US 20160260433A1

(19) United States
(12) Patent Application Publication
SUMNER et al.
(10) Pub. No.: US 2016/0260433 A1
(43) Pub. Date: Sep. 8, 2016

(54) **STRUCTURED DICTATION USING INTELLIGENT AUTOMATED ASSISTANTS**

(71) Applicant: **Apple Inc.**, Cupertino, CA (US)

(72) Inventors: **Michael R. SUMNER**, Cupertino, CA (US); **Brandon J. NEWENDORP**, Cupertino, CA (US); **Ryan M. ORR**, Cupertino, CA (US)

(73) Assignee: **APPLE INC.**, Cupertino, CA (US)

(21) Appl. No.: **14/839,826**

(22) Filed: **Aug. 28, 2015**

Related U.S. Application Data

(60) Provisional application No. 62/129,441, filed on Mar. 6, 2015.

Publication Classification

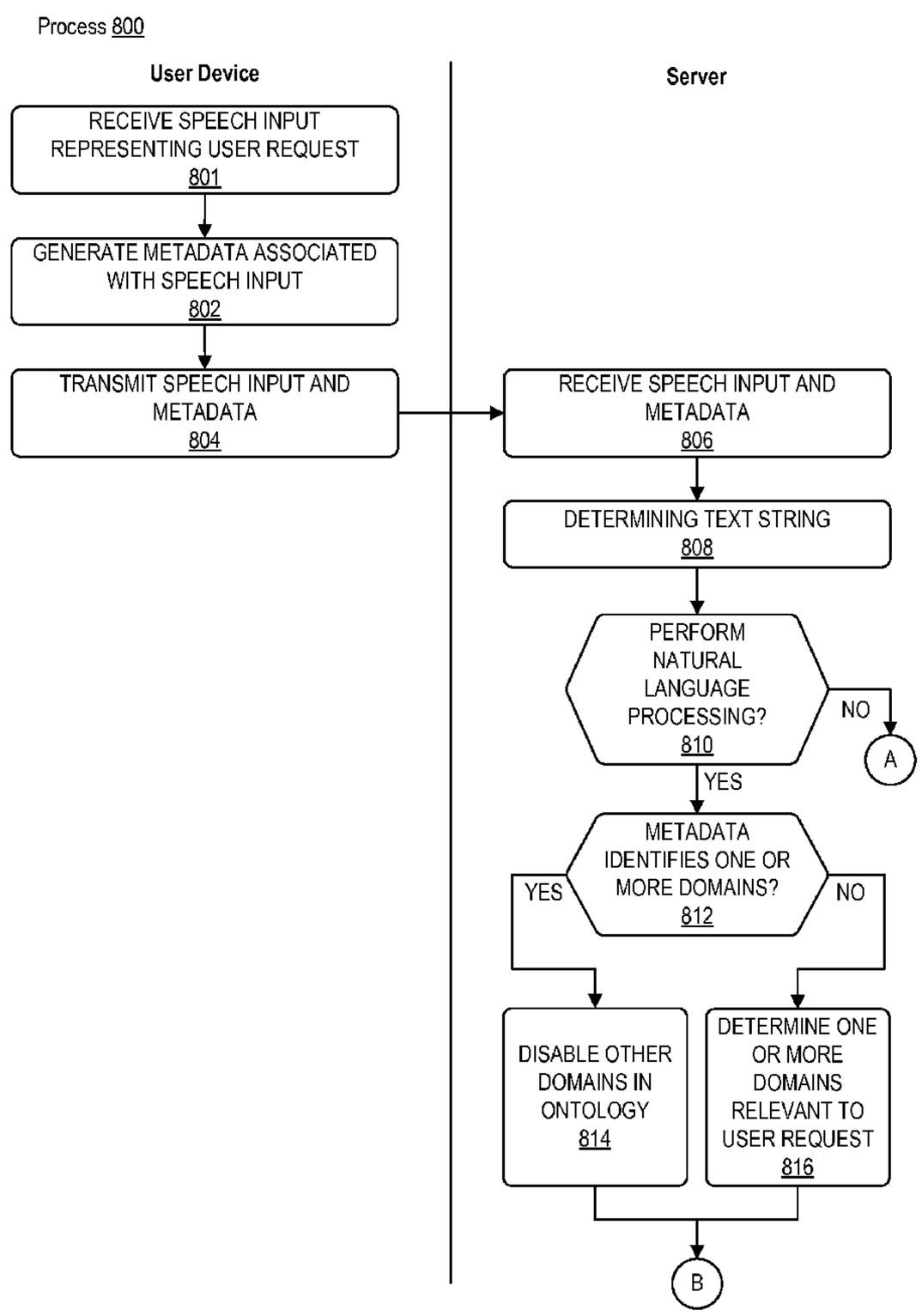
(51) **Int. Cl.**

G10L 15/26	(2006.01)
G10L 15/01	(2006.01)
G10L 25/72	(2006.01)
G06F 17/25	(2006.01)

(52) **U.S. Cl.**
CPC ***G10L 15/265*** (2013.01); ***G06F 17/25*** (2013.01); ***G10L 15/01*** (2013.01); ***G10L 25/72*** (2013.01)

(57) **ABSTRACT**

Systems and processes for structured dictation using intelligent automated assistants are provided. In one example process, a speech input representing a user request can be received. In addition, metadata associated with the speech input can be received. A text string corresponding to the speech input can be determined. The process can determine whether to perform natural language processing on the text string and whether the metadata identifies one or more domains corresponding to the user request. In response to the determination that natural language processing is to be performed on the text string and that the metadata identifies one or more domains corresponding to the user request, natural language processing of the text string can be constrained to the one or more domains. A result can be obtained based on the one or more domains and the result can be outputted from the electronic device.



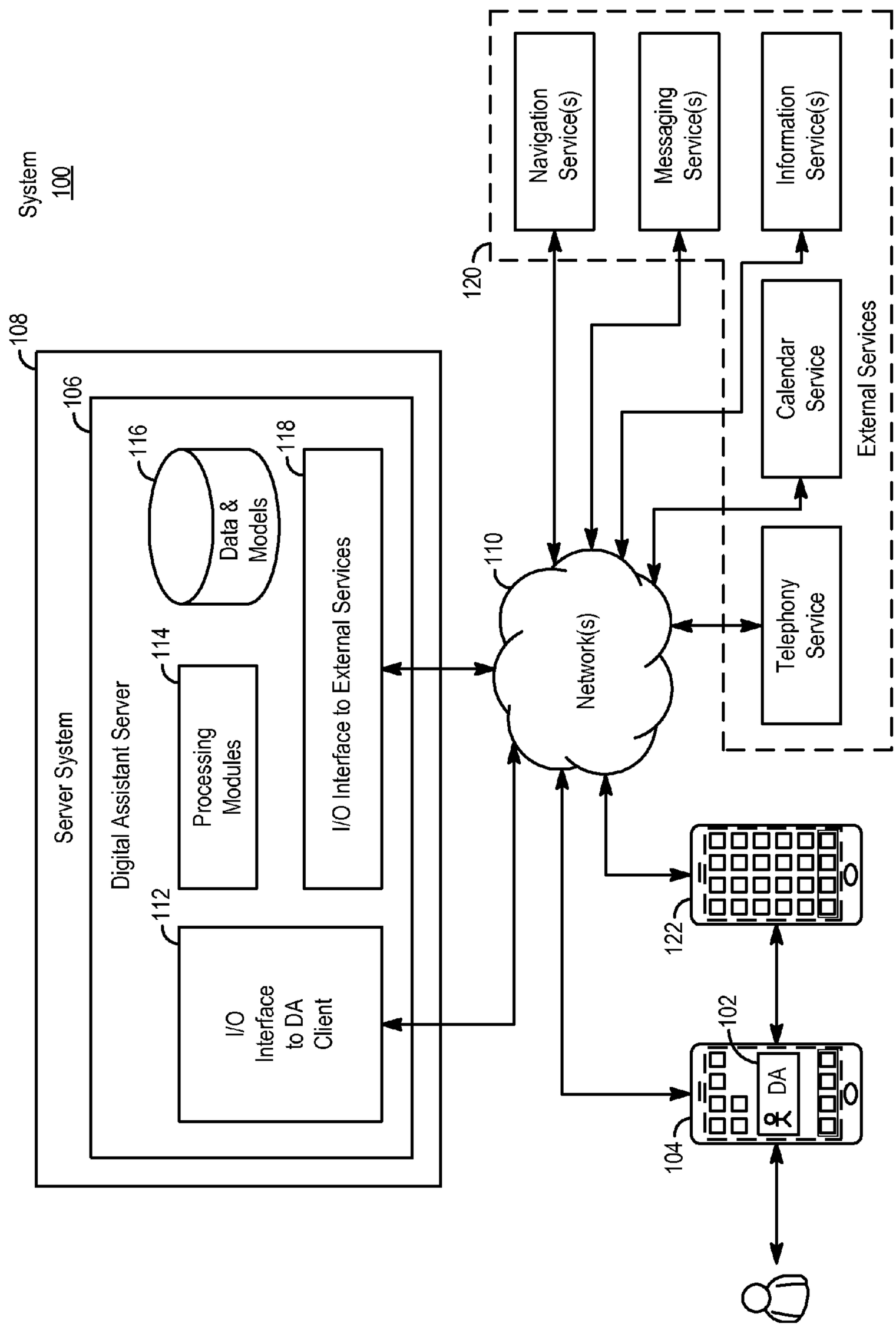

System
100
108
Server System
106
Digital Assistant Server
116
Data & Models
118
I/O Interface to External Services
114
Processing Modules
112
I/O Interface to DA Client
110
Network(s)
120
Navigation Service(s)
Messaging Service(s)
Information Service(s)
Calendar Service
Telephony Service
External Services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

FIG. 1

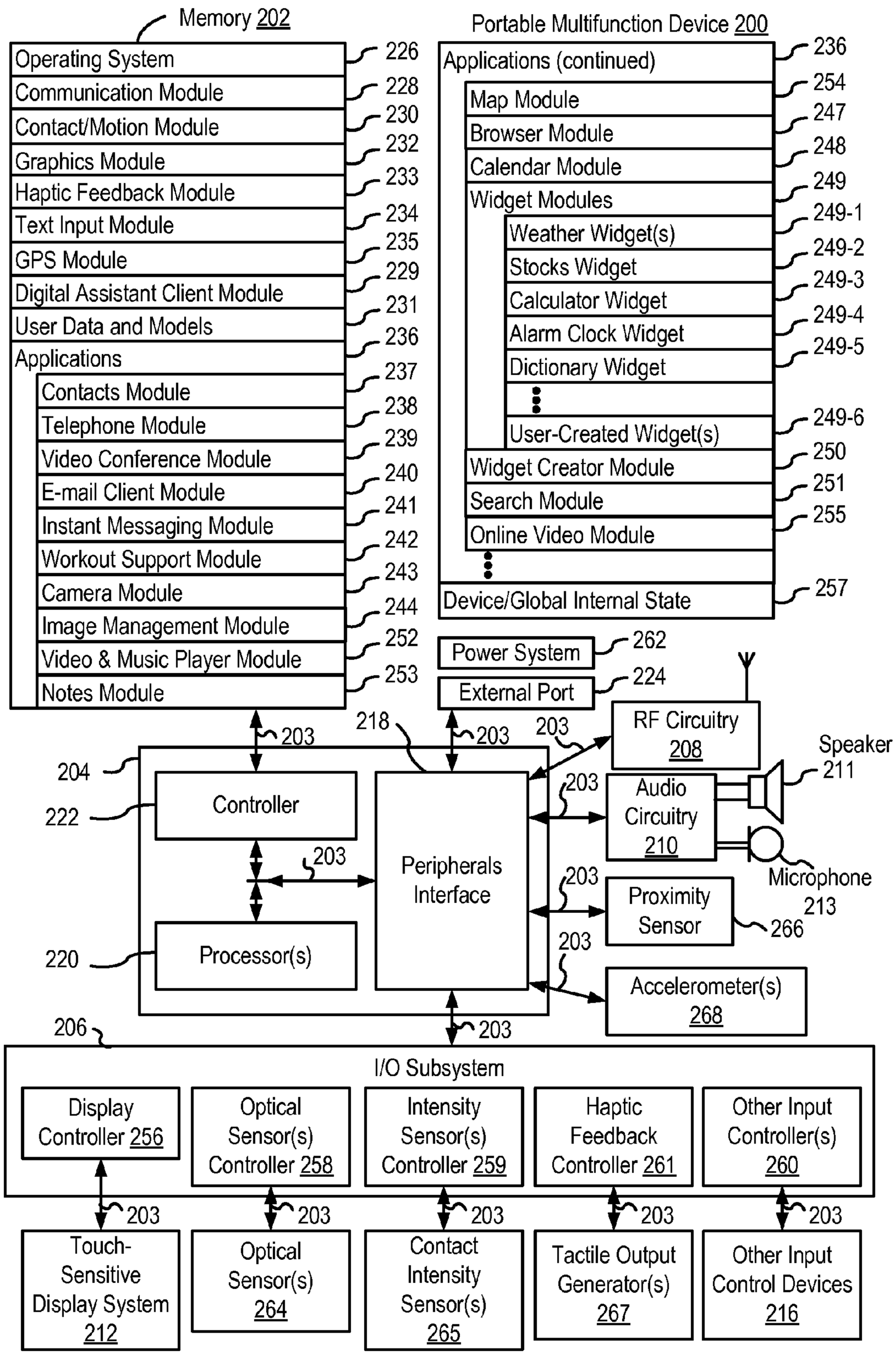

Memory 202
Operating System 226
Communication Module 228
Contact/Motion Module 230
Graphics Module 232
Haptic Feedback Module 233
Text Input Module 234
GPS Module 235
Digital Assistant Client Module 229
User Data and Models 231
Applications 236
Contacts Module 237
Telephone Module 238
Video Conference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

FIG. 2A

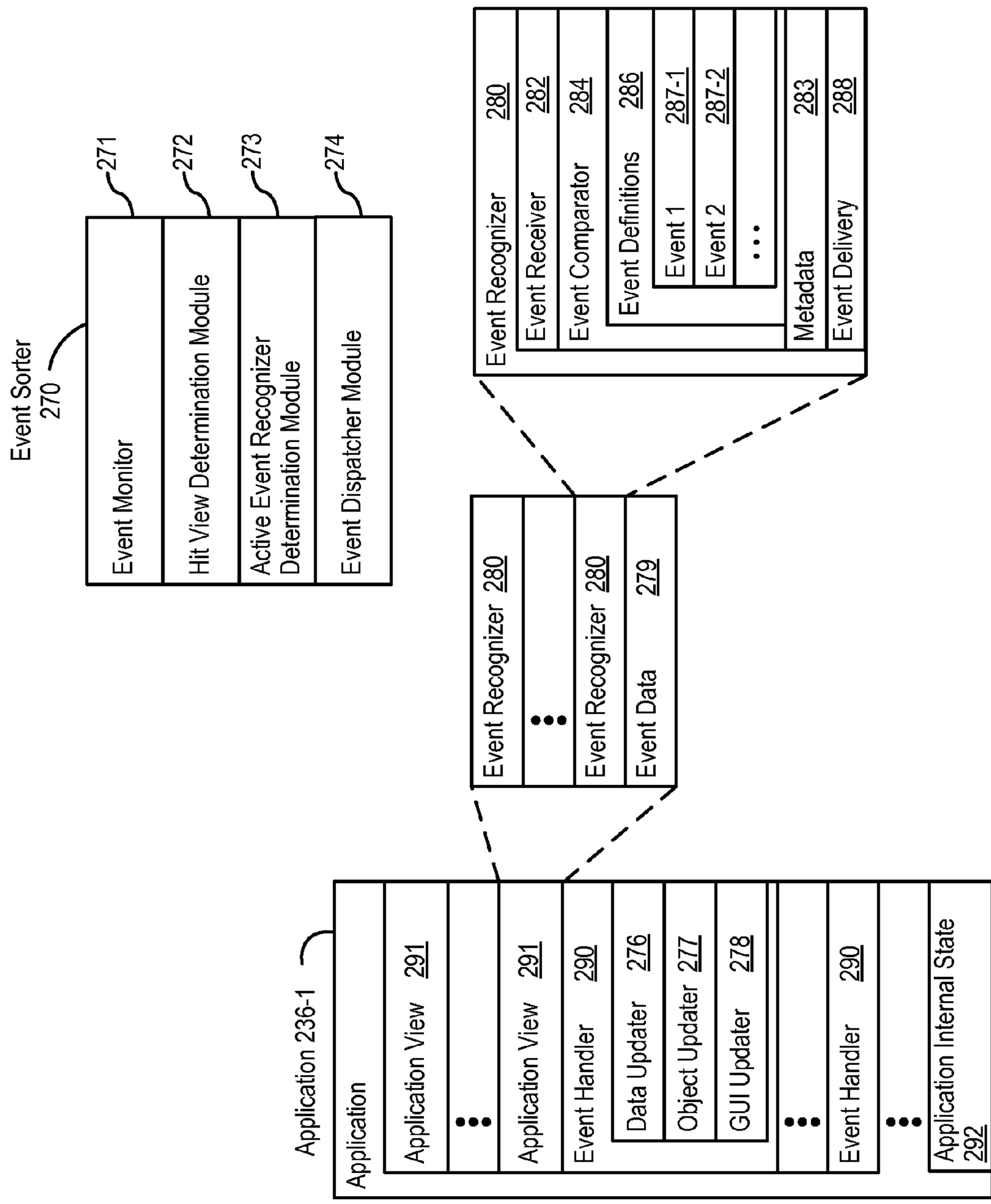

Event Sorter
270
Event Monitor
271
Hit View Determination Module
272
Active Event Recognizer Determination Module
273
Event Dispatcher Module
274
Event Recognizer
280
Event Receiver
282
Event Comparator
284
Event Definitions
286
Event 1
287-1
Event 2
287-2
...
Metadata
283
Event Delivery
288
Event Recognizer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

