other processes can be implemented based on the components depicted in Figure 2 and/or Figures 3A-3B.

[0142] It should be understood that the particular order in which the operations have been described above 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.

[0143] 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 invention to the precise forms disclosed. Many modifications and variations are possible view of the above teachings. The implementations were chosen and described in order to best explain the principles of the invention and its practical applications, to thereby enable others skilled in the art to best utilize the invention and various implementations with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A method for operating an intelligent automated assistant, comprising:
at an electronic device with one or more processors and memory storing one or more programs for execution by the one or more processors:
receiving, from a user, a speech input containing a request;
performing at least one action in furtherance of satisfying the request; detecting a user interaction;
determining whether the user interaction is indicative of a problem in the performing of the at least one action;
upon determining that the user interaction is indicative of a problem, storing information relating to the request in a repository for error analysis.

2. The method of claim 1, wherein detecting the user interaction comprises detecting an additional speech input, and determining whether the user interaction is indicative of a problem comprises determining that the additional speech input indicates dissatisfaction with the at least one action.

3. The method of claim 2, wherein determining whether the additional speech input indicates dissatisfaction includes determining whether the additional speech input includes at least one predefined word.

4. The method of any of claims 2-3, wherein determining whether the additional speech input indicates dissatisfaction includes determining a volume of the additional speech input.

5. The method of claim 4, wherein the additional speech input indicates dissatisfaction when the volume of the additional speech input is above a predefined threshold.

6. The method of any of claims 2-5, wherein determining whether the additional speech input indicates dissatisfaction includes determining whether the additional speech input contains prosodic indications of frustration.

7. The method of any of claims 2-6, wherein determining whether the additional speech input indicates dissatisfaction includes determining whether the additional speech input includes substantially the same words as the first speech input.

8. The method of claim 1, wherein detecting the user interaction comprises detecting a

additional speech input and a further speech input, and determining whether the user interaction is indicative of a problem comprises determining that the additional speech input and the further speech input indicate dissatisfaction with the at least one action, wherein determining whether the additional speech input indicates dissatisfaction includes determining that the additional speech input and the further speech input each include substantially similar words as the speech input.

9. The method of claim 1, wherein the user interaction is a predefined motion of the device.

10. The method of claim 1, wherein the user interaction is a selection of an affordance.

11. The method of claim 1, wherein the user interaction is a termination of a dialog session with the intelligent automated assistant.

12. The method of claim 1, wherein the user interaction corresponds to a rejection of a proposed task.

13. The method of any of claims 1-12, further comprising, upon determining that the user interaction is indicative of a problem:

providing a prompt requesting the user to confirm whether there was a problem in the performing of the at least one action; and

receiving, from the user, a confirmation or a disconfirmation of whether there was a problem in the performing of the at least one action.

14. The method of claim 13, further comprising, upon detecting a confirmation that there was a problem in the performing of the at least one action, providing an additional prompt acknowledging that the problem occurred.

15. The method of claim 14, wherein the additional prompt includes a request for the user to restate the speech input containing the request.

16. The method of any of claims 1-15, wherein the repository includes a plurality of entries from a plurality of users, the method further comprising:

analyzing the repository to identify a set of entries, each entry of the set of entries having one or more similar characteristics indicative of an error; and

adjusting one or more of a speech-to-text module and a natural language processing module based on the set of entries so as to reduce reproduction of the error.

17. An electronic device, comprising:

one or more processors;

memory; and

one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions for:

receiving, from a user, a speech input containing a request;

performing at least one action in furtherance of satisfying the request;

detecting a user interaction;

determining whether the user interaction is indicative of a problem in the performing of the at least one action;

upon determining that the user interaction is indicative of a problem, storing information relating to the request in a repository for error analysis.

18. The electronic device of claim 17, further comprising instructions for performing the operations of any of the methods of claims 1-16.

19. A non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by an electronic device with one or more processors and memory, cause the device to:

receive, from a user, a speech input containing a request;

perform at least one action in furtherance of satisfying the request;

detect a user interaction;

determine whether the user interaction is indicative of a problem in the performing of the at least one action;

upon determining that the user interaction is indicative of a problem, store information relating to the request in a repository for error analysis.

20. The non-transitory computer readable storage medium of claim 18, further comprising instructions, which when executed by the electronic device, cause the electronic device to perform the operations of any of the methods of claims 1-16.

21. A method, comprising:

at an electronic device with one or more processors and memory storing one or more programs for execution by the one or more processors:

performing the operations of any of the methods of claims 1-16.

22. An electronic device, comprising:

one or more processors;

memory; and

one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions for performing the operations of any of the methods of claims 1-16.

23. A computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by an electronic device with one or more processors and memory, cause the device to perform the operations of any of the methods of claims 1-16.

24. An electronic device, comprising:

means for performing the operations of any of the methods of claims 1-16.

25. An information processing apparatus for use in an electronic device, comprising: means for performing the operations of any of the methods of claims 1-16.