FIG. 2B

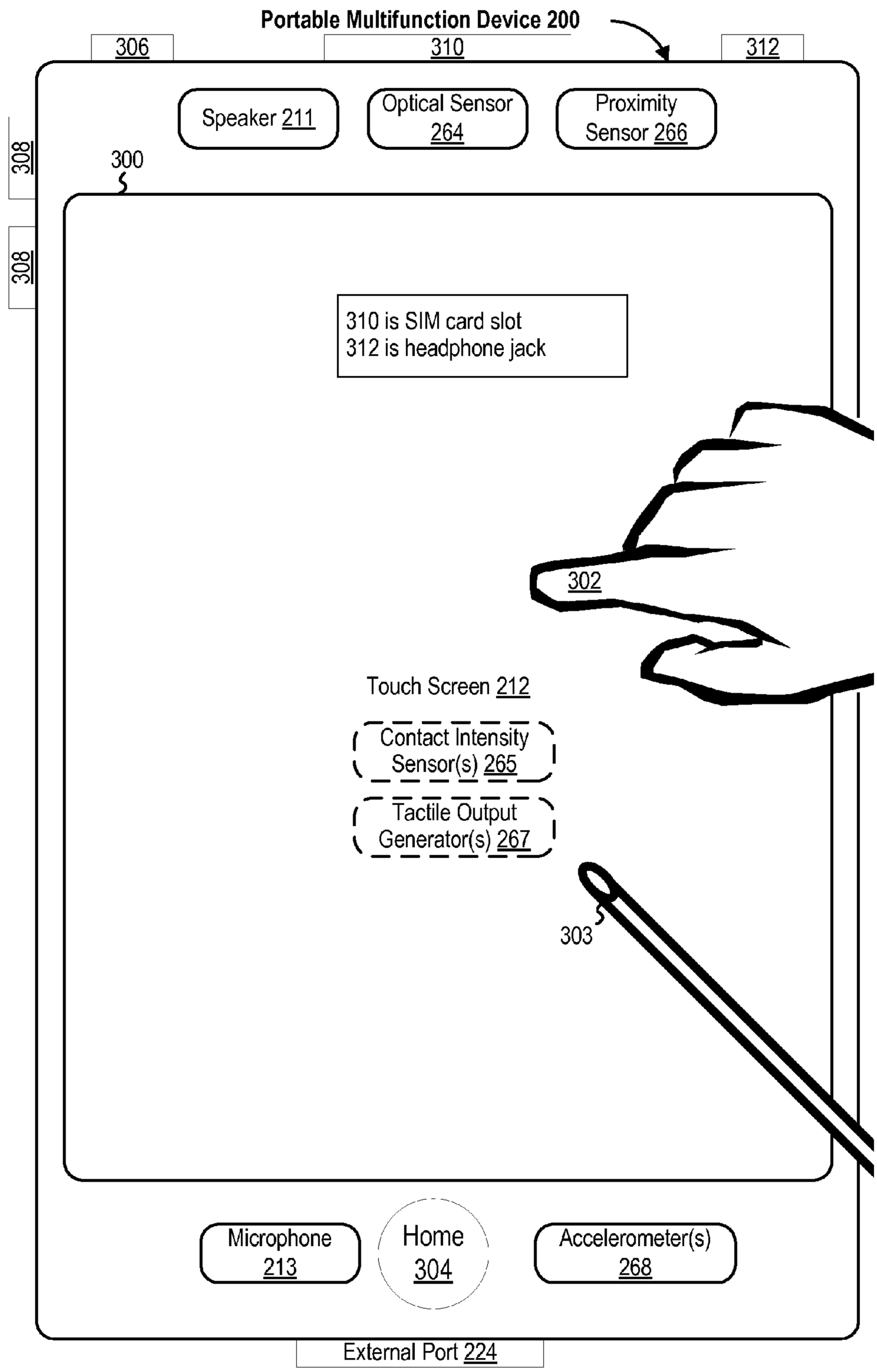

Portable Multifunction Device 200
306
310
312
Speaker 211
Optical Sensor 264
Proximity Sensor 266
308
300
308
310 is SIM card slot
312 is headphone jack
302
Touch Screen 212
Contact Intensity Sensor(s) 265
Tactile Output Generator(s) 267
303
Microphone 213
Home 304
Accelerometer(s) 268
External Port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

FIG. 3

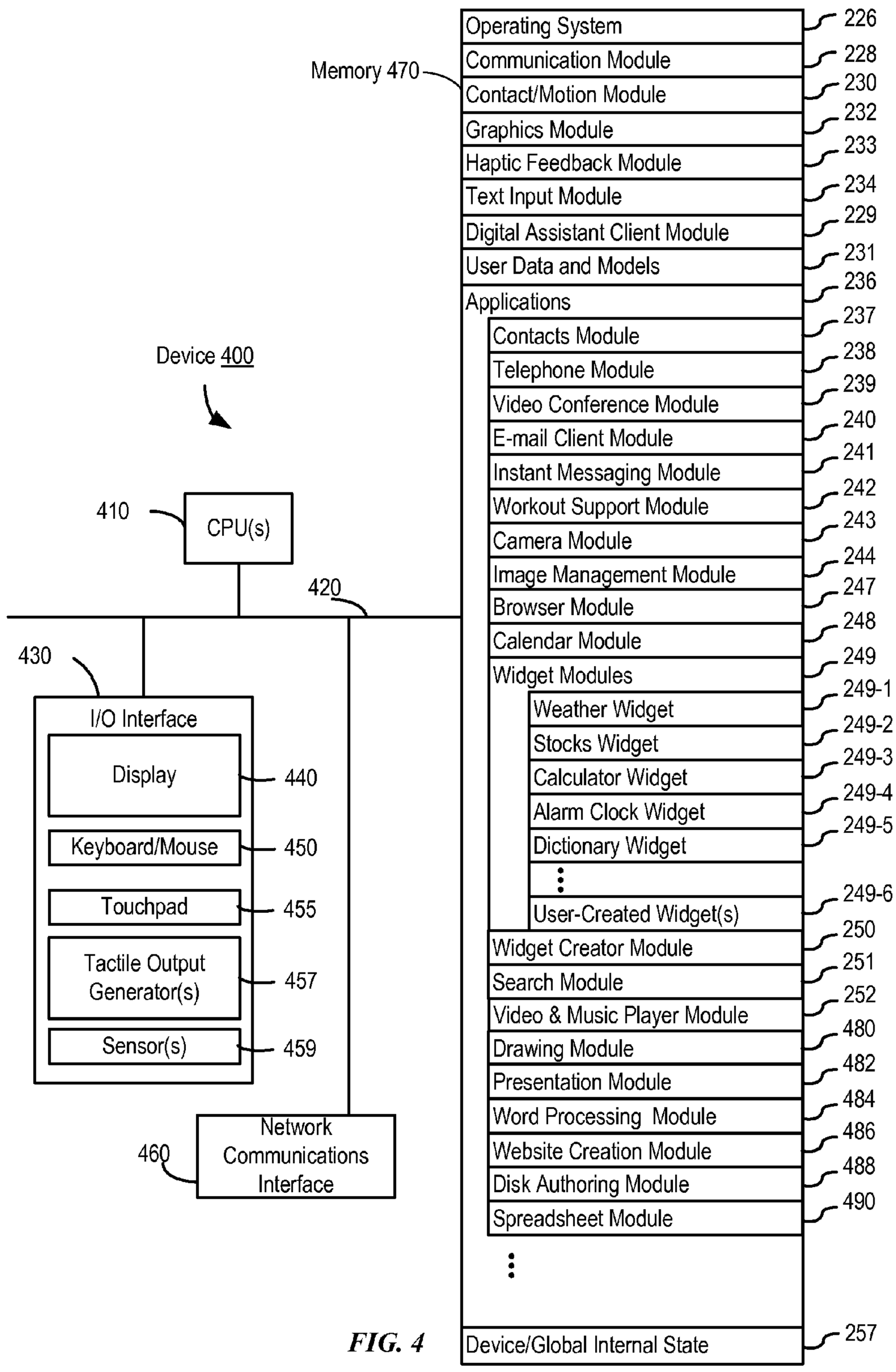

Device 400
410
CPU(s)
420
430
I/O Interface
Display
440
Keyboard/Mouse
450
Touchpad
455
Tactile Output Generator(s)
457
Sensor(s)
459
460
Network Communications Interface
Memory 470
Operating System
226
Communication Module
228
Contact/Motion Module
230
Graphics Module
232
Haptic Feedback Module
233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

FIG. 4

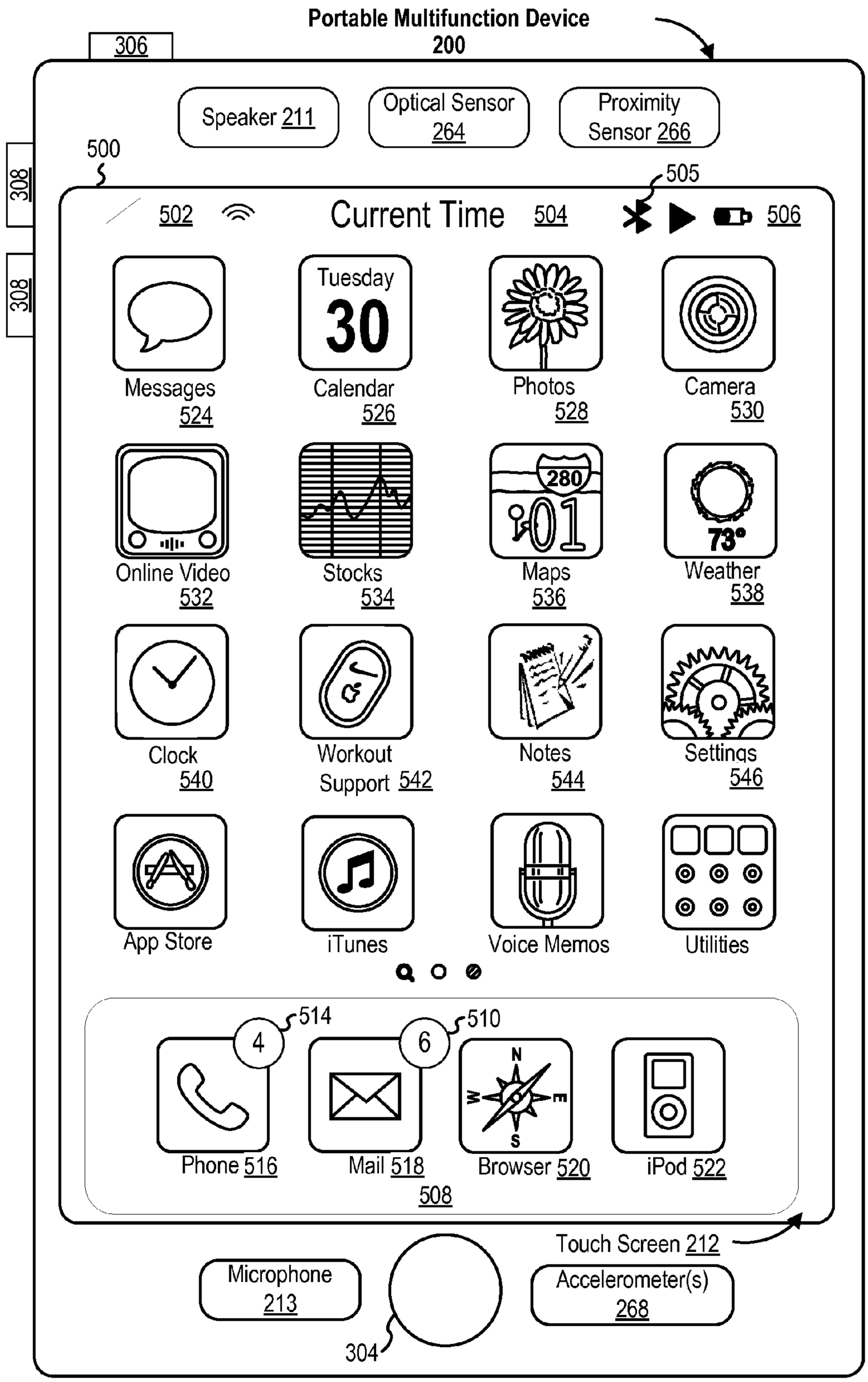

Portable Multifunction Device
200
306
Speaker 211
Optical Sensor 264
Proximity Sensor 266
500
308
308
505
502
Current Time 504
506
Tuesday
30
Messages 524
Calendar 526
Photos 528
Camera 530
Online Video 532
Stocks 534
Maps 536
Weather 538
Clock 540
Workout Support 542
Notes 544
Settings 546
App Store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

FIG. 5A

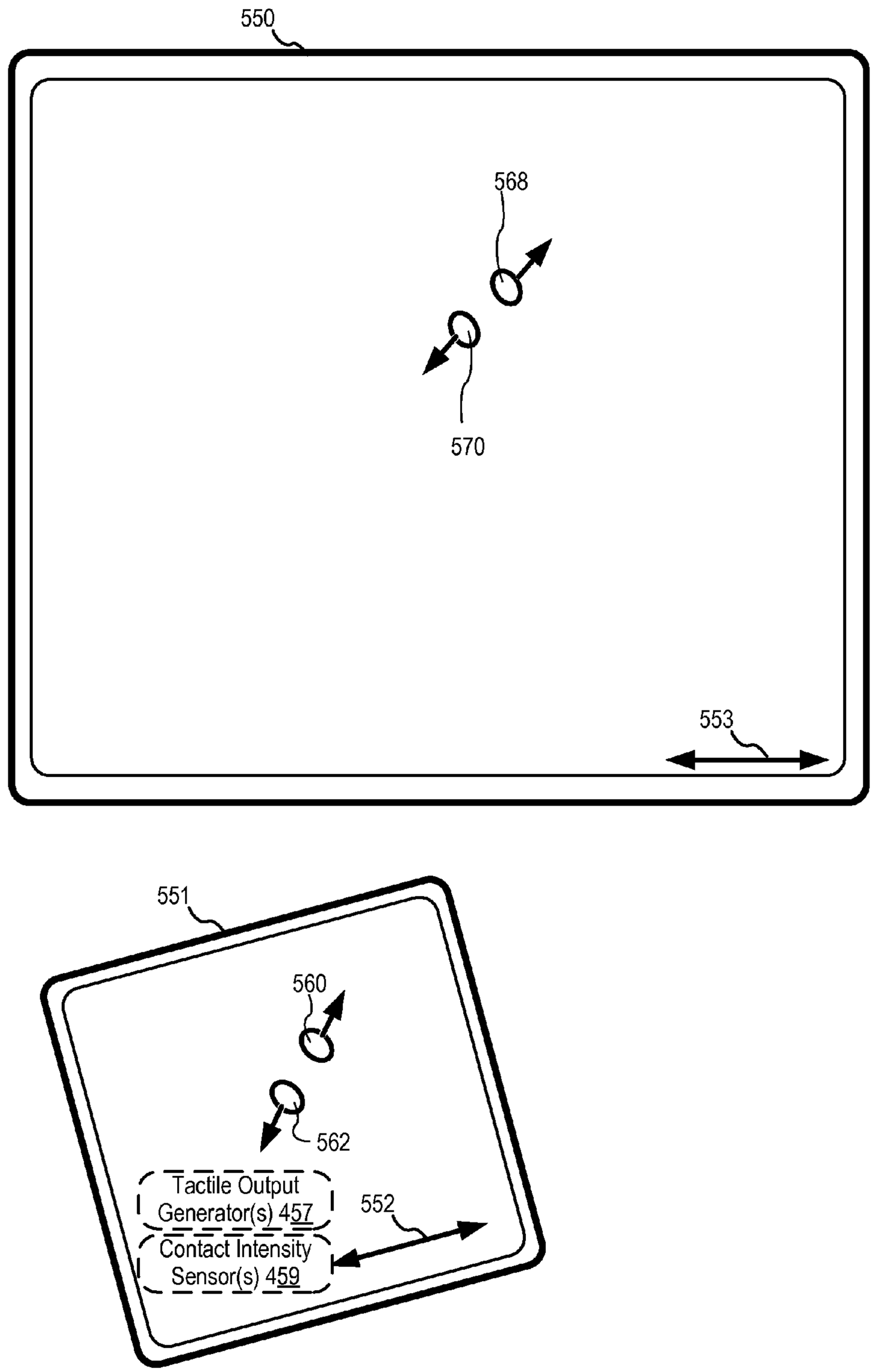

550
568
570
553
551
560
562
Tactile Output Generator(s) 457
552
Contact Intensity Sensor(s) 459