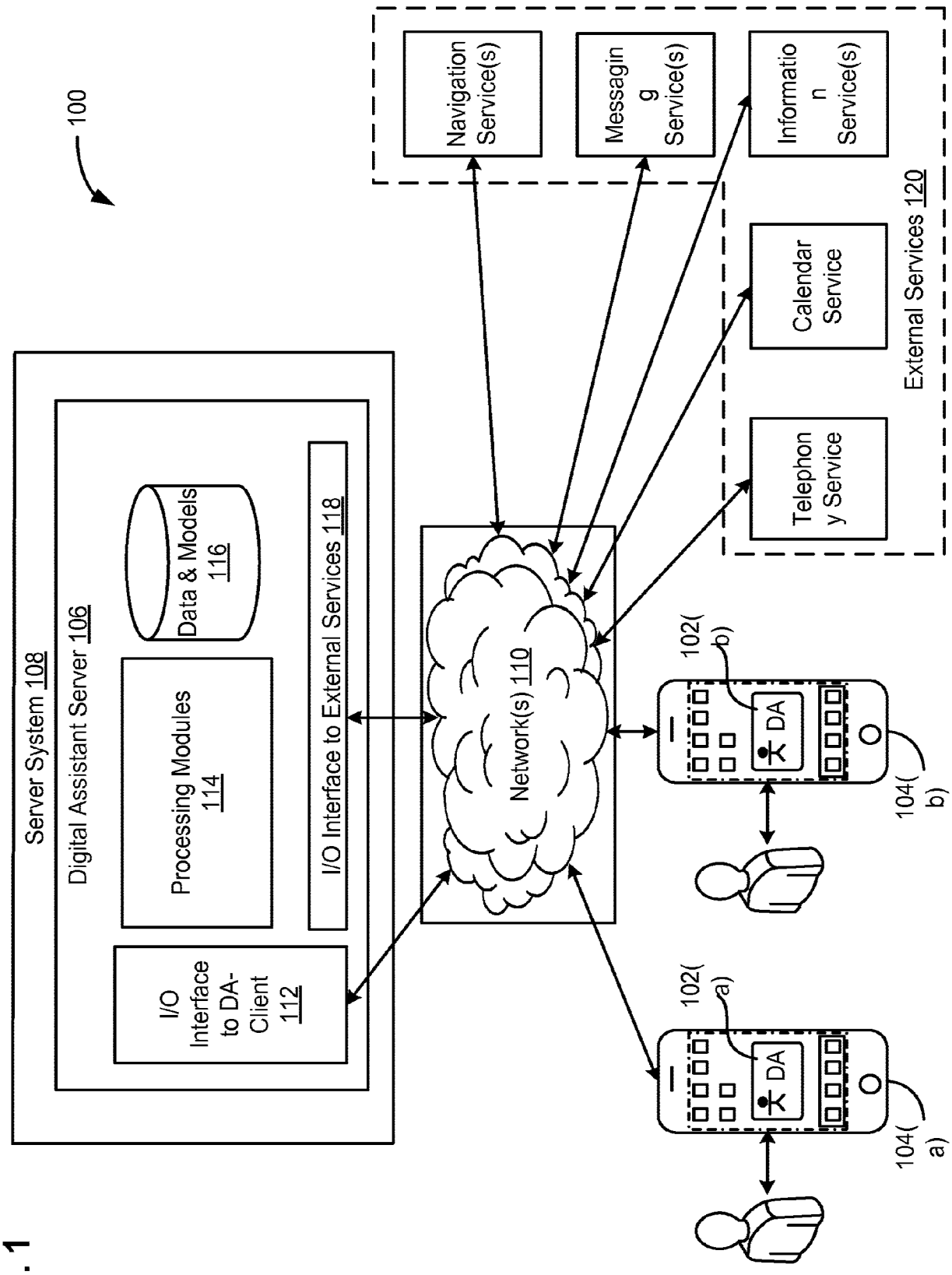

100
Server System 108
Digital Assistant Server 106
Processing Modules 114
Data & Models 116
I/O Interface to DA-Client 112
I/O Interface to External Services 118
Network(s) 110
External Services 120
Navigation Service(s)
Messaging Service(s)
Information Service(s)
Calendar Service
Telephony Service
102(a)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

FIG. 1

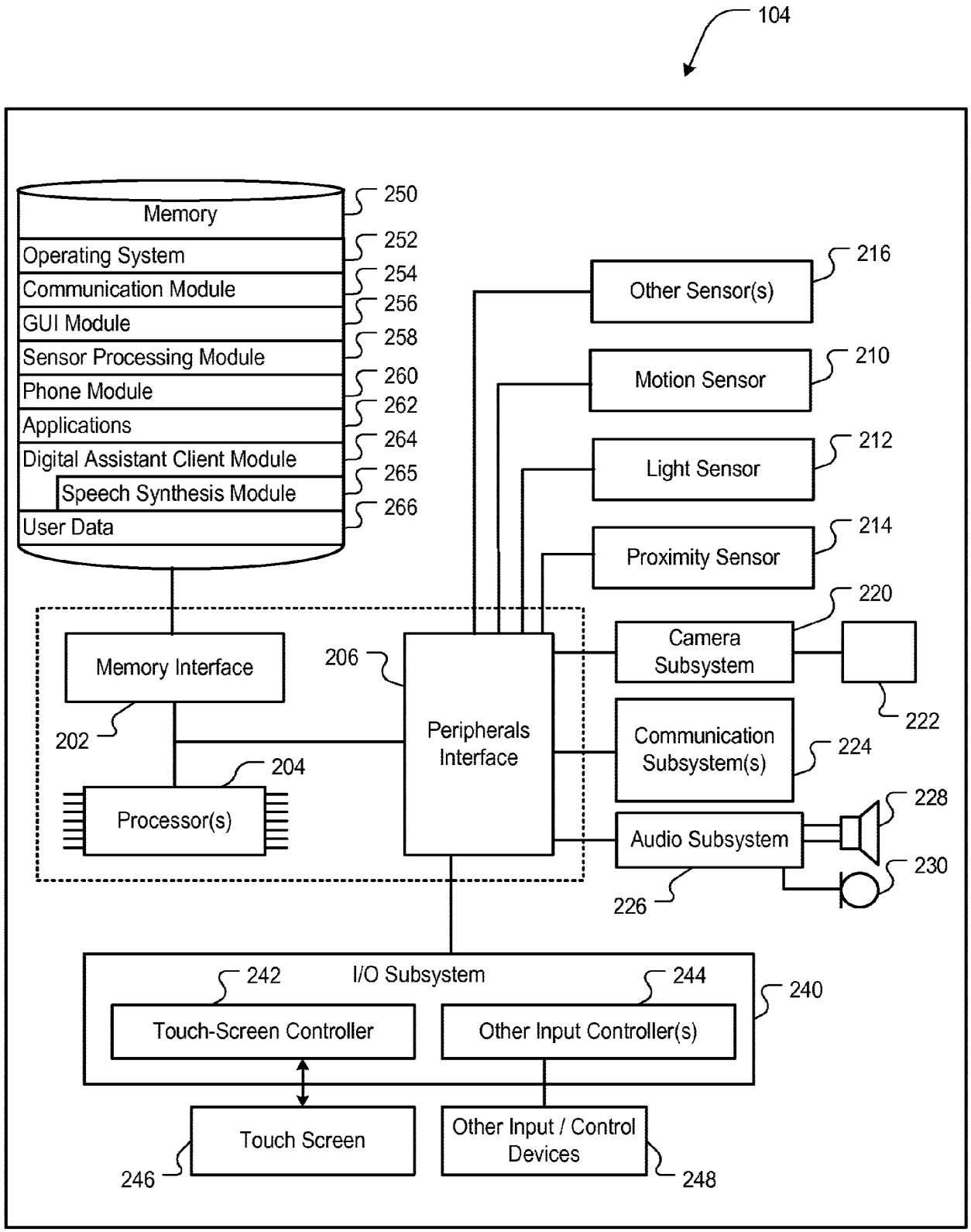

104
Memory
250
Operating System
252
Communication Module
254
GUI Module
256
Sensor Processing Module
258
Phone Module
260
Applications
262
Digital Assistant Client Module
264
Speech Synthesis Module
265
User Data
266
Other Sensor(s)
216
Motion Sensor
210
Light Sensor
212
Proximity Sensor
214
Memory Interface
202
206
Peripherals Interface
Camera Subsystem
220
222
Communication Subsystem(s)
224
Processor(s)
204
Audio Subsystem
228
230
226
I/O Subsystem
240
242
244
Touch-Screen Controller
Other Input Controller(s)
Touch Screen
246
Other Input / Control Devices
248