FIG. 5B

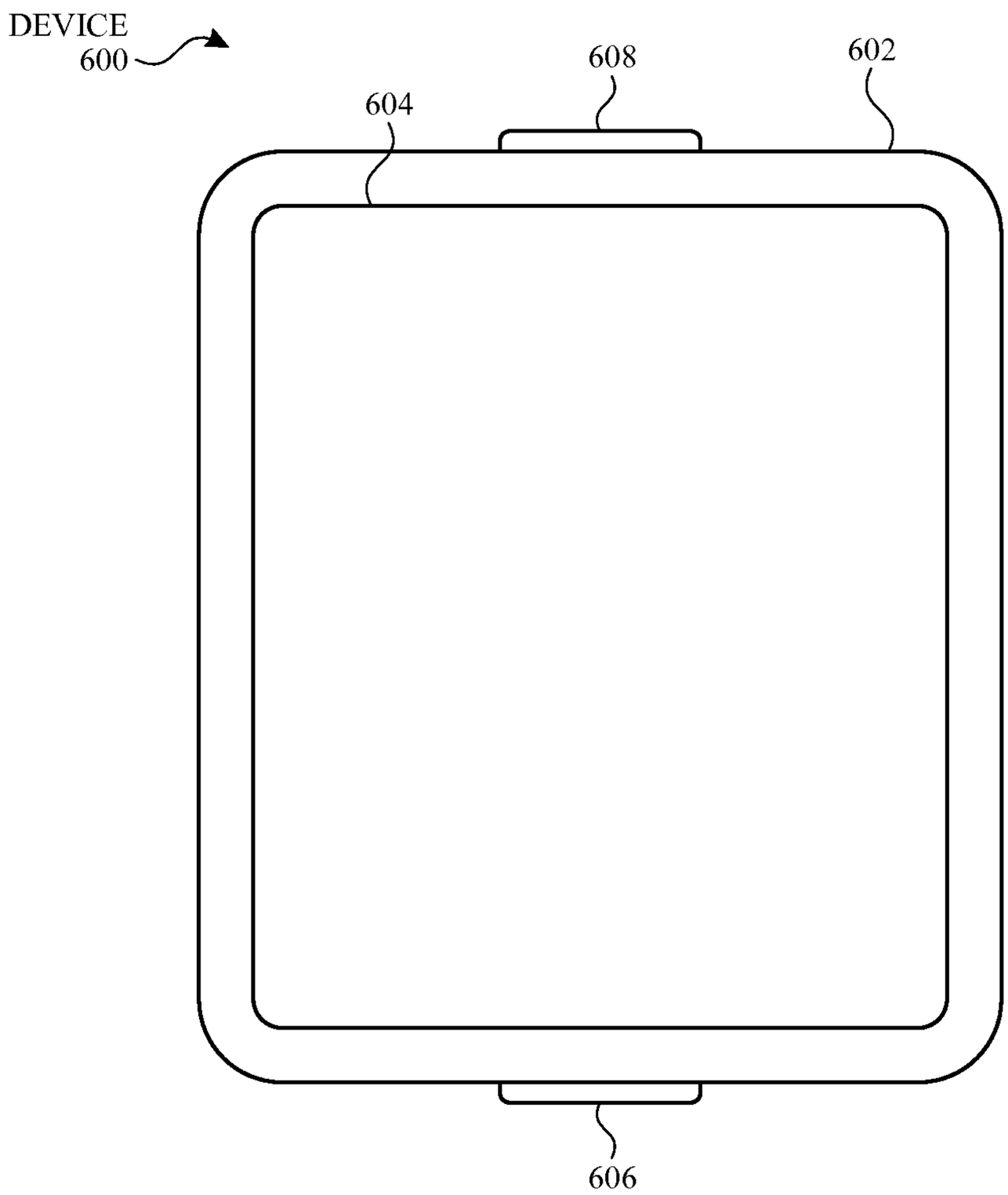


FIG. 6A

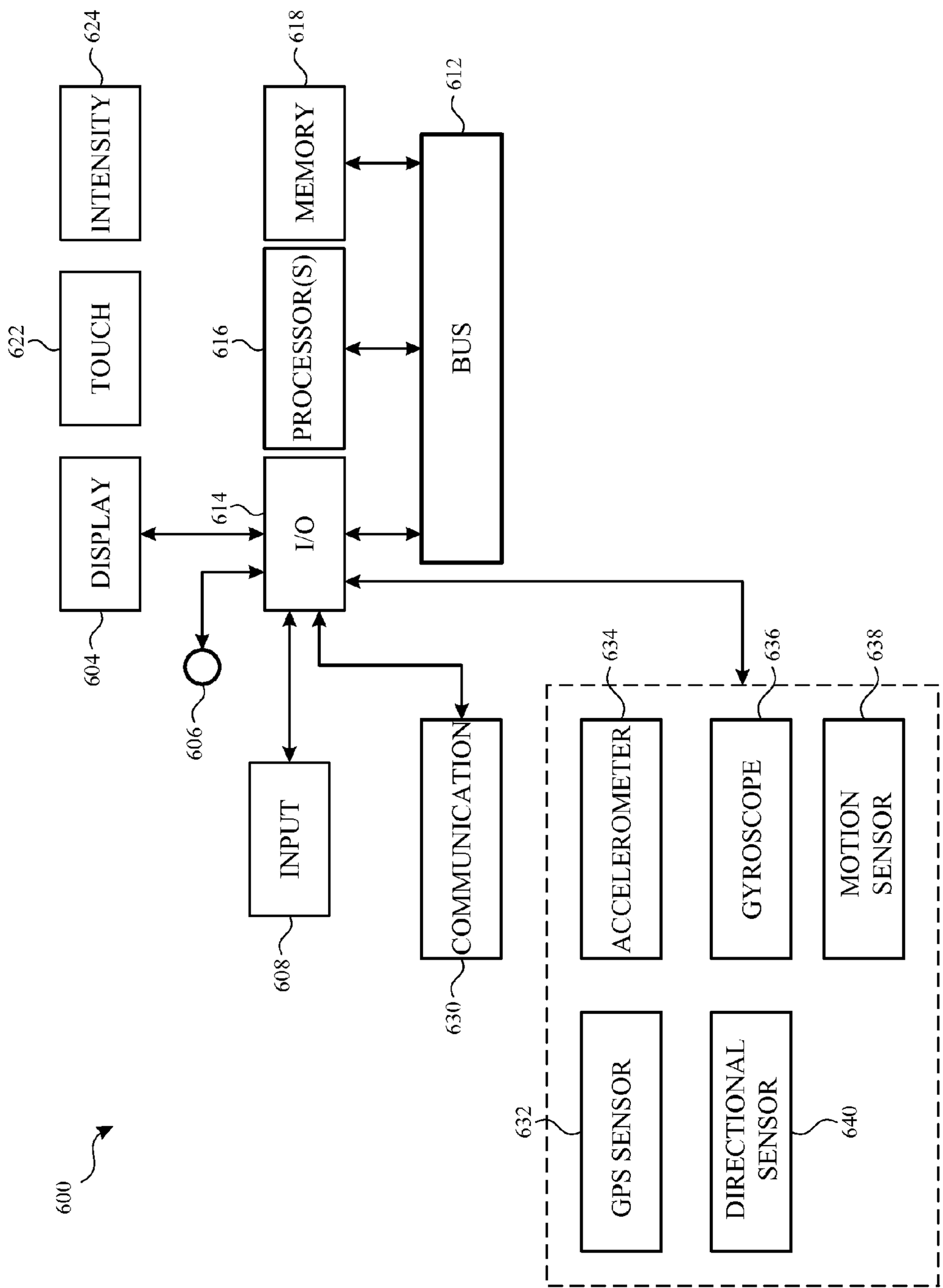

600
604
DISPLAY
622
TOUCH
624
INTENSITY
606
608
INPUT
614
I/O
616
PROCESSOR(S)
618
MEMORY
612
BUS
630
COMMUNICATION
632
GPS SENSOR
634
ACCELEROMETER
636
GYROSCOPE
638
MOTION SENSOR
640
DIRECTIONAL SENSOR

FIG. 6B

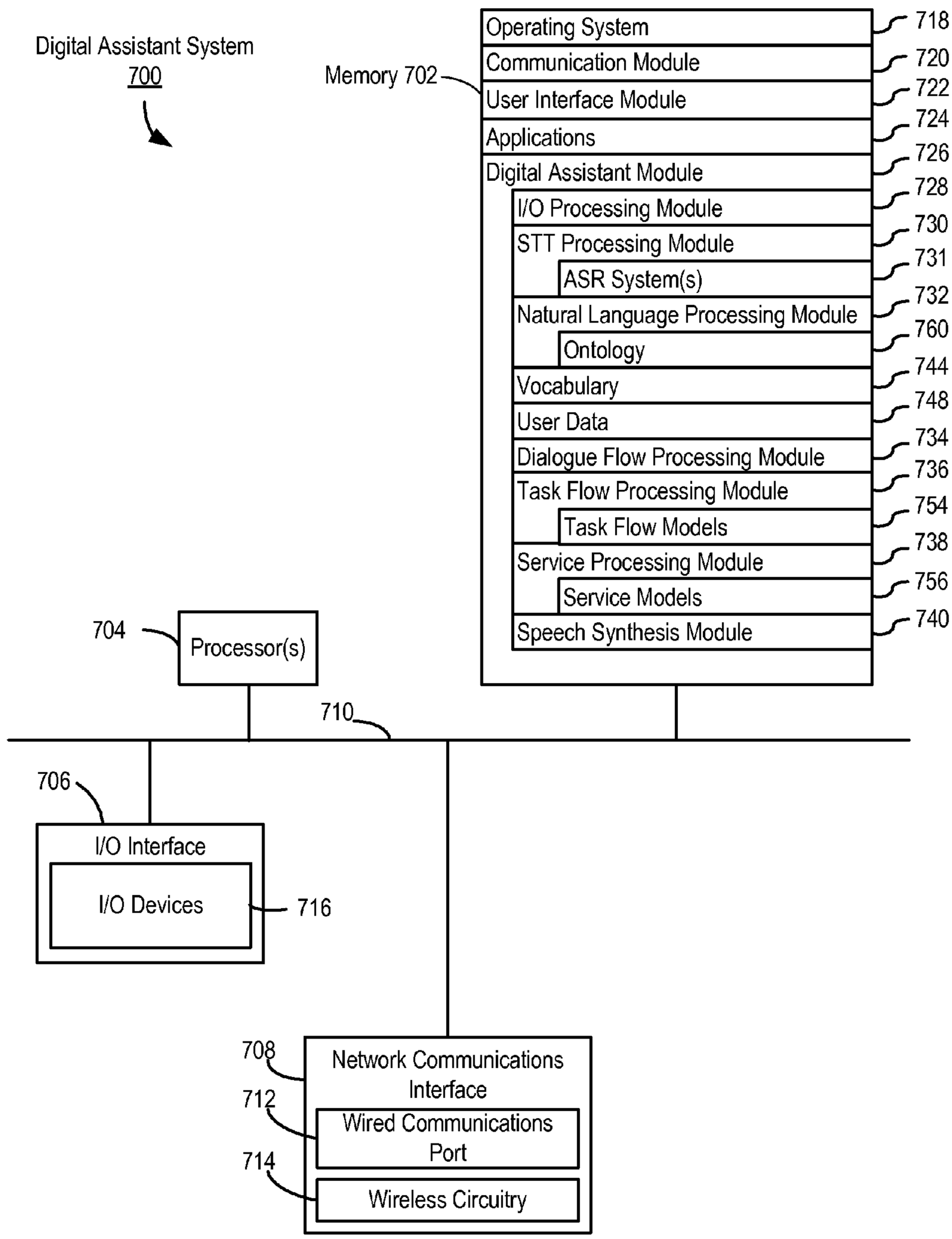

Digital Assistant System
700
Memory 702
Operating System
Communication Module
User Interface Module
Applications
Digital Assistant Module
I/O Processing Module
STT Processing Module
ASR System(s)
Natural Language Processing Module
Ontology
Vocabulary
User Data
Dialogue Flow Processing Module
Task Flow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

FIG. 7A

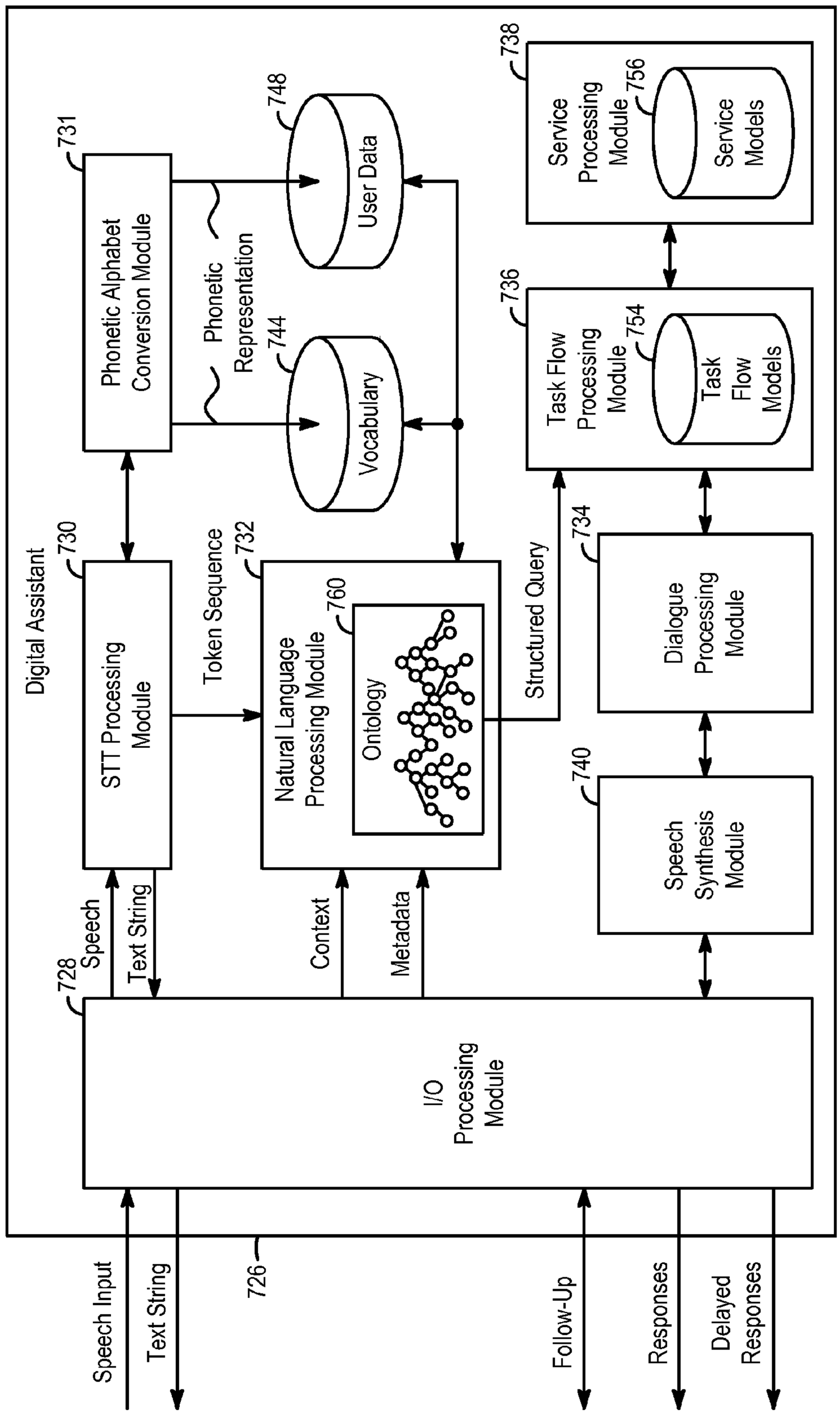

Digital Assistant
730
STT Processing Module
731
Phonetic Alphabet Conversion Module
Speech
Text String
728
I/O Processing Module
Speech Input
Text String
726
Follow-Up
Responses
Delayed Responses
Token Sequence
732
Natural Language Processing Module
760
Ontology
Context
Metadata
Phonetic Representation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

FIG. 7B

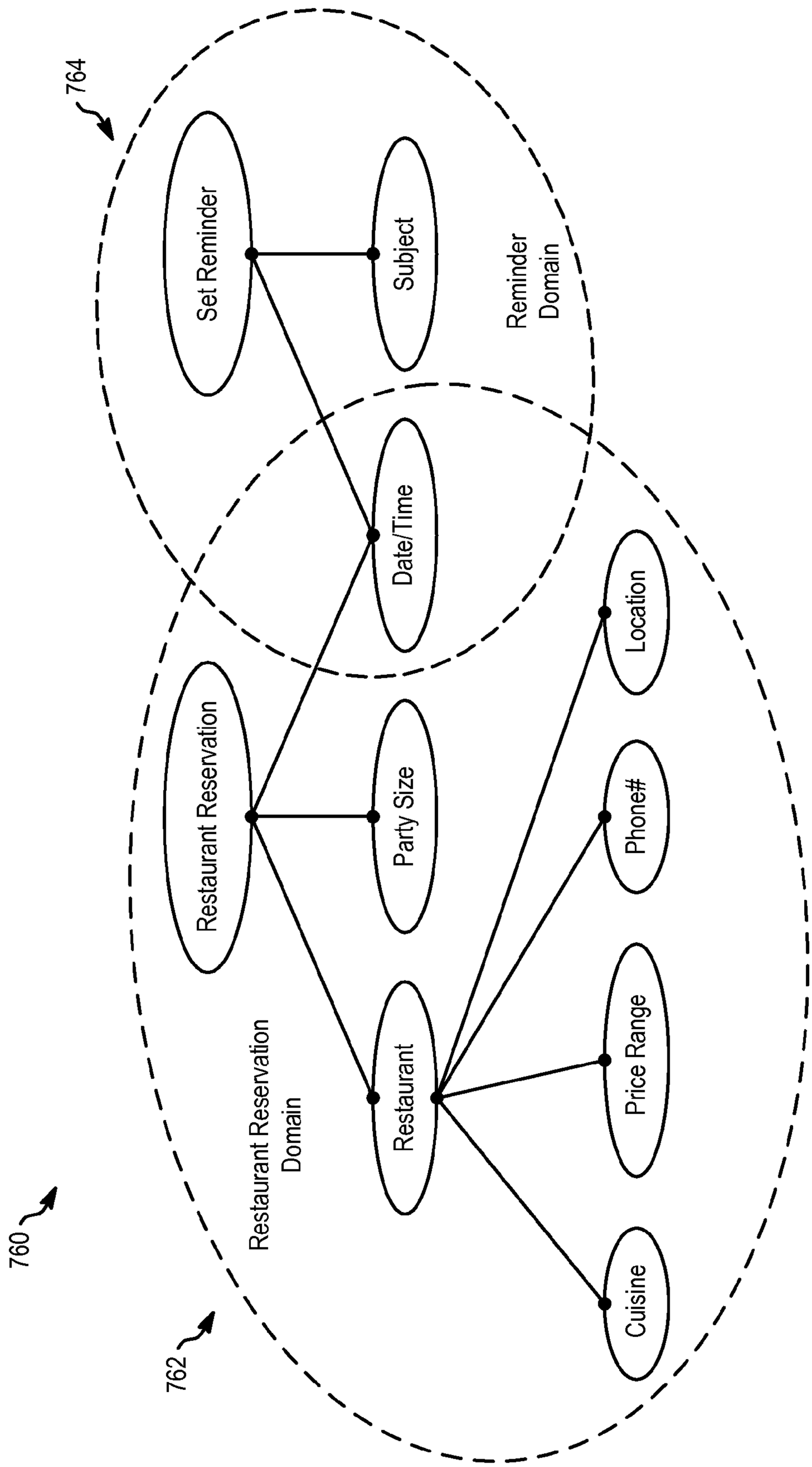

764
Set Reminder
Subject
Reminder Domain
Date/Time
Location
Restaurant Reservation
Party Size
Phone#
Restaurant
Price Range
Restaurant Reservation Domain
Cuisine
760
762