FIG. 2

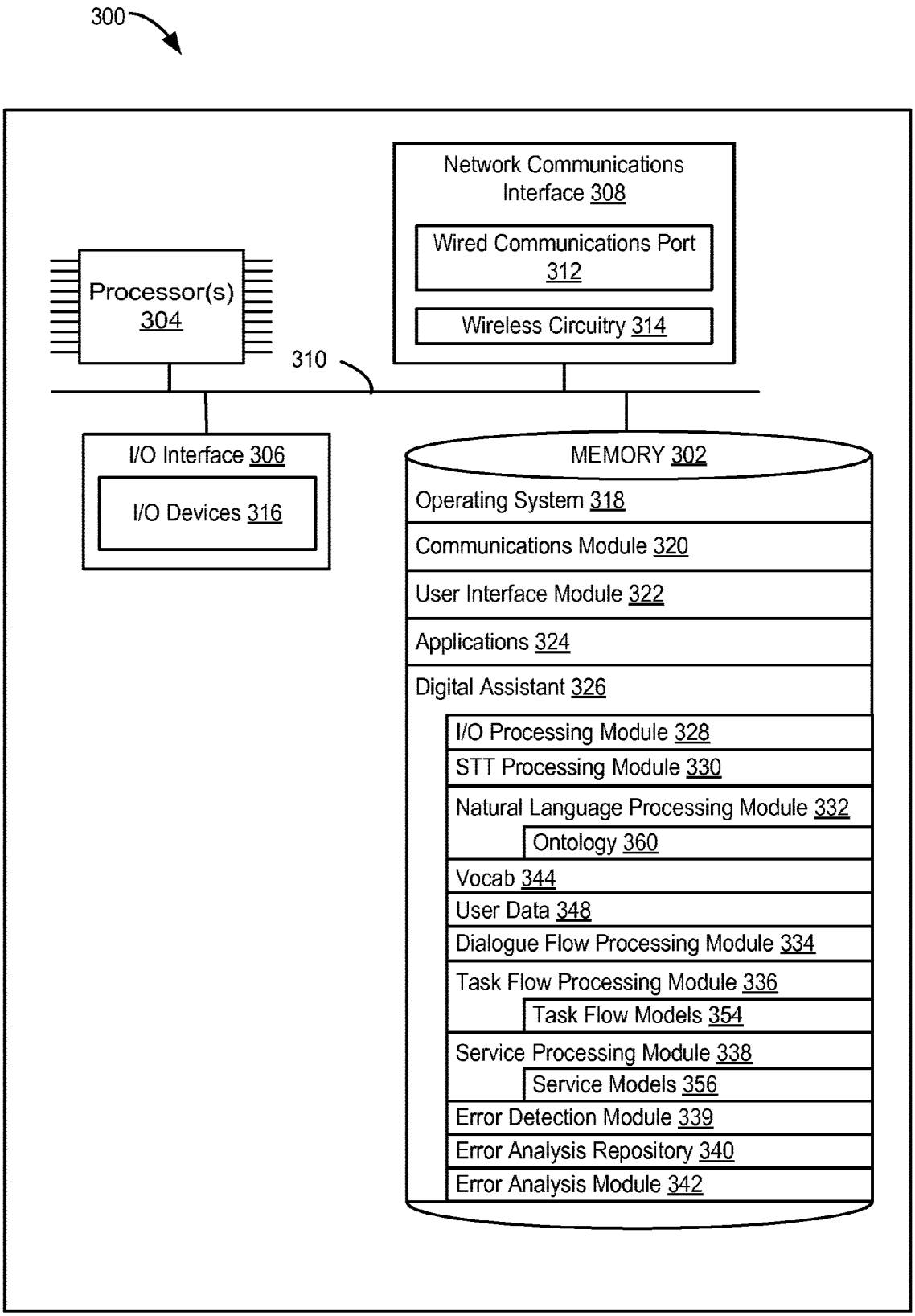
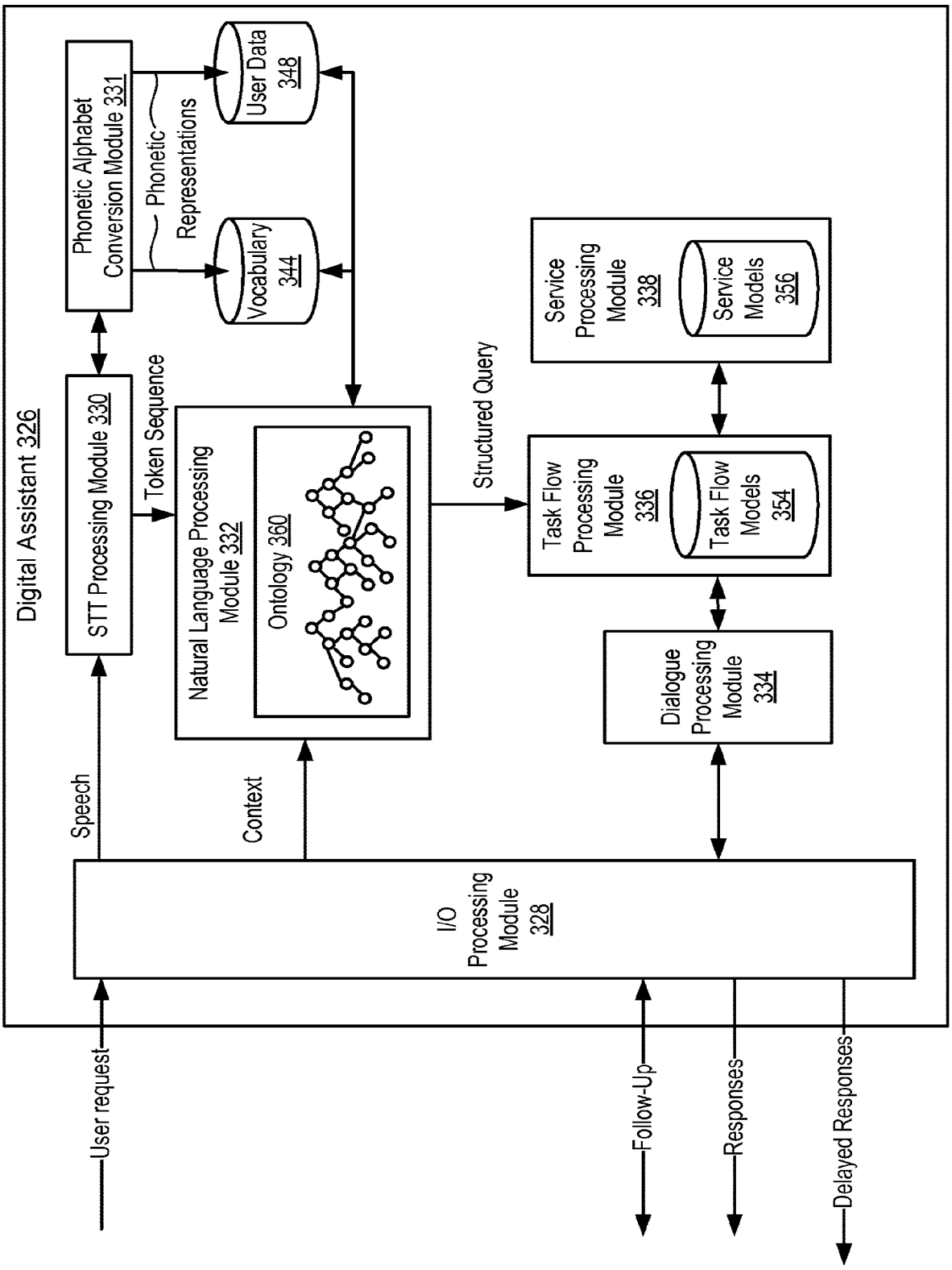