FIG. 7C

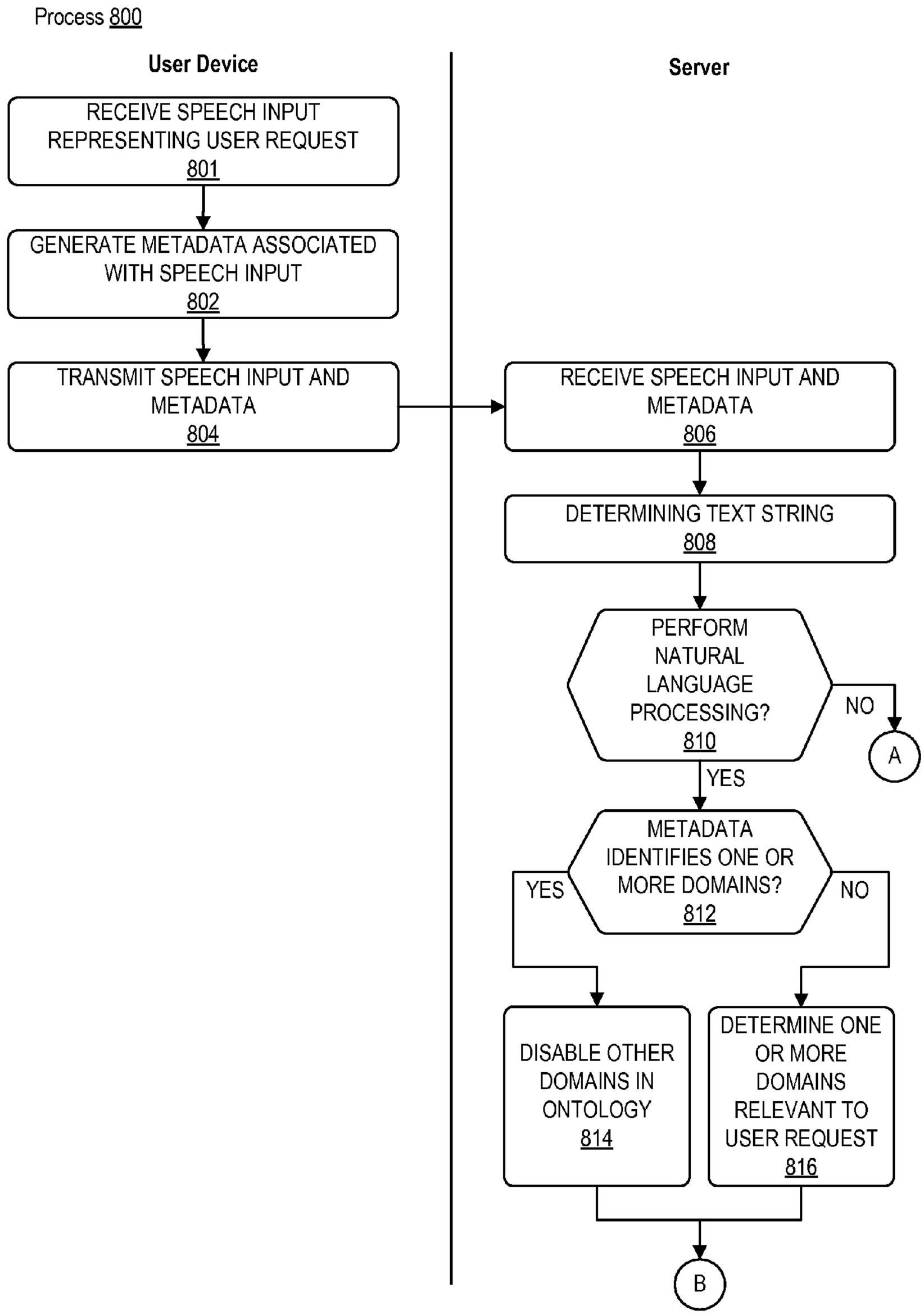

Process 800
User Device
Server
RECEIVE SPEECH INPUT REPRESENTING USER REQUEST 801
GENERATE METADATA ASSOCIATED WITH SPEECH INPUT 802
TRANSMIT SPEECH INPUT AND METADATA 804
RECEIVE SPEECH INPUT AND METADATA 806
DETERMINING TEXT STRING 808
PERFORM NATURAL LANGUAGE PROCESSING? 810
NO
A
YES
METADATA IDENTIFIES ONE OR MORE DOMAINS? 812
YES
NO
DISABLE OTHER DOMAINS IN ONTOLOGY 814
DETERMINE ONE OR MORE DOMAINS RELEVANT TO USER REQUEST 816
B

FIG. 8A

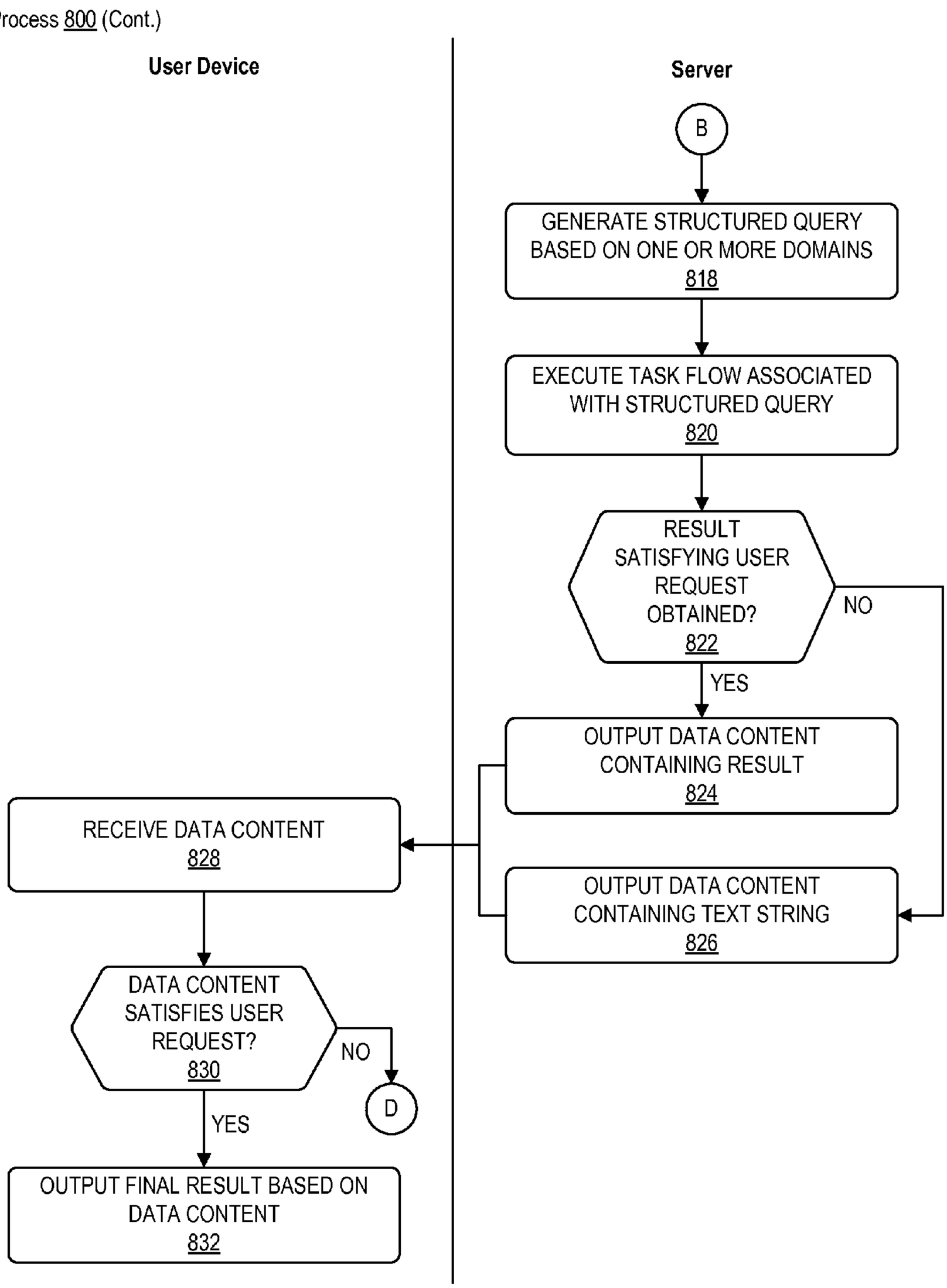

Process 800 (Cont.)
User Device
Server
B
GENERATE STRUCTURED QUERY BASED ON ONE OR MORE DOMAINS 818
EXECUTE TASK FLOW ASSOCIATED WITH STRUCTURED QUERY 820
RESULT SATISFYING USER REQUEST OBTAINED? 822
NO
YES
OUTPUT DATA CONTENT CONTAINING RESULT 824
OUTPUT DATA CONTENT CONTAINING TEXT STRING 826
RECEIVE DATA CONTENT 828
DATA CONTENT SATISFIES USER REQUEST? 830
NO
D
YES
OUTPUT FINAL RESULT BASED ON DATA CONTENT 832

FIG. 8B

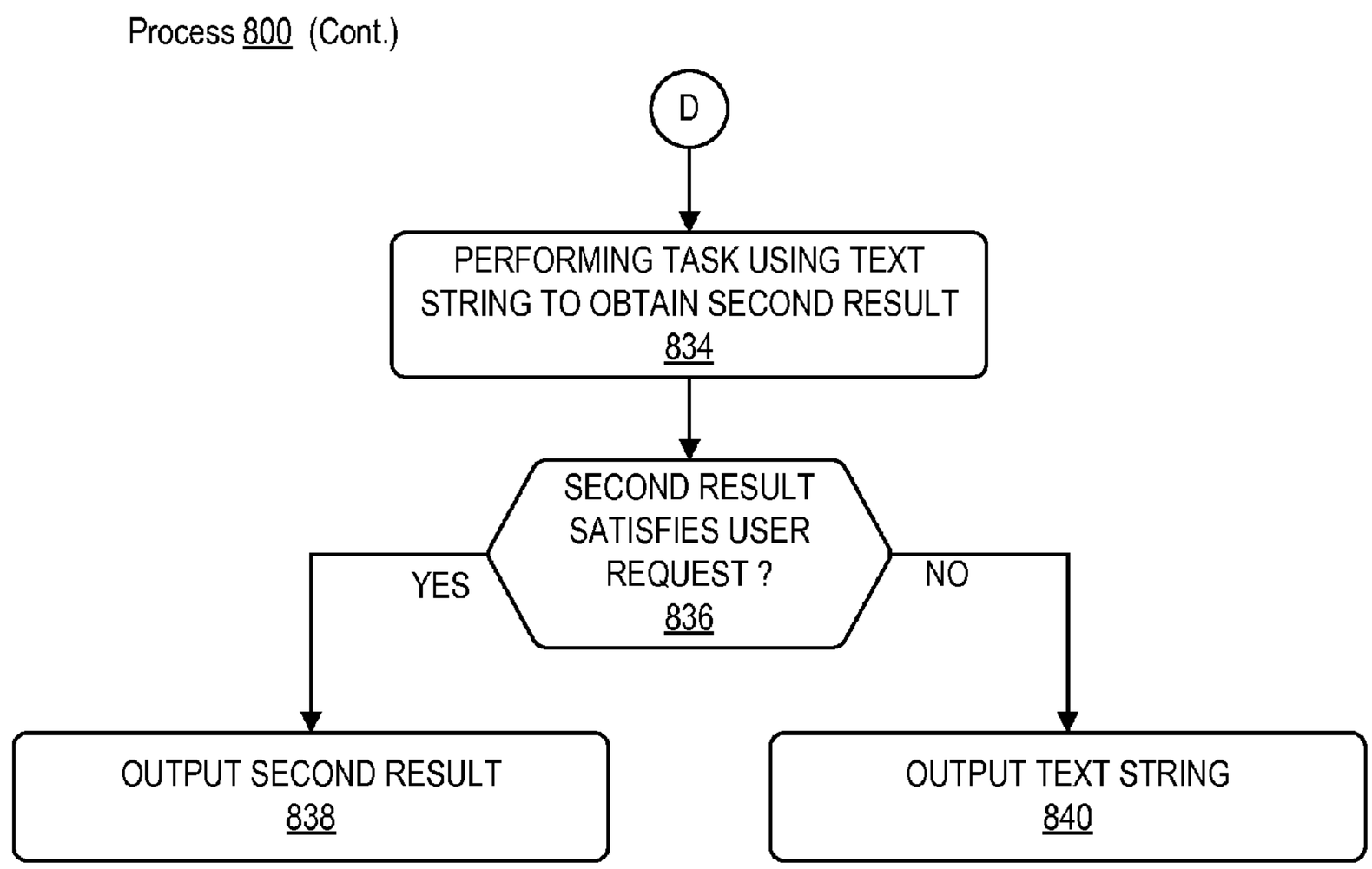

Process 800 (Cont.)
D
PERFORMING TASK USING TEXT STRING TO OBTAIN SECOND RESULT 834
SECOND RESULT SATISFIES USER REQUEST ? 836
YES
NO
OUTPUT SECOND RESULT 838
OUTPUT TEXT STRING 840

FIG. 8C

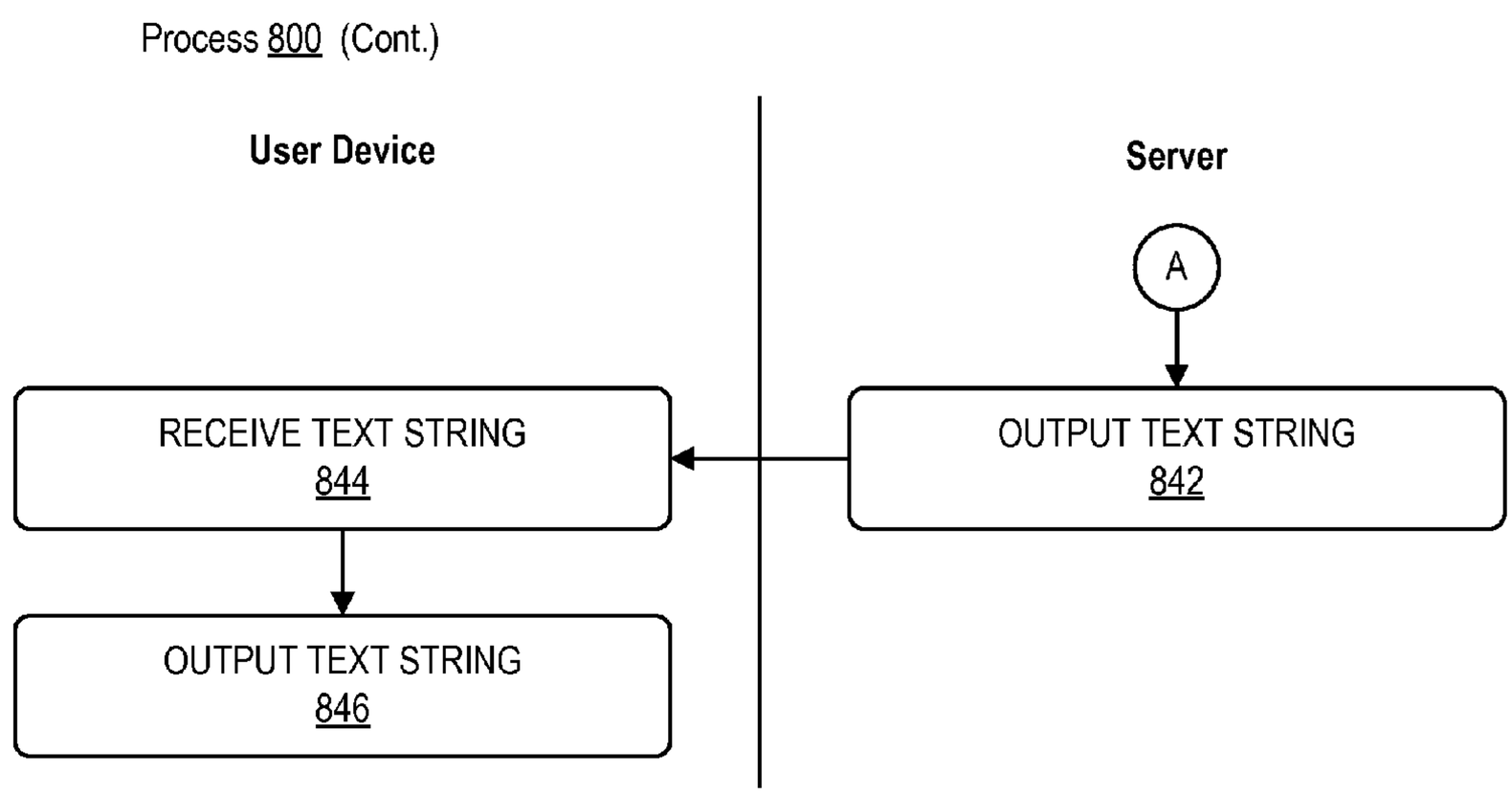

Process 800 (Cont.)
User Device
Server
A
RECEIVE TEXT STRING 844
OUTPUT TEXT STRING 842
OUTPUT TEXT STRING 846