300
Network Communications Interface 308
Wired Communications Port 312
Wireless Circuitry 314
Processor(s) 304
310
I/O Interface 306
I/O Devices 316
MEMORY 302
Operating System 318
Communications Module 320
User Interface Module 322
Applications 324
Digital Assistant 326
I/O Processing Module 328
STT Processing Module 330
Natural Language Processing Module 332
Ontology 360
Vocab 344
User Data 348
Dialogue Flow Processing Module 334
Task Flow Processing Module 336
Task Flow Models 354
Service Processing Module 338
Service Models 356
Error Detection Module 339
Error Analysis Repository 340
Error Analysis Module 342

FIG. 3A

FIG. 3B


Digital Assistant 326
STT Processing Module 330
Token Sequence
Natural Language Processing Module 332
Ontology 360
Structured Query
Phonetic Alphabet Conversion Module 331
Phonetic Representations
User Data 348
Vocabulary 344
Service Processing Module 338
Service Models 356
Task Flow Processing Module 336
Task Flow Models 354
Dialogue Processing Module 334
I/O Processing Module 328
Speech
Context
User request
Follow-Up
Responses
Delayed Responses

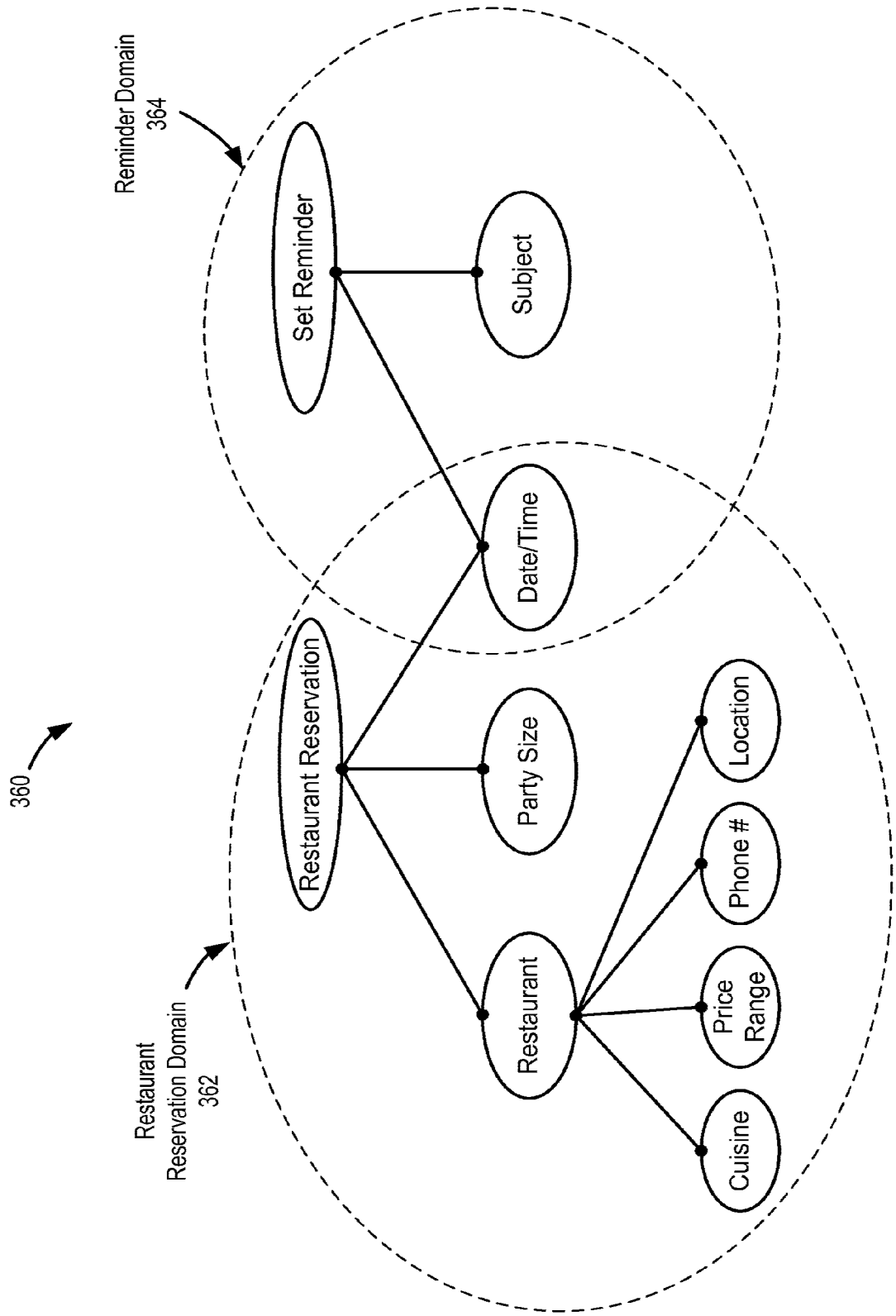

360
Reminder Domain
364
Set Reminder
Subject
Date/Time
Restaurant Reservation
Party Size
Restaurant
Location
Phone #
Price Range
Cuisine
Restaurant Reservation Domain
362