FIG. 8D

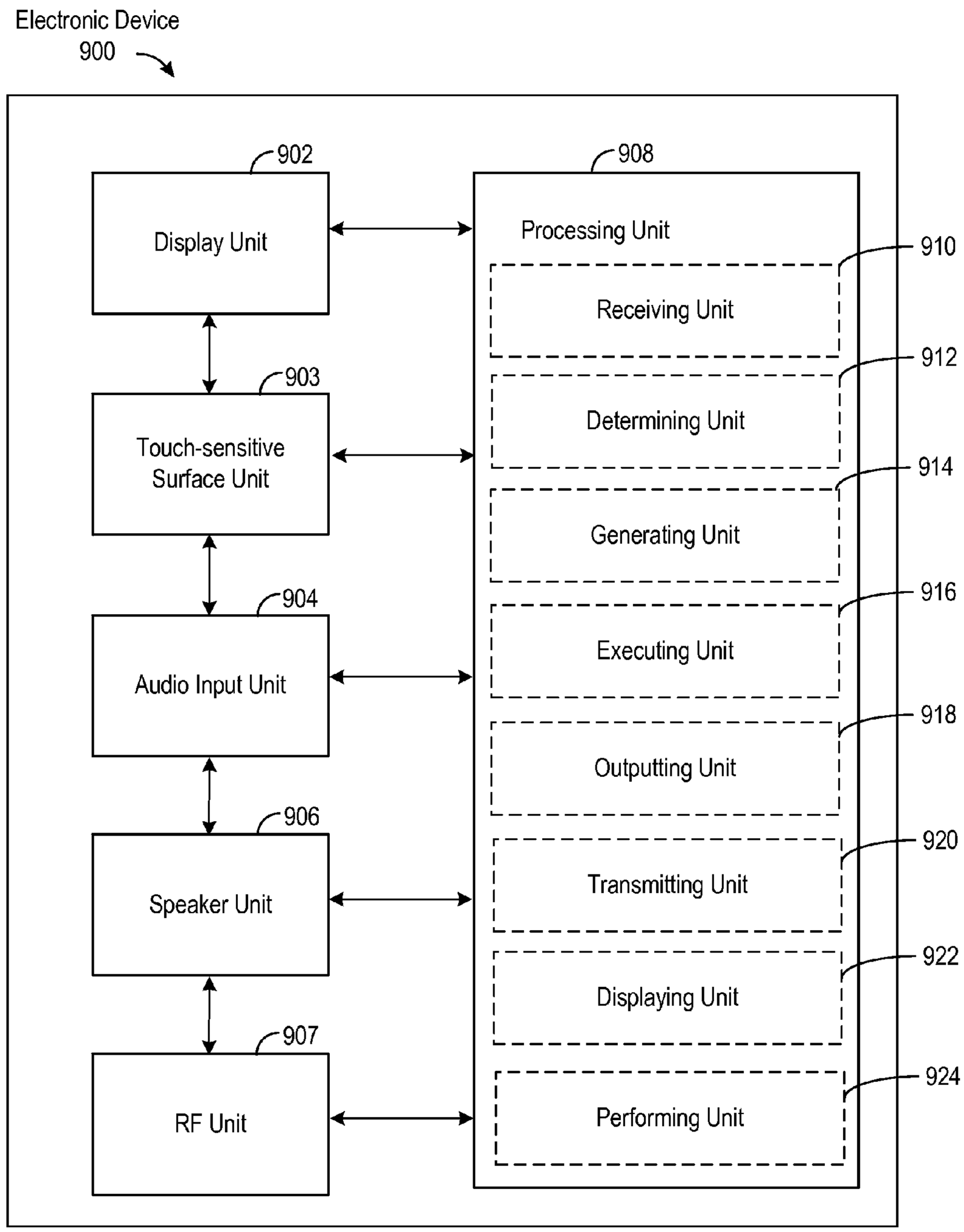

Electronic Device
900
902
Display Unit
903
Touch-sensitive Surface Unit
904
Audio Input Unit
906
Speaker Unit
907
RF Unit
908
Processing Unit
910
Receiving Unit
912
Determining Unit
914
Generating Unit
916
Executing Unit
918
Outputting Unit
920
Transmitting Unit
922
Displaying Unit
924
Performing Unit

FIG. 9

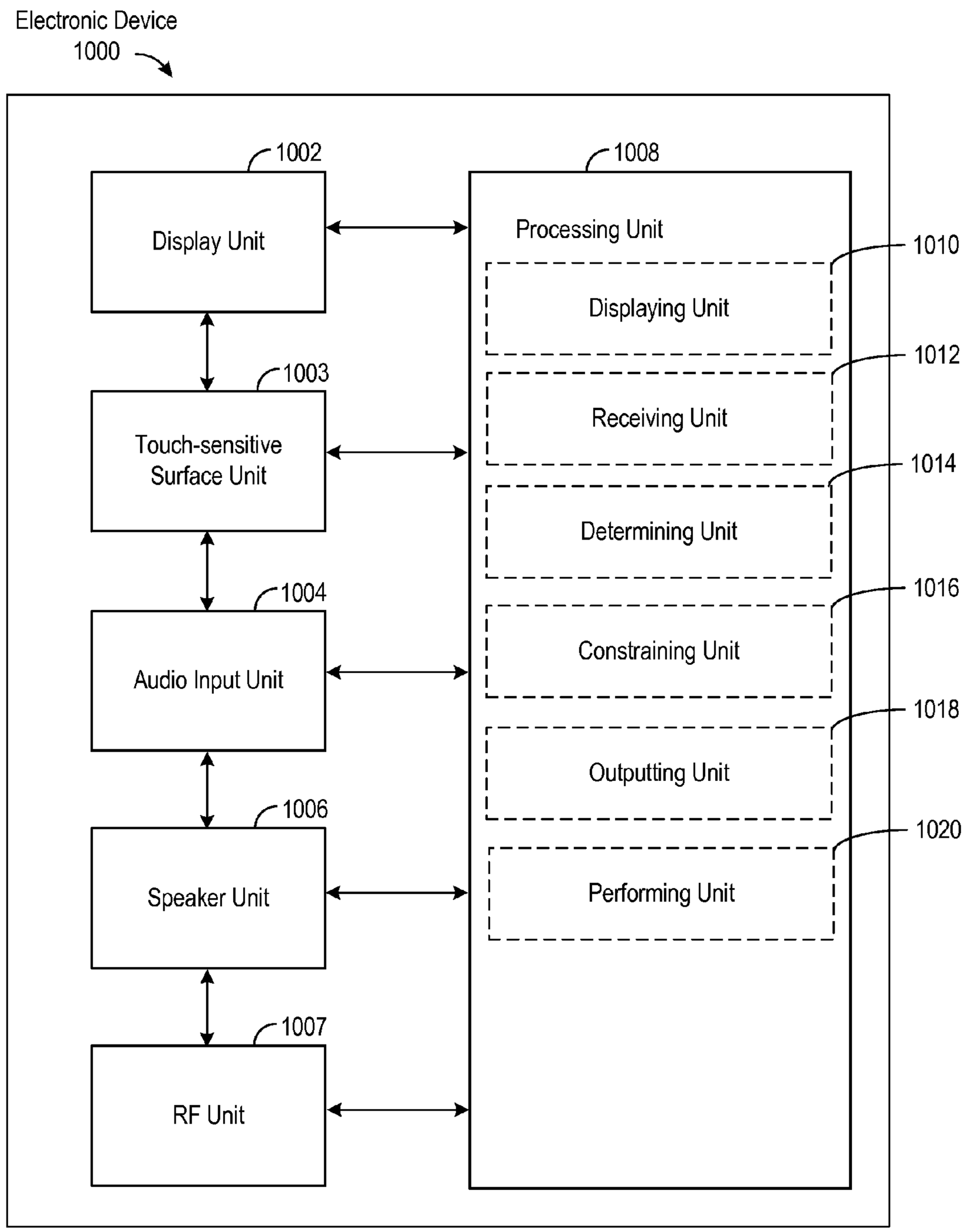

Electronic Device
1000
1002
Display Unit
1003
Touch-sensitive Surface Unit
1004
Audio Input Unit
1006
Speaker Unit
1007
RF Unit
1008
Processing Unit
1010
Displaying Unit
1012
Receiving Unit
1014
Determining Unit
1016
Constraining Unit
1018
Outputting Unit
1020
Performing Unit

FIG. 10

STRUCTURED DICTATION USING INTELLIGENT AUTOMATED ASSISTANTS

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims priority from U.S. Provisional Ser. No. 62/129,441, filed on Mar. 6, 2015, entitled STRUCTURED DICTATION USING INTELLIGENT AUTOMATED ASSISTANTS, which is hereby incorporated by reference in its entirety for all purposes.

FIELD

[0002] This relates generally to intelligent automated assistants and, more specifically, to structured dictation using intelligent automated assistants.

BACKGROUND

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

[0004] Digital assistants can interpret user intent by means of natural language processing. In particular, the user's speech input can be parsed to determine the semantic intent that is most likely implicated by the speech input. The process of identifying the most likely semantic intent can be computationally intensive. Further, in some cases, an incorrect semantic intent may be deduced due to the speech input implicating several possible semantic intents. This can adversely affect response latency of the digital assistant as well as the accuracy of the response obtained.

SUMMARY

[0005] Systems and processes for performing structured dictation using intelligent automated assistants are provided. In one example process, a speech input representing a user request can be received. In addition, metadata associated with the speech input can be received. A text string corresponding to the speech input can be determined. The process can determine whether to perform natural language processing on the text string and whether the metadata identifies one or more domains corresponding to the user request. In response to determining that natural language processing is to be performed on the text string and that the metadata identifies one or more domains corresponding to the user request, natural language processing of the text string can be constrained to the one or more domains. A result can be obtained based on the one or more domains and the result can be outputted.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] FIG. **1** is a block diagram illustrating a system and environment for implementing a digital assistant according to various examples.

[0007] FIG. **2**A is a block diagram illustrating a portable multifunction device implementing the client-side portion of a digital assistant according to various examples.

[0008] FIG. **2**B is a block diagram illustrating exemplary components for event handling according to various examples.

[0009] FIG. **3** illustrates a portable multifunction device implementing the client-side portion of a digital assistant according to various examples.

[0010] FIG. **4** is a block diagram of an exemplary multifunction device with a display and a touch-sensitive surface according to various examples.

[0011] FIG. **5**A illustrates an exemplary user interface for a menu of applications on a portable multifunction device according to various examples.

[0012] FIG. **5**B illustrates an exemplary user interface for a multifunction device with a touch-sensitive surface that is separate from the display according to various examples.

[0013] FIG. **6**A illustrates a personal electronic device according to various examples.

[0014] FIG. **6**B is a block diagram illustrating a personal electronic device according to various examples.

[0015] FIG. **7**A is a block diagram illustrating a digital assistant system or a server portion thereof according to various examples.

[0016] FIG. **7**B illustrates the functions of the digital assistant shown in FIG. **7**A according to various examples.

[0017] FIG. **7**C illustrates a portion of an ontology according to various examples.

[0018] FIGS. **8**A-D illustrate a process for operating a digital assistant according to various examples.

[0019] FIG. **9** illustrates a functional block diagram of an electronic device according to various examples.

[0020] FIG. **10** illustrates a functional block diagram of an electronic device according to various examples.

DETAILED DESCRIPTION

[0021] In the following description of examples, reference is made to the accompanying drawings in which it is shown by way of illustration specific examples that can be practiced. It is to be understood that other examples can be used and structural changes can be made without departing from the scope of the various examples.

[0022] As described above, digital assistants can implement natural language processing to determine the semantic intent of the user's speech input. In certain interactions with digital assistants, it can be desirable to leverage the context of the interaction to more accurately and efficiently identify the semantic intent. For example, a user can be composing an email message using the email application and can provide the speech input "mom" to the digital assistant while the recipient field of the email message is selected. In isolation, the semantic intent of the speech input "mom" can be difficult to deduce. However, in this example, it can be inferred, based on the selected recipient field and the context of the email application that the user is requesting the email address of the user's mother to be retrieved and then populated into the recipient field of the email message.

[0023] In another example, a user can be browsing a particular geographic region on the maps application and can provide the speech input "best restaurants" to the digital assistant while the search field of the maps application is selected. In this example, it can be inferred based on the geographic region being browsed and the context of the maps

application that the user is requesting the locations of restaurants having the highest ratings within that geographic region to be displayed on the map.

[0024] By utilizing the context of the interaction with the digital assistant to identify the semantic intent of the user's speech input, natural language processing of the speech input can be completed more quickly and accurately. This can enable lower response latency and a more accurate response from the digital assistant, which can improve user experience as well as improve battery usage in mobile applications.

[0025] In systems and processes for structured dictation using intelligent automated assistants described herein, the context of the interaction with the digital assistant is leveraged to enhance natural language processing. In one example process, an application comprising one or more text input fields can be displayed on an electronic device. A speech input can be received from a user via a microphone on the electronic device. A text string corresponding to the speech input can be determined. The process can determine whether the speech input is directed to the one or more text input fields of the application. In response to determining that the speech input is directed to the one or more text input fields of the application, natural language processing of the text string can be constrained to a domain of two or more domains and a result based on the domain can be outputted on the electronic device. In response to determining that the speech input is not directed to the one or more text input fields, natural language processing of the text string can be performed across the two or more domains and a result based on the two or more domains can be outputted on the electronic device.

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

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

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

1. System and Environment

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

[0030] Specifically, a digital assistant can be capable of accepting a user request at least partially in the form of a natural language command, request, statement, narrative, and/or inquiry. Typically, the user request can seek either an informational answer or performance of a task by the digital assistant. A satisfactory response to the user request can be a provision of the requested informational answer, a performance of the requested task, or a combination of the two. For example, a user can ask the digital assistant a question, such as "Where am I right now?" Based on the user's current location, the digital assistant can answer, "You are in Central Park near the west gate." The user can also request the performance of a task, for example, "Please invite my friends to my girlfriend's birthday party next week." In response, the digital assistant can acknowledge the request by saying "Yes, right away," and then send a suitable calendar invite on behalf of the user to each of the user's friends listed in the user's electronic address book. During performance of a requested task, the digital assistant can sometimes interact with the user in a continuous dialogue involving multiple exchanges of information over an extended period of time. There are numerous other ways of interacting with a digital assistant to request information or performance of various tasks. In addition to providing verbal responses and taking programmed actions, the digital assistant can also provide responses in other visual or audio forms, e.g., as text, alerts, music, videos, animations, etc.

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

[0032] In some examples, DA server **106** can include client-facing I/O interface **112**, one or more processing modules **114**, data and models **116**, and I/O interface to external services **118**. The client-facing I/O interface **112** can facilitate the client-facing input and output processing for DA server **106**. One or more processing modules **114** can utilize data and models **116** to process speech input and determine the user's intent based on natural language input. Further, one or more

processing modules **114** perform task execution based on inferred user intent. In some examples, DA server **106** can communicate with external services **120** through network(s) **110** for task completion or information acquisition. I/O interface to external services **118** can facilitate such communications.

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

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

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

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

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

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

2. Electronic Devices

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

[0040] As used in the specification and claims, the term "intensity" of a contact on a touch-sensitive surface refers to the force or pressure (force per unit area) of a contact (e.g., a finger contact) on the touch-sensitive surface, or to a substitute (proxy) for the force or pressure of a contact on the

touch-sensitive surface. The intensity of a contact has a range of values that includes at least four distinct values and more typically includes hundreds of distinct values (e.g., at least 256). Intensity of a contact is, optionally, determined (or measured) using various approaches and various sensors or combinations of sensors. For example, one or more force sensors underneath or adjacent to the touch-sensitive surface are, optionally, used to measure force at various points on the touch-sensitive surface. In some implementations, force measurements from multiple force sensors are combined (e.g., a weighted average) to determine an estimated force of a contact. Similarly, a pressure-sensitive tip of a stylus is, optionally, used to determine a pressure of the stylus on the touch-sensitive surface. Alternatively, the size of the contact area detected on the touch-sensitive surface and/or changes thereto, the capacitance of the touch-sensitive surface proximate to the contact and/or changes thereto, and/or the resistance of the touch-sensitive surface proximate to the contact and/or changes thereto are, optionally, used as a substitute for the force or pressure of the contact on the touch-sensitive surface. In some implementations, the substitute measurements for contact force or pressure are used directly to determine whether an intensity threshold has been exceeded (e.g., the intensity threshold is described in units corresponding to the substitute measurements). In some implementations, the substitute measurements for contact force or pressure are converted to an estimated force or pressure, and the estimated force or pressure is used to determine whether an intensity threshold has been exceeded (e.g., the intensity threshold is a pressure threshold measured in units of pressure). Using the intensity of a contact as an attribute of a user input allows for user access to additional device functionality that may otherwise not be accessible by the user on a reduced-size device with limited real estate for displaying affordances (e.g., on a touch-sensitive display) and/or receiving user input (e.g., via a touch-sensitive display, a touch-sensitive surface, or a physical/mechanical control such as a knob or a button).

[0041] As used in the specification and claims, the term "tactile output" refers to physical displacement of a device relative to a previous position of the device, physical displacement of a component (e.g., a touch-sensitive surface) of a device relative to another component (e.g., housing) of the device, or displacement of the component relative to a center of mass of the device that will be detected by a user with the user's sense of touch. For example, in situations where the device or the component of the device is in contact with a surface of a user that is sensitive to touch (e.g., a finger, palm, or other part of a user's hand), the tactile output generated by the physical displacement will be interpreted by the user as a tactile sensation corresponding to a perceived change in physical characteristics of the device or the component of the device. For example, movement of a touch-sensitive surface (e.g., a touch-sensitive display or trackpad) is, optionally, interpreted by the user as a "down click" or "up click" of a physical actuator button. In some cases, a user will feel a tactile sensation such as an "down click" or "up click" even when there is no movement of a physical actuator button associated with the touch-sensitive surface that is physically pressed (e.g., displaced) by the user's movements. As another example, movement of the touch-sensitive surface is, optionally, interpreted or sensed by the user as "roughness" of the touch-sensitive surface, even when there is no change in smoothness of the touch-sensitive surface. While such interpretations of touch by a user will be subject to the individualized sensory perceptions of the user, there are many sensory perceptions of touch that are common to a large majority of users. Thus, when a tactile output is described as corresponding to a particular sensory perception of a user (e.g., an "up click," a "down click," "roughness"), unless otherwise stated, the generated tactile output corresponds to physical displacement of the device or a component thereof that will generate the described sensory perception for a typical (or average) user.

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

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

[0044] In some examples, a non-transitory computer-readable storage medium of memory **202** can be used to store instructions (e.g., for performing aspects of process **800**, described below) for use by or in connection with an instruction execution system, apparatus, or device, such as a computer-based system, processor-containing system, or other system that can fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions. In other examples, the instructions (e.g., for performing aspects of process **800**, described below) can be stored on a non-transitory computer-readable storage medium (not shown) of the server system **108** or can be divided between the non-transitory computer-readable storage medium of memory **202** and the non-transitory computer-readable storage medium of server system **108**. In the context of this document, a "non-transitory computer-readable storage medium" can be any medium that can contain or store the program for use by or in connection with the instruction execution system, apparatus, or device.

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

[0046] RF (radio frequency) circuitry **208** receives and sends RF signals, also called electromagnetic signals. RF circuitry **208** converts electrical signals to/from electromagnetic signals and communicates with communications networks and other communications devices via the electromagnetic signals. RF circuitry **208** optionally includes well-known circuitry for performing these functions, including but not limited to an antenna system, an RF transceiver, one or more amplifiers, a tuner, one or more oscillators, a digital

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

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

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

[0049] A quick press of the push button may disengage a lock of touch screen **212** or begin a process that uses gestures on the touch screen to unlock the device, as described in U.S. patent application Ser. No. 11/322,549, "Unlocking a Device by Performing Gestures on an Unlock Image," filed Dec. 23, 2005, U.S. Pat. No. 7,657,849, which is hereby incorporated by reference in its entirety. A longer press of the push button (e.g., **306**) may turn power to device **200** on or off. The user may be able to customize a functionality of one or more of the buttons. Touch screen **212** is used to implement virtual or soft buttons and one or more soft keyboards.