FIG. 3C

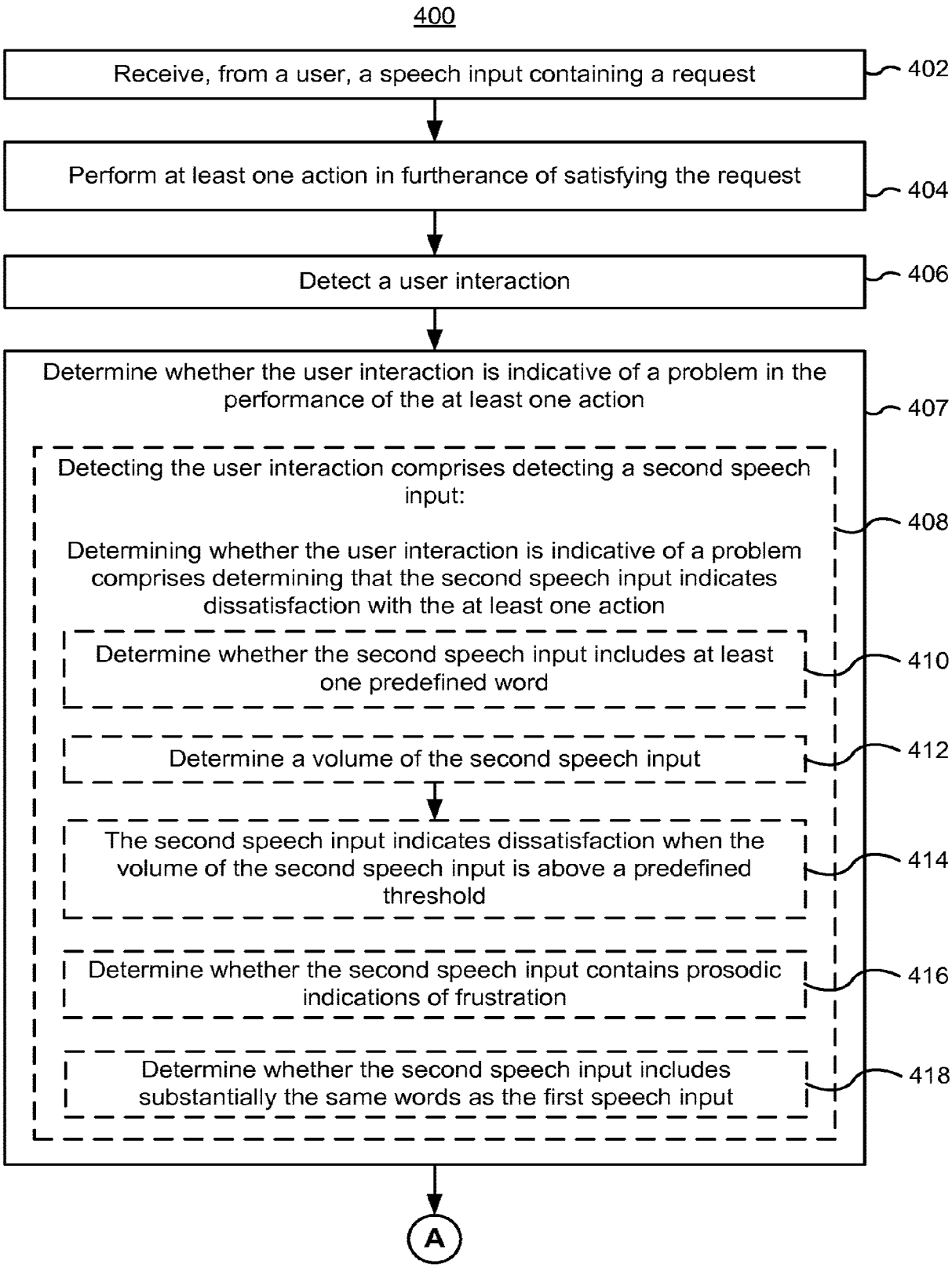

400
Receive, from a user, a speech input containing a request
402
Perform at least one action in furtherance of satisfying the request
404
Detect a user interaction
406
Determine whether the user interaction is indicative of a problem in the performance of the at least one action
407
Detecting the user interaction comprises detecting a second speech input:
Determining whether the user interaction is indicative of a problem comprises determining that the second speech input indicates dissatisfaction with the at least one action
408
Determine whether the second speech input includes at least one predefined word
410
Determine a volume of the second speech input
412
The second speech input indicates dissatisfaction when the volume of the second speech input is above a predefined threshold
414
Determine whether the second speech input contains prosodic indications of frustration
416
Determine whether the second speech input includes substantially the same words as the first speech input
418
A