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

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

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

[0053] A touch-sensitive display in some embodiments of touch screen **212** may be analogous to the multi-touch sensitive touchpads described in the following U.S. Pat. No. 6,323,846 (Westerman et al.), U.S. Pat. No. 6,570,557 (Westerman et al.), and/or U.S. Pat. No. 6,677,932 (Westerman), and/or U.S. Patent Publication 2002/0015024A1, each of which is hereby incorporated by reference in its entirety. However, touch screen **212** displays visual output from device **200**, whereas touch-sensitive touchpads do not provide visual output.

[0054] A touch-sensitive display in some embodiments of touch screen **212** may be as described in the following applications: (1) U.S. patent application Ser. No. 11/381,313, "Multipoint Touch Surface Controller," filed May 2, 2006; (2)

U.S. patent application Ser. No. 10/840,862, "Multipoint Touchscreen," filed May 6, 2004; (3) U.S. patent application Ser. No. 10/903,964, "Gestures For Touch Sensitive Input Devices," filed Jul. 30, 2004; (4) U.S. patent application Ser. No. 11/048,264, "Gestures For Touch Sensitive Input Devices," filed Jan. 31, 2005; (5) U.S. patent application Ser. No. 11/038,590, "Mode-Based Graphical User Interfaces For Touch Sensitive Input Devices," filed Jan. 18, 2005; (6) U.S. patent application Ser. No. 11/228,758, "Virtual Input Device Placement On A Touch Screen User Interface," filed Sep. 16, 2005; (7) U.S. patent application Ser. No. 11/228,700, "Operation Of A Computer With A Touch Screen Interface," filed Sep. 16, 2005; (8) U.S. patent application Ser. No. 11/228,737, "Activating Virtual Keys Of A Touch-Screen Virtual Keyboard," filed Sep. 16, 2005; and (9) U.S. patent application Ser. No. 11/367,749, "Multi-Functional Hand-Held Device," filed Mar. 3, 2006. All of these applications are incorporated by reference herein in their entirety.

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

[0056] In some embodiments, in addition to the touch screen, device **200** may include a touchpad (not shown) for activating or deactivating particular functions. In some embodiments, the touchpad is a touch-sensitive area of the device that, unlike the touch screen, does not display visual output. The touchpad may be a touch-sensitive surface that is separate from touch screen **212** or an extension of the touch-sensitive surface formed by the touch screen.

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

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

[0059] Device **200** optionally also includes one or more contact intensity sensors **265**. FIG. **2**A shows a contact intensity sensor coupled to intensity sensor controller **259** in I/O subsystem **206**. Contact intensity sensor **265** optionally includes one or more piezoresistive strain gauges, capacitive force sensors, electric force sensors, piezoelectric force sensors, optical force sensors, capacitive touch-sensitive surfaces, or other intensity sensors (e.g., sensors used to measure the force (or pressure) of a contact on a touch-sensitive surface). Contact intensity sensor **265** receives contact intensity information (e.g., pressure information or a proxy for pressure information) from the environment. In some embodiments, at least one contact intensity sensor is collocated with, or proximate to, a touch-sensitive surface (e.g., touch-sensitive display system **212**). In some embodiments, at least one contact intensity sensor is located on the back of device **200**, opposite touch screen display **212**, which is located on the front of device **200**.

[0060] Device **200** may also include one or more proximity sensors **266**. FIG. **2**A shows proximity sensor **266** coupled to peripherals interface **218**. Alternately, proximity sensor **266** may be coupled to input controller **260** in I/O subsystem **206**. Proximity sensor **266** may perform as described in U.S. patent application Ser. No. 11/241,839, "Proximity Detector In Handheld Device"; U.S. patent application Ser. No. 11/240,788, "Proximity Detector In Handheld Device"; U.S. patent application Ser, No. 11/620,702, "Using Ambient Light Sensor To Augment Proximity Sensor Output";U.S. patent application Ser. No. 11/586,862, "Automated Response To And Sensing Of User Activity In Portable Devices"; and Ser. U.S. patent application Ser. No. 11/638,251, "Methods And Systems For Automatic Configuration Of Peripherals," which are hereby incorporated by reference in their entirety. In some embodiments, the proximity sensor turns off and disables touch screen **212** when the multifunction device is placed near the user's ear (e.g., when the user is making a phone call).

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

erator sensor is located on the back of device **200**, opposite touch screen display **212**, which is located on the front of device **200**.

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

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

[0064] Operating system **226** (e.g., Darwin, RTXC, LINUX, UNIX, OS X, iOS, WINDOWS, or an embedded operating system such as VxWorks) includes various software components and/or drivers for controlling and managing general system tasks (e.g., memory management, storage device control, power management, etc.) and facilitates communication between various hardware and software components.

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

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

[0067] In some embodiments, contact/motion module **230** uses a set of one or more intensity thresholds to determine whether an operation has been performed by a user (e.g., to determine whether a user has "clicked" on an icon). In some embodiments, at least a subset of the intensity thresholds are determined in accordance with software parameters (e.g., the intensity thresholds are not determined by the activation thresholds of particular physical actuators and can be adjusted without changing the physical hardware of device **200**). For example, a mouse "click" threshold of a trackpad or touch screen display can be set to any of a large range of predefined threshold values without changing the trackpad or touch screen display hardware. Additionally, in some implementations, a user of the device is provided with software settings for adjusting one or more of the set of intensity thresholds (e.g., by adjusting individual intensity thresholds and/or by adjusting a plurality of intensity thresholds at once with a system-level click "intensity" parameter).

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

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

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

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

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

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

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

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

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

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

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

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

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

- **[0081]** Contacts module **237** (sometimes called an address book or contact list);
- **[0082]** Telephone module **238**;
- **[0083]** Video conference module **239**;
- **[0084]** E-mail client module **240**;
- **[0085]** Instant messaging (IM) module **241**;
- **[0086]** Workout support module **242**;
- **[0087]** Camera module **243** for still and/or video images;
- **[0088]** Image management module **244**;
- **[0089]** Video player module;
- **[0090]** Music player module;
- **[0091]** Browser module **247**;
- **[0092]** Calendar module **248**;
- **[0093]** Widget modules **249**, which may include one or more of: weather widget **249-1**, stocks widget **249-2**, calculator widget **249-3**, alarm clock widget **249-4**, dictionary widget **249-5**, and other widgets obtained by the user, as well as user-created widgets **249-6**;
- **[0094]** Widget creator module **250** for making user-created widgets **249-6**;
- **[0095]** Search module **251**;
- **[0096]** Video and music player module **252**, which merges video player module and music player module;
- **[0097]** Notes module **253**;
- **[0098]** Map module **254**; and/or
- **[0099]** Online video module **255**.

[0100] Examples of other applications **236** that may be stored in memory **202** include other word processing applications, other image editing applications, drawing applica-

tions, presentation applications, JAVA-enabled applications, encryption, digital rights management, voice recognition, and voice replication.

[0101] In conjunction with touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, and text input module **234**, contacts module **237** may be used to manage an address book or contact list (e.g., stored in application internal state **292** of contacts module **237** in memory **202** or memory **470**), including: adding name(s) to the address book; deleting name(s) from the address book; associating telephone number(s), e-mail address(es), physical address(es) or other information with a name; associating an image with a name; categorizing and sorting names; providing telephone numbers or e-mail addresses to initiate and/or facilitate communications by telephone **238**, video conference module **239**, e-mail **240**, or IM **241**; and so forth.

[0102] In conjunction with RF circuitry **208**, audio circuitry **210**, speaker **211**, microphone **213**, touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, and text input module **234**, telephone module **238** may be used to enter a sequence of characters corresponding to a telephone number, access one or more telephone numbers in contacts module **237**, modify a telephone number that has been entered, dial a respective telephone number, conduct a conversation, and disconnect or hang up when the conversation is completed. As noted above, the wireless communication may use any of a plurality of communications standards, protocols, and technologies.

[0103] In conjunction with RF circuitry **208**, audio circuitry **210**, speaker **211**, microphone **213**, touch screen **212**, display controller **256**, optical sensor **264**, optical sensor controller **258**, contact/motion module **230**, graphics module **232**, text input module **234**, contacts module **237**, and telephone module **238**, video conference module **239** includes executable instructions to initiate, conduct, and terminate a video conference between a user and one or more other participants in accordance with user instructions.

[0104] In conjunction with RF circuitry **208**, touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, and text input module **234**, e-mail client module **240** includes executable instructions to create, send, receive, and manage e-mail in response to user instructions. In conjunction with image management module **244**, e-mail client module **240** makes it very easy to create and send e-mails with still or video images taken with camera module **243**.

[0105] In conjunction with RF circuitry **208**, touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, and text input module **234**, the instant messaging module **241** includes executable instructions to enter a sequence of characters corresponding to an instant message, to modify previously entered characters, to transmit a respective instant message (for example, using a Short Message Service (SMS) or Multimedia Message Service (MMS) protocol for telephony-based instant messages or using XMPP, SIMPLE, or IMPS for Internet-based instant messages), to receive instant messages, and to view received instant messages. In some embodiments, transmitted and/or received instant messages may include graphics, photos, audio files, video files and/or other attachments as are supported in an MMS and/or an Enhanced Messaging Service (EMS). As used herein, "instant messaging" refers to both telephony-based messages (e.g., messages sent using SMS or MMS) and Internet-based messages (e.g., messages sent using XMPP, SIMPLE, or IMPS).

[0106] In conjunction with RF circuitry **208**, touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, text input module **234**, GPS module **235**, map module **254**, and music player module, workout support module **242** includes executable instructions to create workouts (e.g., with time, distance, and/or calorie burning goals); communicate with workout sensors (sports devices); receive workout sensor data; calibrate sensors used to monitor a workout; select and play music for a workout; and display, store, and transmit workout data.

[0107] In conjunction with touch screen **212**, display controller **256**, optical sensor(s) **264**, optical sensor controller **258**, contact/motion module **230**, graphics module **232**, and image management module **244**, camera module **243** includes executable instructions to capture still images or video (including a video stream) and store them into memory **202**, modify characteristics of a still image or video, or delete a still image or video from memory **202**.

[0108] In conjunction with touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, text input module **234**, and camera module **243**, image management module **244** includes executable instructions to arrange, modify (e.g., edit), or otherwise manipulate, label, delete, present (e.g., in a digital slide show or album), and store still and/or video images.

[0109] In conjunction with RF circuitry **208**, touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, and text input module **234**, browser module **247** includes executable instructions to browse the Internet in accordance with user instructions, including searching, linking to, receiving, and displaying web pages or portions thereof, as well as attachments and other files linked to web pages.

[0110] In conjunction with RF circuitry **208**, touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, text input module **234**, e-mail client module **240**, and browser module **247**, calendar module **248** includes executable instructions to create, display, modify, and store calendars and data associated with calendars (e.g., calendar entries, to-do lists, etc.) in accordance with user instructions.

[0111] In conjunction with RF circuitry **208**, touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, text input module **234**, and browser module **247**, widget modules **249** are mini-applications that may be downloaded and used by a user (e.g., weather widget **249-1**, stocks widget **249-2**, calculator widget **249-3**, alarm clock widget **249-4**, and dictionary widget **249-5**) or created by the user (e.g., user-created widget **249-6**). In some embodiments, a widget includes an HTML (Hypertext Markup Language) file, a CSS (Cascading Style Sheets) file, and a JavaScript file. In some embodiments, a widget includes an XML (Extensible Markup Language) file and a JavaScript file (e.g., Yahoo! Widgets).

[0112] In conjunction with RF circuitry **208**, touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, text input module **234**, and browser module **247**, the widget creator module **250** may be used by a user to create widgets (e.g., turning a user-specified portion of a web page into a widget).

[0113] In conjunction with touch screen **212**, display controller **256**, contact/motion module **230**, graphics module

232, and text input module **234**, search module **251** includes executable instructions to search for text, music, sound, image, video, and/or other files in memory **202** that match one or more search criteria (e.g., one or more user-specified search terms) in accordance with user instructions.

[0114] In conjunction with touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, audio circuitry **210**, speaker **211**, RF circuitry **208**, and browser module **247**, video and music player module **252** includes executable instructions that allow the user to download and play back recorded music and other sound files stored in one or more file formats, such as MP3 or AAC files, and executable instructions to display, present, or otherwise play back videos (e.g., on touch screen **212** or on an external, connected display via external port **224**). In some embodiments, device **200** optionally includes the functionality of an MP3 player, such as an iPod (trademark of Apple Inc.).

[0115] In conjunction with touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, and text input module **234**, notes module **253** includes executable instructions to create and manage notes, to-do lists, and the like in accordance with user instructions.

[0116] In conjunction with RF circuitry **208**, touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, text input module **234**, GPS module **235**, and browser module **247**, map module **254** may be used to receive, display, modify, and store maps and data associated with maps (e.g., driving directions, data on stores and other points of interest at or near a particular location, and other location-based data) in accordance with user instructions.

[0117] In conjunction with touch screen **212**, display controller **256**, contact/motion module **230**, graphics module **232**, audio circuitry **210**, speaker **211**, RF circuitry **208**, text input module **234**, e-mail client module **240**, and browser module **247**, online video module **255** includes instructions that allow the user to access, browse, receive (e.g., by streaming and/or download), play back (e.g., on the touch screen or on an external, connected display via external port **224**), send an e-mail with a link to a particular online video, and otherwise manage online videos in one or more file formats, such as H.264. In some embodiments, instant messaging module **241**, rather than e-mail client module **240**, is used to send a link to a particular online video. Additional description of the online video application can be found in U.S. Provisional Patent Application No. 60/936,562, "Portable Multifunction Device, Method, and Graphical User Interface for Playing Online Videos," filed Jun. 20, 2007, and U.S. patent application Ser. No. 11/968,067, "Portable Multifunction Device, Method, and Graphical User Interface for Playing Online Videos," filed Dec. 31, 2007, the contents of which are hereby incorporated by reference in their entirety.

[0118] Each of the above-identified modules and applications corresponds to a set of executable instructions for performing one or more functions described above and the methods described in this application (e.g., the computer-implemented methods and other information processing methods described herein). These modules (e.g., sets of instructions) need not be implemented as separate software programs, procedures, or modules, and thus various subsets of these modules may be combined or otherwise rearranged in various embodiments. For example, video player module may be combined with music player module into a single module (e.g., video and music player module **252**, FIG. **2**A). In some embodiments, memory **202** may store a subset of the modules and data structures identified above. Furthermore, memory **202** may store additional modules and data structures not described above.

[0119] In some embodiments, device **200** is a device where operation of a predefined set of functions on the device is performed exclusively through a touch screen and/or a touchpad. By using a touch screen and/or a touchpad as the primary input control device for operation of device **200**, the number of physical input control devices (such as push buttons, dials, and the like) on device **200** may be reduced.

[0120] The predefined set of functions that are performed exclusively through a touch screen and/or a touchpad optionally include navigation between user interfaces. In some embodiments, the touchpad, when touched by the user, navigates device **200** to a main, home, or root menu from any user interface that is displayed on device **200**. In such embodiments, a "menu button" is implemented using a touchpad. In some other embodiments, the menu button is a physical push button or other physical input control device instead of a touchpad.

[0121] FIG. **2**B is a block diagram illustrating exemplary components for event handling in accordance with some embodiments. In some embodiments, memory **202** (FIG. **2**A) or **470** (FIG. **4**) includes event sorter **270** (e.g., in operating system **226**) and a respective application **236-1** (e.g., any of the aforementioned applications **237-251**, **255**, **480-490**).

[0122] Event sorter **270** receives event information and determines the application **236-1** and application view **291** of application **236-1** to which to deliver the event information. Event sorter **270** includes event monitor **271** and event dispatcher module **274**. In some embodiments, application **236-1** includes application internal state **292**, which indicates the current application view(s) displayed on touch-sensitive display **212** when the application is active or executing. In some embodiments, device/global internal state **257** is used by event sorter **270** to determine which application(s) is (are) currently active, and application internal state **292** is used by event sorter **270** to determine application views **291** to which to deliver event information.

[0123] In some embodiments, application internal state **292** includes additional information, such as one or more of: resume information to be used when application **236-1** resumes execution, user interface state information that indicates information being displayed or that is ready for display by application **236-1**, a state queue for enabling the user to go back to a prior state or view of application **236-1**, and a redo/undo queue of previous actions taken by the user.

[0124] Event monitor **271** receives event information from peripherals interface **218**. Event information includes information about a sub-event (e.g., a user touch on touch-sensitive display **212**, as part of a multi-touch gesture). Peripherals interface **218** transmits information it receives from I/O subsystem **206** or a sensor, such as proximity sensor **266**, accelerometer(s) **268**, and/or microphone **213** (through audio circuitry **210**). Information that peripherals interface **218** receives from I/O subsystem **206** includes information from touch-sensitive display **212** or a touch-sensitive surface.

[0125] In some embodiments, event monitor **271** sends requests to the peripherals interface **218** at predetermined intervals. In response, peripherals interface **218** transmits event information. In other embodiments, peripherals interface **218** transmits event information only when there is a

significant event (e.g., receiving an input above a predetermined noise threshold and/or for more than a predetermined duration).

[0126] In some embodiments, event sorter **270** also includes a hit view determination module **272** and/or an active event recognizer determination module **273**.

[0127] Hit view determination module **272** provides software procedures for determining where a sub-event has taken place within one or more views when touch-sensitive display **212** displays more than one view. Views are made up of controls and other elements that a user can see on the display.

[0128] Another aspect of the user interface associated with an application is a set of views, sometimes herein called application views or user interface windows, in which information is displayed and touch-based gestures occur. The application views (of a respective application) in which a touch is detected may correspond to programmatic levels within a programmatic or view hierarchy of the application. For example, the lowest level view in which a touch is detected may be called the hit view, and the set of events that are recognized as proper inputs may be determined based, at least in part, on the hit view of the initial touch that begins a touch-based gesture.

[0129] Hit view determination module **272** receives information related to sub events of a touch-based gesture. When an application has multiple views organized in a hierarchy, hit view determination module **272** identifies a hit view as the lowest view in the hierarchy which should handle the sub-event. In most circumstances, the hit view is the lowest level view in which an initiating sub-event occurs (e.g., the first sub-event in the sequence of sub-events that form an event or potential event). Once the hit view is identified by the hit view determination module **272**, the hit view typically receives all sub-events related to the same touch or input source for which it was identified as the hit view.