FIG. 4A

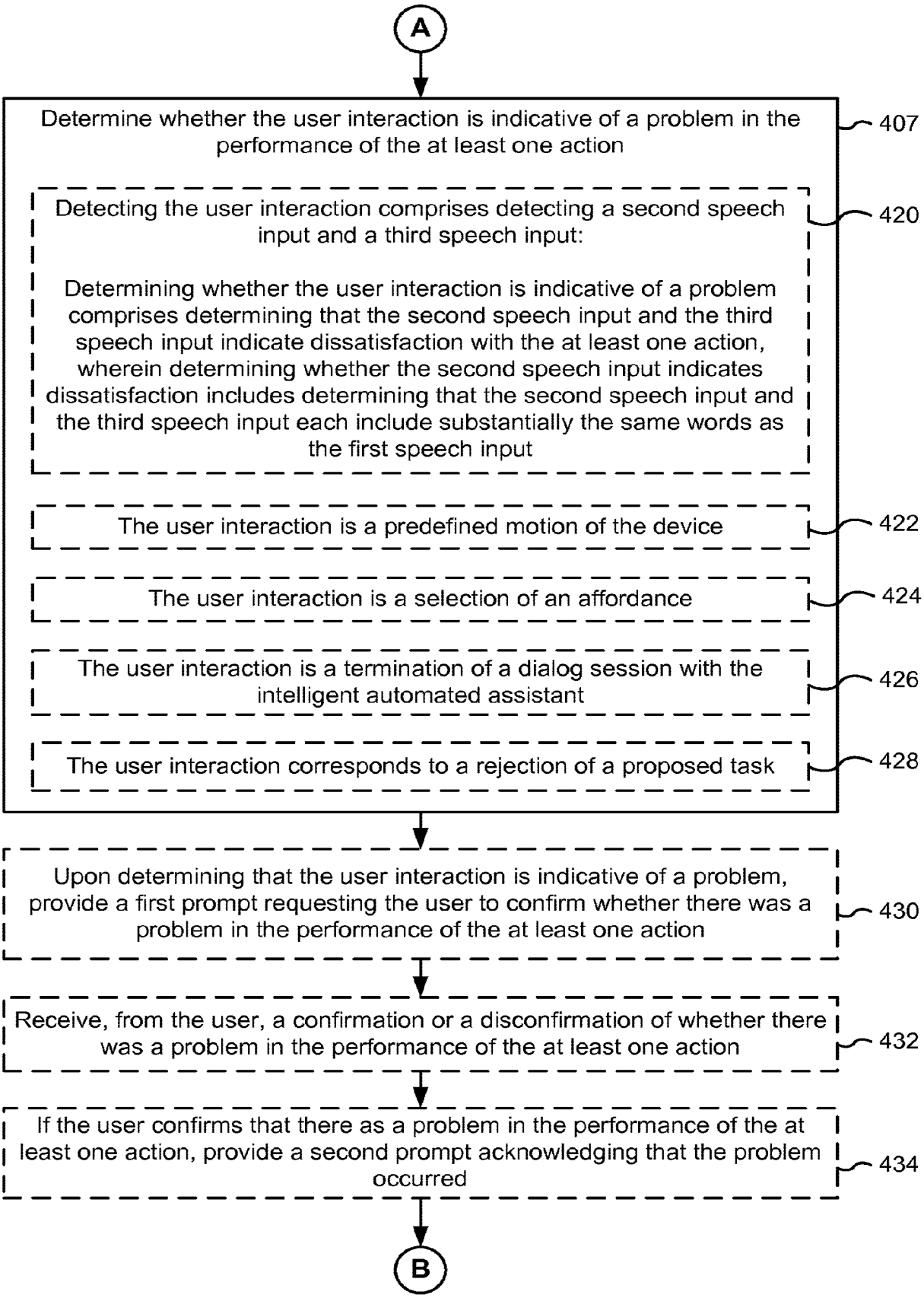

A
Determine whether the user interaction is indicative of a problem in the performance of the at least one action
407
Detecting the user interaction comprises detecting a second speech input and a third speech input:
Determining whether the user interaction is indicative of a problem comprises determining that the second speech input and the third speech input indicate dissatisfaction with the at least one action, wherein determining whether the second speech input indicates dissatisfaction includes determining that the second speech input and the third speech input each include substantially the same words as the first speech input
420
The user interaction is a predefined motion of the device
422
The user interaction is a selection of an affordance
424
The user interaction is a termination of a dialog session with the intelligent automated assistant
426
The user interaction corresponds to a rejection of a proposed task
428
Upon determining that the user interaction is indicative of a problem, provide a first prompt requesting the user to confirm whether there was a problem in the performance of the at least one action
430
Receive, from the user, a confirmation or a disconfirmation of whether there was a problem in the performance of the at least one action
432
If the user confirms that there as a problem in the performance of the at least one action, provide a second prompt acknowledging that the problem occurred
434
B