[0130] Active event recognizer determination module **273** determines which view or views within a view hierarchy should receive a particular sequence of sub-events. In some embodiments, active event recognizer determination module **273** determines that only the hit view should receive a particular sequence of sub-events. In other embodiments, active event recognizer determination module **273** determines that all views that include the physical location of a sub-event are actively involved views, and therefore determines that all actively involved views should receive a particular sequence of sub-events. In other embodiments, even if touch sub-events were entirely confined to the area associated with one particular view, views higher in the hierarchy would still remain as actively involved views.

[0131] Event dispatcher module **274** dispatches the event information to an event recognizer (e.g., event recognizer **280**). In embodiments including active event recognizer determination module **273**, event dispatcher module **274** delivers the event information to an event recognizer determined by active event recognizer determination module **273**. In some embodiments, event dispatcher module **274** stores in an event queue the event information, which is retrieved by a respective event receiver **282**.

[0132] In some embodiments, operating system **226** includes event sorter **270**. Alternatively, application **236-1** includes event sorter **270**. In yet other embodiments, event sorter **270** is a stand-alone module, or a part of another module stored in memory **202**, such as contact/motion module **230**.

[0133] In some embodiments, application **236-1** includes a plurality of event handlers **290** and one or more application views **291**, each of which includes instructions for handling touch events that occur within a respective view of the application's user interface. Each application view **291** of the application **236-1** includes one or more event recognizers **280**. Typically, a respective application view **291** includes a plurality of event recognizers **280**. In other embodiments, one or more of event recognizers **280** are part of a separate module, such as a user interface kit (not shown) or a higher level object from which application **236-1** inherits methods and other properties. In some embodiments, a respective event handler **290** includes one or more of: data updater **276**, object updater **277**, GUI updater **278**, and/or event data **279** received from event sorter **270**. Event handler **290** may utilize or call data updater **276**, object updater **277**, or GUI updater **278** to update the application internal state **292**. Alternatively, one or more of the application views **291** include one or more respective event handlers **290**. Also, in some embodiments, one or more of data updater **276**, object updater **277**, and GUI updater **278** are included in a respective application view **291**.

[0134] A respective event recognizer **280** receives event information (e.g., event data **279**) from event sorter **270** and identifies an event from the event information. Event recognizer **280** includes event receiver **282** and event comparator **284**. In some embodiments, event recognizer **280** also includes at least a subset of: metadata **283**, and event delivery instructions **288** (which may include sub-event delivery instructions).

[0135] Event receiver **282** receives event information from event sorter **270**. The event information includes information about a sub-event, for example, a touch or a touch movement. Depending on the sub-event, the event information also includes additional information, such as location of the sub-event. When the sub-event concerns motion of a touch, the event information may also include speed and direction of the sub-event. In some embodiments, events include rotation of the device from one orientation to another (e.g., from a portrait orientation to a landscape orientation, or vice versa), and the event information includes corresponding information about the current orientation (also called device attitude) of the device.

[0136] Event comparator **284** compares the event information to predefined event or sub-event definitions and, based on the comparison, determines an event or sub event, or determines or updates the state of an event or sub-event. In some embodiments, event comparator **284** includes event definitions **286**. Event definitions **286** contain definitions of events (e.g., predefined sequences of sub-events), for example, event **1** (**287-1**), event **2** (**287-2**), and others. In some embodiments, sub-events in an event (**287**) include, for example, touch begin, touch end, touch movement, touch cancellation, and multiple touching. In one example, the definition for event **1** (**287-1**) is a double tap on a displayed object. The double tap, for example, comprises a first touch (touch begin) on the displayed object for a predetermined phase, a first liftoff (touch end) for a predetermined phase, a second touch (touch begin) on the displayed object for a predetermined phase, and a second liftoff (touch end) for a predetermined phase. In another example, the definition for event **2** (**287-2**) is a dragging on a displayed object. The dragging, for example, comprises a touch (or contact) on the displayed object for a predetermined phase, a movement of the touch across touch-sensitive display **212**, and liftoff of the touch (touch end). In

some embodiments, the event also includes information for one or more associated event handlers **290**.

[0137] In some embodiments, event definition **287** includes a definition of an event for a respective user-interface object. In some embodiments, event comparator **284** performs a hit test to determine which user-interface object is associated with a sub-event. For example, in an application view in which three user-interface objects are displayed on touch-sensitive display **212**, when a touch is detected on touch-sensitive display **212**, event comparator **284** performs a hit test to determine which of the three user-interface objects is associated with the touch (sub-event). If each displayed object is associated with a respective event handler **290**, the event comparator uses the result of the hit test to determine which event handler **290** should be activated. For example, event comparator **284** selects an event handler associated with the sub-event and the object triggering the hit test.

[0138] In some embodiments, the definition for a respective event (**287**) also includes delayed actions that delay delivery of the event information until after it has been determined whether the sequence of sub-events does or does not correspond to the event recognizer's event type.

[0139] When a respective event recognizer **280** determines that the series of sub-events do not match any of the events in event definitions **286**, the respective event recognizer **280** enters an event impossible, event failed, or event ended state, after which it disregards subsequent sub-events of the touch-based gesture. In this situation, other event recognizers, if any, that remain active for the hit view continue to track and process sub-events of an ongoing touch-based gesture.

[0140] In some embodiments, a respective event recognizer **280** includes metadata **283** with configurable properties, flags, and/or lists that indicate how the event delivery system should perform sub-event delivery to actively involved event recognizers. In some embodiments, metadata **283** includes configurable properties, flags, and/or lists that indicate how event recognizers may interact, or are enabled to interact, with one another. In some embodiments, metadata **283** includes configurable properties, flags, and/or lists that indicate whether sub-events are delivered to varying levels in the view or programmatic hierarchy.

[0141] In some embodiments, a respective event recognizer **280** activates event handler **290** associated with an event when one or more particular sub-events of an event are recognized. In some embodiments, a respective event recognizer **280** delivers event information associated with the event to event handler **290**. Activating an event handler **290** is distinct from sending (and deferred sending) sub-events to a respective hit view. In some embodiments, event recognizer **280** throws a flag associated with the recognized event, and event handler **290** associated with the flag catches the flag and performs a predefined process.

[0142] In some embodiments, event delivery instructions **288** include sub-event delivery instructions that deliver event information about a sub-event without activating an event handler. Instead, the sub-event delivery instructions deliver event information to event handlers associated with the series of sub-events or to actively involved views. Event handlers associated with the series of sub-events or with actively involved views receive the event information and perform a predetermined process.

[0143] In some embodiments, data updater **276** creates and updates data used in application **236-1**. For example, data updater **276** updates the telephone number used in contacts module **237**, or stores a video file used in video player module. In some embodiments, object updater **277** creates and updates objects used in application **236-1**. For example, object updater **277** creates a new user-interface object or updates the position of a user-interface object. GUI updater **278** updates the GUI. For example, GUI updater **278** prepares display information and sends it to graphics module **232** for display on a touch-sensitive display.

[0144] In some embodiments, event handler(s) **290** includes or has access to data updater **276**, object updater **277**, and GUI updater **278**. In some embodiments, data updater **276**, object updater **277**, and GUI updater **278** are included in a single module of a respective application **236-1** or application view **291**. In other embodiments, they are included in two or more software modules.

[0145] It shall be understood that the foregoing discussion regarding event handling of user touches on touch-sensitive displays also applies to other forms of user inputs to operate multifunction devices **200** with input devices, not all of which are initiated on touch screens. For example, mouse movement and mouse button presses, optionally coordinated with single or multiple keyboard presses or holds; contact movements such as taps, drags, scrolls, etc. on touchpads; pen stylus inputs; movement of the device; oral instructions; detected eye movements; biometric inputs; and/or any combination thereof are optionally utilized as inputs corresponding to sub-events which define an event to be recognized.

[0146] FIG. **3** illustrates a portable multifunction device **200** having a touch screen **212** in accordance with some embodiments. The touch screen optionally displays one or more graphics within user interface (UI) **300**. In this embodiment, as well as others described below, a user is enabled to select one or more of the graphics by making a gesture on the graphics, for example, with one or more fingers **302** (not drawn to scale in the figure) or one or more styluses **303** (not drawn to scale in the figure). In some embodiments, selection of one or more graphics occurs when the user breaks contact with the one or more graphics. In some embodiments, the gesture optionally includes one or more taps, one or more swipes (from left to right, right to left, upward and/or downward), and/or a rolling of a finger (from right to left, left to right, upward and/or downward) that has made contact with device **200**. In some implementations or circumstances, inadvertent contact with a graphic does not select the graphic. For example, a swipe gesture that sweeps over an application icon optionally does not select the corresponding application when the gesture corresponding to selection is a tap.

[0147] Device **200** may also include one or more physical buttons, such as "home" or menu button **304**. As described previously, menu button **304** may be used to navigate to any application **236** in a set of applications that may be executed on device **200**. Alternatively, in some embodiments, the menu button is implemented as a soft key in a GUI displayed on touch screen **212**.

[0148] In one embodiment, device **200** includes touch screen **212**, menu button **304**, push button **306** for powering the device on/off and locking the device, volume adjustment button(s) **308**, subscriber identity module (SIM) card slot **310**, headset jack **312**, and docking/charging external port **224**. Push button **306** is, optionally, used to turn the power on/off on the device by depressing the button and holding the button in the depressed state for a predefined time interval; to lock the device by depressing the button and releasing the button before the predefined time interval has elapsed; and/or

to unlock the device or initiate an unlock process. In an alternative embodiment, device **200** also accepts verbal input for activation or deactivation of some functions through microphone **213**. Device **200** also, optionally, includes one or more contact intensity sensors **265** for detecting intensity of contacts on touch screen **212** and/or one or more tactile output generators **267** for generating tactile outputs for a user of device **200**.

[0149] FIG. **4** is a block diagram of an exemplary multifunction device with a display and a touch-sensitive surface in accordance with some embodiments. Device **400** need not be portable. In some embodiments, device **400** is a laptop computer, a desktop computer, a tablet computer, a multimedia player device, a navigation device, an educational device (such as a child's learning toy), a gaming system, or a control device (e.g., a home or industrial controller). Device **400** typically includes one or more processing units (CPUs) **410**, one or more network or other communications interfaces **460**, memory **470**, and one or more communication buses **420** for interconnecting these components. Communication buses **420** optionally include circuitry (sometimes called a chipset) that interconnects and controls communications between system components. Device **400** includes input/output (I/O) interface **430** comprising display **440**, which is typically a touch screen display. I/O interface **430** also optionally includes a keyboard and/or mouse (or other pointing device) **450** and touchpad **455**, tactile output generator **457** for generating tactile outputs on device **400** (e.g., similar to tactile output generator(s) **267** described above with reference to FIG. **2**A), sensors **459** (e.g., optical, acceleration, proximity, touch-sensitive, and/or contact intensity sensors similar to contact intensity sensor(s) **265** described above with reference to FIG. **2**A). Memory **470** includes high-speed random access memory, such as DRAM, SRAM, DDR RAM, or other random access solid state memory devices; and optionally includes non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. Memory **470** optionally includes one or more storage devices remotely located from CPU(s) **410**. In some embodiments, memory **470** stores programs, modules, and data structures analogous to the programs, modules, and data structures stored in memory **202** of portable multifunction device **200** (FIG. **2**A), or a subset thereof. Furthermore, memory **470** optionally stores additional programs, modules, and data structures not present in memory **202** of portable multifunction device **200**. For example, memory **470** of device **400** optionally stores drawing module **480**, presentation module **482**, word processing module **484**, website creation module **486**, disk authoring module **488**, and/or spreadsheet module **490**, while memory **202** of portable multifunction device **200** (FIG. **2**A) optionally does not store these modules.

[0150] Each of the above-identified elements in FIG. **4** may be stored in one or more of the previously mentioned memory devices. Each of the above-identified modules corresponds to a set of instructions for performing a function described above. The above-identified modules or programs (e.g., sets of instructions) need not be implemented as separate software programs, procedures, or modules, and thus various subsets of these modules may be combined or otherwise rearranged in various embodiments. In some embodiments, memory **470** may store a subset of the modules and data structures identified above. Furthermore, memory **470** may store additional modules and data structures not described above.

[0151] Attention is now directed towards embodiments of user interfaces that may be implemented on, for example, portable multifunction device **200**.

[0152] FIG. **5**A illustrates an exemplary user interface for a menu of applications on portable multifunction device **200** in accordance with some embodiments. Similar user interfaces may be implemented on device **400**. In some embodiments, user interface **500** includes the following elements, or a subset or superset thereof:

[0153] Signal strength indicator(s) **502** for wireless communication(s), such as cellular and Wi-Fi signals;

- **[0154]** Time **504**;
- **[0155]** Bluetooth indicator **505**;
- **[0156]** Battery status indicator **506**;
- **[0157]** Tray **508** with icons for frequently used applications, such as:
  - **[0158]** Icon **516** for telephone module **238**, labeled "Phone," which optionally includes an indicator **514** of the number of missed calls or voicemail messages;
  - **[0159]** Icon **518** for e-mail client module **240**, labeled "Mail," which optionally includes an indicator **510** of the number of unread e-mails;
  - **[0160]** Icon **520** for browser module **247**, labeled "Browser;" and
  - **[0161]** Icon **522** for video and music player module **252**, also referred to as iPod (trademark of Apple Inc.) module **252**, labeled "iPod;" and
- **[0162]** Icons for other applications, such as:
  - **[0163]** Icon **524** for IM module **241**, labeled "Messages;"
  - **[0164]** Icon **526** for calendar module **248**, labeled "Calendar;"
  - **[0165]** Icon **528** for image management module **244**, labeled "Photos;"
  - **[0166]** Icon **530** for camera module **243**, labeled "Camera;"
  - **[0167]** Icon **532** for online video module **255**, labeled "Online Video;"
  - **[0168]** Icon **534** for stocks widget **249-2**, labeled "Stocks;"
  - **[0169]** Icon **536** for map module **254**, labeled "Maps;"
  - **[0170]** Icon **538** for weather widget **249-1**, labeled "Weather;"
  - **[0171]** Icon **540** for alarm clock widget **249-4**, labeled "Clock;"
  - **[0172]** Icon **542** for workout support module **242**, labeled "Workout Support;"
  - **[0173]** Icon **544** for notes module **253**, labeled "Notes;" and
  - **[0174]** Icon **546** for a settings application or module, labeled "Settings," which provides access to settings for device **200** and its various applications **236**.

[0175] It should be noted that the icon labels illustrated in FIG. **5**A are merely exemplary. For example, icon **522** for video and music player module **252** may optionally be labeled "Music" or "Music Player." Other labels are, optionally, used for various application icons. In some embodiments, a label for a respective application icon includes a name of an application corresponding to the respective application icon. In some embodiments, a label for a particular application icon is distinct from a name of an application corresponding to the particular application icon.

[0176] FIG. **5**B illustrates an exemplary user interface on a device (e.g., device **400**, FIG. **4**) with a touch-sensitive surface **551** (e.g., a tablet or touchpad **455**, FIG. **4**) that is separate from the display **550** (e.g., touch screen display **212**). Device **400** also, optionally, includes one or more contact intensity sensors (e.g., one or more of sensors **457**) for detecting intensity of contacts on touch-sensitive surface **551** and/or one or more tactile output generators **459** for generating tactile outputs for a user of device **400**.

[0177] Although some of the examples which follow will be given with reference to inputs on touch screen display **212** (where the touch-sensitive surface and the display are combined), in some embodiments, the device detects inputs on a touch-sensitive surface that is separate from the display, as shown in FIG. **5**B. In some embodiments, the touch-sensitive surface (e.g., **551** in FIG. **5**B) has a primary axis (e.g., **552** in FIG. **5**B) that corresponds to a primary axis (e.g., **553** in FIG. **5**B) on the display (e.g., **550**). In accordance with these embodiments, the device detects contacts (e.g., **560** and **562** in FIG. **5**B) with the touch-sensitive surface **551** at locations that correspond to respective locations on the display (e.g., in FIG. **5**B, **560** corresponds to **568** and **562** corresponds to **570**). In this way, user inputs (e.g., contacts **560** and **562**, and movements thereof) detected by the device on the touch-sensitive surface (e.g., **551** in FIG. **5**B) are used by the device to manipulate the user interface on the display (e.g., **550** in FIG. **5**B) of the multifunction device when the touch-sensitive surface is separate from the display. It should be understood that similar methods are, optionally, used for other user interfaces described herein.

[0178] Additionally, while the following examples are given primarily with reference to finger inputs (e.g., finger contacts, finger tap gestures, finger swipe gestures), it should be understood that, in some embodiments, one or more of the finger inputs are replaced with input from another input device (e.g., a mouse-based input or stylus input). For example, a swipe gesture is, optionally, replaced with a mouse click (e.g., instead of a contact) followed by movement of the cursor along the path of the swipe (e.g., instead of movement of the contact). As another example, a tap gesture is, optionally, replaced with a mouse click while the cursor is located over the location of the tap gesture (e.g., instead of detection of the contact followed by ceasing to detect the contact). Similarly, when multiple user inputs are simultaneously detected, it should be understood that multiple computer mice are, optionally, used simultaneously, or a mouse and finger contacts are, optionally, used simultaneously.

[0179] FIG. **6**A illustrates exemplary personal electronic device **600**. Device **600** includes body **602**. In some embodiments, device **600** can include some or all of the features described with respect to devices **200** and **400** (e.g., FIGS. **2**A-**4**B). In some embodiments, device **600** has touch-sensitive display screen **604**, hereafter touch screen **604**. Alternatively, or in addition to touch screen **604**, device **600** has a display and a touch-sensitive surface. As with devices **200** and **400**, in some embodiments, touch screen **604** (or the touch-sensitive surface) may have one or more intensity sensors for detecting intensity of contacts (e.g., touches) being applied. The one or more intensity sensors of touch screen **604** (or the touch-sensitive surface) can provide output data that represents the intensity of touches. The user interface of device **600** can respond to touches based on their intensity, meaning that touches of different intensities can invoke different user interface operations on device **600**.

[0180] Techniques for detecting and processing touch intensity may be found, for example, in related applications: International Patent Application Serial No. PCT/US2013/040061, titled "Device, Method, and Graphical User Interface for Displaying User Interface Objects Corresponding to an Application," filed May 8, 2013, and International Patent Application Serial No. PCT/US2013/069483, titled "Device, Method, and Graphical User Interface for Transitioning Between Touch Input to Display Output Relationships," filed Nov. 11, 2013, each of which is hereby incorporated by reference in their entirety.