FIG. 4B

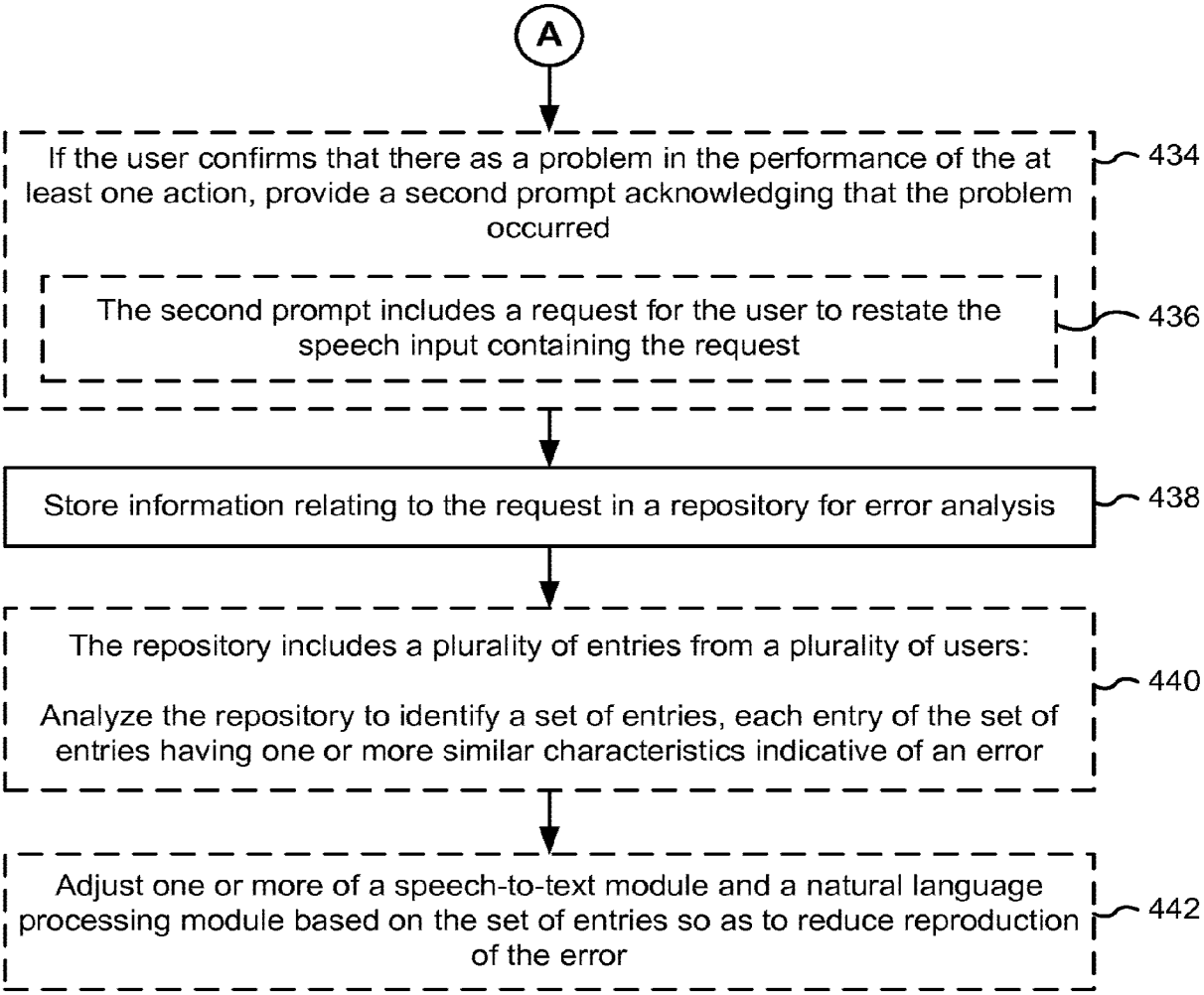

A
If the user confirms that there as a problem in the performance of the at least one action, provide a second prompt acknowledging that the problem occurred
434
The second prompt includes a request for the user to restate the speech input containing the request
436
Store information relating to the request in a repository for error analysis
438
The repository includes a plurality of entries from a plurality of users:
Analyze the repository to identify a set of entries, each entry of the set of entries having one or more similar characteristics indicative of an error
440
Adjust one or more of a speech-to-text module and a natural language processing module based on the set of entries so as to reduce reproduction of the error
442

FIG. 4C

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/040401

A. CLASSIFICATION OF SUBJECT MATTER
INV. G10L15/22 H04M1/27
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)
G10L G06F 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, INSPEC, 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	Stephen Choularton ET AL: "User Responses to Speech Recognition Errors: Consistency of Behaviour Across Domains", Proceedings of the 10th Australian International Conference on Speech Science & Technology, 10 December 2004 (2004-12-10), XP055135887, Macquarie University, Sydney, December 8 to 10, 2004. Retrieved from the Internet: URL:http://www.assta.org/sst/2004/proceedings/papers/sst2004-357.pdf [retrieved on 2014-08-21]	1-3,6-8, 10-25
A	abstract Section 1 "Introduction" lines 16-20 Section1 "Introduction" lines 50-54 Section 2 "Related work" lines 16-25 Section2 "Related Work" lines 41-49	9

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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	Date of mailing of the international search report
25 August 2014	04/09/2014

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Van Doremalen, Hans

3

Form PCT/ISA/210 (second sheet) (April 2005)

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	page 459, left-hand column, line 20 - line 26 page 460, left-hand column, line 9 - line 14	

X	KAZEMZADEH A ET AL: "Acoustic correlates of user response to error in human-computer dialogues", AUTOMATIC SPEECH RECOGNITION AND UNDERSTANDING, 2003. ASRU '03. 2003 I EEE WORKSHOP ON ST. THOMAS, VI, USA NOV. 30-DEC. 3, 2003, PISCATAWAY, NJ, USA,IEEE, 30 November 2003 (2003-11-30), pages 215-220, XP010713315, DOI: 10.1109/ASRU.2003.1318443 ISBN: 978-0-7803-7980-0	1,2,4-8, 10,11, 16-25
A	abstract Section 1. "Introduction" lines 10-13 page 215, line 22 - line 28 Section 2.2 "Tagging" lines 9-16 Section 3.1 "Acoustic Measurements" lines 1-6	9

X	EP 1 107 229 A2 (COMVERSE NETWORK SYST INC [US]) 13 June 2001 (2001-06-13) abstract; claims 35-37	1

A	US 2012/310652 A1 (O'SULLIVAN DANIEL [US]) 6 December 2012 (2012-12-06) abstract	1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2014/040401

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1107229 A2	13-06-2001	EP 1107229 A2	13-06-2001
		US 6526382 B1	25-02-2003
		US 2002198722 A1	26-12-2002
		US 2003004719 A1	02-01-2003
		US 2003004730 A1	02-01-2003
		US 2003004731 A1	02-01-2003
		US 2003046086 A1	06-03-2003
		US 2003046088 A1	06-03-2003
US 2012310652 A1	06-12-2012	NONE	