[0181] In some embodiments, device **600** has one or more input mechanisms **606** and **608**. Input mechanisms **606** and **608**, if included, can be physical. Examples of physical input mechanisms include push buttons and rotatable mechanisms. In some embodiments, device **600** has one or more attachment mechanisms. Such attachment mechanisms, if included, can permit attachment of device **600** with, for example, hats, eyewear, earrings, necklaces, shirts, jackets, bracelets, watch straps, chains, trousers, belts, shoes, purses, backpacks, and so forth. These attachment mechanisms may permit device **600** to be worn by a user.

[0182] FIG. **6**B depicts exemplary personal electronic device **600**. In some embodiments, device **600** can include some or all of the components described with respect to FIGS. **2**A, **2**B, and **4**. Device **600** has bus **612** that operatively couples I/O section **614** with one or more computer processors **616** and memory **618**. I/O section **614** can be connected to display **604**, which can have touch-sensitive component **622** and, optionally, touch-intensity sensitive component **624**. In addition, I/O section **614** can be connected with communication unit **630** for receiving application and operating system data, using Wi-Fi, Bluetooth, near field communication (NFC), cellular, and/or other wireless communication techniques. Device **600** can include input mechanisms **606** and/or **608**. Input mechanism **606** may be a rotatable input device or a depressible and rotatable input device, for example. Input mechanism **608** may be a button, in some examples.

[0183] Input mechanism **608** may be a microphone, in some examples. Personal electronic device **600** can include various sensors, such as GPS sensor **632**, accelerometer **634**, directional sensor **640** (e.g., compass), gyroscope **636**, motion sensor **638**, and/or a combination thereof, all of which can be operatively connected to I/O section **614**.

[0184] Memory **618** of personal electronic device **600** can be a non-transitory computer-readable storage medium, for storing computer-executable instructions, which, when executed by one or more computer processors **616**, for example, can cause the computer processors to perform the techniques described below, including process **800** (FIGS. **8**A-D). The computer-executable instructions can also be stored and/or transported within any non-transitory computer-readable storage medium for use by or in connection with an instruction execution system, apparatus, or device, such as a computer-based system, processor-containing system, or other system that can fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions. For purposes of this document, a "non-transitory computer-readable storage medium" can be any medium that can tangibly contain or store computer-executable instructions for use by or in connection with the instruction execution system, apparatus, or device. The non-transitory computer-readable storage medium can include, but is not limited to, magnetic, optical, and/or semiconductor

storages. Examples of such storage include magnetic disks, optical discs based on CD, DVD, or Blu-ray technologies, as well as persistent solid-state memory such as flash, solid-state drives, and the like. Personal electronic device **600** is not limited to the components and configuration of FIG. **6**B, but can include other or additional components in multiple configurations.

[0185] As used here, the term "affordance" refers to a user-interactive graphical user interface object that may be displayed on the display screen of devices **200**, **400**, and/or **600** (FIGS. **2**, **4**, and **6**). For example, an image (e.g., icon), a button, and text (e.g., hyperlink) may each constitute an affordance.

[0186] As used herein, the term "focus selector" refers to an input element that indicates a current part of a user interface with which a user is interacting. In some implementations that include a cursor or other location marker, the cursor acts as a "focus selector" so that when an input (e.g., a press input) is detected on a touch-sensitive surface (e.g., touchpad **455** in FIG. **4** or touch-sensitive surface **551** in FIG. **5**B) while the cursor is over a particular user interface element (e.g., a button, window, slider or other user interface element), the particular user interface element is adjusted in accordance with the detected input. In some implementations that include a touch screen display (e.g., touch-sensitive display system **212** in FIG. **2**A or touch screen **212** in FIG. **5**A) that enables direct interaction with user interface elements on the touch screen display, a detected contact on the touch screen acts as a "focus selector" so that when an input (e.g., a press input by the contact) is detected on the touch screen display at a location of a particular user interface element (e.g., a button, window, slider, or other user interface element), the particular user interface element is adjusted in accordance with the detected input. In some implementations, focus is moved from one region of a user interface to another region of the user interface without corresponding movement of a cursor or movement of a contact on a touch screen display (e.g., by using a tab key or arrow keys to move focus from one button to another button); in these implementations, the focus selector moves in accordance with movement of focus between different regions of the user interface. Without regard to the specific form taken by the focus selector, the focus selector is generally the user interface element (or contact on a touch screen display) that is controlled by the user so as to communicate the user's intended interaction with the user interface (e.g., by indicating, to the device, the element of the user interface with which the user is intending to interact). For example, the location of a focus selector (e.g., a cursor, a contact, or a selection box) over a respective button while a press input is detected on the touch-sensitive surface (e.g., a touchpad or touch screen) will indicate that the user is intending to activate the respective button (as opposed to other user interface elements shown on a display of the device).

[0187] As used in the specification and claims, the term "characteristic intensity" of a contact refers to a characteristic of the contact based on one or more intensities of the contact. In some embodiments, the characteristic intensity is based on multiple intensity samples. The characteristic intensity is, optionally, based on a predefined number of intensity samples, or a set of intensity samples collected during a predetermined time period (e.g., 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10 seconds) relative to a predefined event (e.g., after detecting the contact, prior to detecting liftoff of the contact, before or after detecting a start of movement of the contact, prior to detecting an end of the contact, before or after detecting an increase in intensity of the contact, and/or before or after detecting a decrease in intensity of the contact). A characteristic intensity of a contact is, optionally based on one or more of: a maximum value of the intensities of the contact, a mean value of the intensities of the contact, an average value of the intensities of the contact, a top 10 percentile value of the intensities of the contact, a value at the half maximum of the intensities of the contact, a value at the 90 percent maximum of the intensities of the contact, or the like. In some embodiments, the duration of the contact is used in determining the characteristic intensity (e.g., when the characteristic intensity is an average of the intensity of the contact over time). In some embodiments, the characteristic intensity is compared to a set of one or more intensity thresholds to determine whether an operation has been performed by a user. For example, the set of one or more intensity thresholds may include a first intensity threshold and a second intensity threshold. In this example, a contact with a characteristic intensity that does not exceed the first threshold results in a first operation, a contact with a characteristic intensity that exceeds the first intensity threshold and does not exceed the second intensity threshold results in a second operation, and a contact with a characteristic intensity that exceeds the second threshold results in a third operation. In some embodiments, a comparison between the characteristic intensity and one or more thresholds is used to determine whether or not to perform one or more operations (e.g., whether to perform a respective operation or forgo performing the respective operation) rather than being used to determine whether to perform a first operation or a second operation.

[0188] In some embodiments, a portion of a gesture is identified for purposes of determining a characteristic intensity. For example, a touch-sensitive surface may receive a continuous swipe contact transitioning from a start location and reaching an end location, at which point the intensity of the contact increases. In this example, the characteristic intensity of the contact at the end location may be based on only a portion of the continuous swipe contact, and not the entire swipe contact (e.g., only the portion of the swipe contact at the end location). In some embodiments, a smoothing algorithm may be applied to the intensities of the swipe contact prior to determining the characteristic intensity of the contact. For example, the smoothing algorithm optionally includes one or more of: an unweighted sliding-average smoothing algorithm, a triangular smoothing algorithm, a median filter smoothing algorithm, and/or an exponential smoothing algorithm. In some circumstances, these smoothing algorithms eliminate narrow spikes or dips in the intensities of the swipe contact for purposes of determining a characteristic intensity.

[0189] The intensity of a contact on the touch-sensitive surface may be characterized relative to one or more intensity thresholds, such as a contact-detection intensity threshold, a light press intensity threshold, a deep press intensity threshold, and/or one or more other intensity thresholds. In some embodiments, the light press intensity threshold corresponds to an intensity at which the device will perform operations typically associated with clicking a button of a physical mouse or a trackpad. In some embodiments, the deep press intensity threshold corresponds to an intensity at which the device will perform operations that are different from operations typically associated with clicking a button of a physical mouse or a trackpad. In some embodiments, when a contact is

detected with a characteristic intensity below the light press intensity threshold (e.g., and above a nominal contact-detection intensity threshold below which the contact is no longer detected), the device will move a focus selector in accordance with movement of the contact on the touch-sensitive surface without performing an operation associated with the light press intensity threshold or the deep press intensity threshold. Generally, unless otherwise stated, these intensity thresholds are consistent between different sets of user interface figures.

[0190] An increase of characteristic intensity of the contact from an intensity below the light press intensity threshold to an intensity between the light press intensity threshold and the deep press intensity threshold is sometimes referred to as a "light press" input. An increase of characteristic intensity of the contact from an intensity below the deep press intensity threshold to an intensity above the deep press intensity threshold is sometimes referred to as a "deep press" input. An increase of characteristic intensity of the contact from an intensity below the contact-detection intensity threshold to an intensity between the contact-detection intensity threshold and the light press intensity threshold is sometimes referred to as detecting the contact on the touch-surface. A decrease of characteristic intensity of the contact from an intensity above the contact-detection intensity threshold to an intensity below the contact-detection intensity threshold is sometimes referred to as detecting liftoff of the contact from the touch-surface. In some embodiments, the contact-detection intensity threshold is zero. In some embodiments, the contact-detection intensity threshold is greater than zero.

[0191] In some embodiments described herein, one or more operations are performed in response to detecting a gesture that includes a respective press input or in response to detecting the respective press input performed with a respective contact (or a plurality of contacts), where the respective press input is detected based at least in part on detecting an increase in intensity of the contact (or plurality of contacts) above a press-input intensity threshold. In some embodiments, the respective operation is performed in response to detecting the increase in intensity of the respective contact above the press-input intensity threshold (e.g., a "down stroke" of the respective press input). In some embodiments, the press input includes an increase in intensity of the respective contact above the press-input intensity threshold and a subsequent decrease in intensity of the contact below the press-input intensity threshold, and the respective operation is performed in response to detecting the subsequent decrease in intensity of the respective contact below the press-input threshold (e.g., an "up stroke" of the respective press input).

[0192] In some embodiments, the device employs intensity hysteresis to avoid accidental inputs sometimes termed "jitter," where the device defines or selects a hysteresis intensity threshold with a predefined relationship to the press-input intensity threshold (e.g., the hysteresis intensity threshold is X intensity units lower than the press-input intensity threshold or the hysteresis intensity threshold is 75%, 90%, or some reasonable proportion of the press-input intensity threshold). Thus, in some embodiments, the press input includes an increase in intensity of the respective contact above the press-input intensity threshold and a subsequent decrease in intensity of the contact below the hysteresis intensity threshold that corresponds to the press-input intensity threshold, and the respective operation is performed in response to detecting the subsequent decrease in intensity of the respective contact below the hysteresis intensity threshold (e.g., an "up stroke" of the respective press input). Similarly, in some embodiments, the press input is detected only when the device detects an increase in intensity of the contact from an intensity at or below the hysteresis intensity threshold to an intensity at or above the press-input intensity threshold and, optionally, a subsequent decrease in intensity of the contact to an intensity at or below the hysteresis intensity, and the respective operation is performed in response to detecting the press input (e.g., the increase in intensity of the contact or the decrease in intensity of the contact, depending on the circumstances).

[0193] For ease of explanation, the descriptions of operations performed in response to a press input associated with a press-input intensity threshold or in response to a gesture including the press input are, optionally, triggered in response to detecting either: an increase in intensity of a contact above the press-input intensity threshold, an increase in intensity of a contact from an intensity below the hysteresis intensity threshold to an intensity above the press-input intensity threshold, a decrease in intensity of the contact below the press-input intensity threshold, and/or a decrease in intensity of the contact below the hysteresis intensity threshold corresponding to the press-input intensity threshold. Additionally, in examples where an operation is described as being performed in response to detecting a decrease in intensity of a contact below the press-input intensity threshold, the operation is, optionally, performed in response to detecting a decrease in intensity of the contact below a hysteresis intensity threshold corresponding to, and lower than, the press-input intensity threshold.

3. Digital Assistant System

[0194] FIG. **7**A illustrates a block diagram of digital assistant system **700** in accordance with various examples. In some examples, digital assistant system **700** can be implemented on a standalone computer system. In some examples, digital assistant system **700** can be distributed across multiple computers. In some examples, some of the modules and functions of the digital assistant can be divided into a server portion and a client portion, where the client portion resides on one or more user devices (e.g., devices **104**, **122**, **200**, **400**, or **600**) and communicates with the server portion (e.g., server system **108**) through one or more networks, e.g., as shown in FIG. **1**. In some examples, digital assistant system **700** can be an implementation of server system **108** (and/or DA server **106**) shown in FIG. **1**. It should be noted that digital assistant system **700** is only one example of a digital assistant system, and that digital assistant system **700** can have more or fewer components than shown, may combine two or more components, or may have a different configuration or arrangement of the components. The various components shown in FIG. **7**A can be implemented in hardware, software instructions for execution by one or more processors, firmware, including one or more signal processing and/or application specific integrated circuits, or a combination thereof.

[0195] Digital assistant system **700** can include memory **702**, one or more processors **704**, input/output (I/O) interface **706**, and network communications interface **708**. These components can communicate with one another over one or more communication buses or signal lines **710**.

[0196] In some examples, memory **702** can include a non-transitory computer-readable medium, such as high-speed random access memory and/or a non-volatile computer-read-

able storage medium (e.g., one or more magnetic disk storage devices, flash memory devices, or other non-volatile solid-state memory devices).

[0197] In some examples, I/O interface **706** can couple input/output devices **716** of digital assistant system **700**, such as displays, keyboards, touch screens, and microphones, to user interface module **722**. I/O interface **706**, in conjunction with user interface module **722**, can receive user inputs (e.g., voice input, keyboard inputs, touch inputs, etc.) and processes them accordingly. In some examples, e.g., when the digital assistant is implemented on a standalone user device, digital assistant system **700** can include any of the components and I/O communication interfaces described with respect to devices **200**, **400**, or **600** in FIGS. **2**A, **4**, **6**A-B, respectively. In some examples, digital assistant system **700** can represent the server portion of a digital assistant implementation, and can interact with the user through a client-side portion residing on a user device (e.g., devices **104**, **200**, **400**, or **600**).

[0198] In some examples, the network communications interface **708** can include wired communication port(s) **712** and/or wireless transmission and reception circuitry **714**. The wired communication port(s) can receive and send communication signals via one or more wired interfaces, e.g., Ethernet, Universal Serial Bus (USB), FIREWIRE, etc. The wireless circuitry **714** can receive and send RF signals and/or optical signals from/to communications networks and other communications devices. The wireless communications can use any of a plurality of communications standards, protocols, and technologies, such as GSM, EDGE, CDMA, TDMA, Bluetooth, Wi-Fi, VoIP, Wi-MAX, or any other suitable communication protocol. Network communications interface **708** can enable communication between digital assistant system **700** with networks, such as the Internet, an intranet, and/or a wireless network, such as a cellular telephone network, a wireless local area network (LAN), and/or a metropolitan area network (MAN), and other devices.

[0199] In some examples, memory **702**, or the computer-readable storage media of memory **702**, can store programs, modules, instructions, and data structures including all or a subset of: operating system **718**, communications module **720**, user interface module **722**, one or more applications **724**, and digital assistant module **726**. In particular, memory **702**, or the computer-readable storage media of memory **702**, can store instructions for performing process **800**, described below. One or more processors **704** can execute these programs, modules, and instructions, and reads/writes from/to the data structures.

[0200] Operating system **718** (e.g., Darwin, RTXC, LINUX, UNIX, iOS, OS X, WINDOWS, or an embedded operating system such as VxWorks) can include various software components and/or drivers for controlling and managing general system tasks (e.g., memory management, storage device control, power management, etc.) and facilitates communications between various hardware, firmware, and software components.

[0201] Communications module **720** can facilitate communications between digital assistant system **700** with other devices over network communications interface **708**. For example, communications module **720** can communicate with RF circuitry **208** of electronic devices such as devices **200**, **400**, and **600** shown in FIG. **2**A, **4**, **6**A-B, respectively. Communications module **720** can also include various components for handling data received by wireless circuitry **714** and/or wired communications port **712**.

[0202] User interface module **722** can receive commands and/or inputs from a user via I/O interface **706** (e.g., from a keyboard, touch screen, pointing device, controller, and/or microphone), and generate user interface objects on a display. User interface module **722** can also prepare and deliver outputs (e.g., speech, sound, animation, text, icons, vibrations, haptic feedback, light, etc.) to the user via the I/O interface **706** (e.g., through displays, audio channels, speakers, touchpads, etc.).

[0203] Applications **724** can include programs and/or modules that are configured to be executed by one or more processors **704**. For example, if the digital assistant system is implemented on a standalone user device, applications **724** can include user applications, such as games, a calendar application, a navigation application, or an email application. If digital assistant system **700** is implemented on a server, applications **724** can include resource management applications, diagnostic applications, or scheduling applications, for example.

[0204] Memory **702** can also store digital assistant module **726** (or the server portion of a digital assistant). In some examples, digital assistant module **726** can include the following sub-modules, or a subset or superset thereof: input/output processing module **728**, speech-to-text (STT) processing module **730**, natural language processing module **732**, dialogue flow processing module **734**, task flow processing module **736**, service processing module **738**, and speech synthesis module **740**. Each of these modules can have access to one or more of the following systems or data and models of the digital assistant module **726**, or a subset or superset thereof: ontology **760**, vocabulary index **744**, user data **748**, task flow models **754**, service models **756**, and ASR systems.

[0205] In some examples, using the processing modules, data, and models implemented in digital assistant module **726**, the digital assistant can perform at least some of the following: converting speech input into text; identifying a user's intent expressed in a natural language input received from the user; actively eliciting and obtaining information needed to fully infer the user's intent (e.g., by disambiguating words, games, intentions, etc.); determining the task flow for fulfilling the inferred intent; and executing the task flow to fulfill the inferred intent.

[0206] In some examples, as shown in FIG. **7**B, I/O processing module **728** can interact with the user through I/O devices **716** in FIG. **7**A or with a user device (e.g., devices **104**, **200**, **400**, or **600**) through network communications interface **708** in FIG. **7**A to obtain user input (e.g., a speech input) and to provide responses (e.g., as speech outputs) to the user input. I/O processing module **728** can optionally obtain contextual information associated with the user input from the user device, along with or shortly after the receipt of the user input. The contextual information can include user-specific data, vocabulary, and/or preferences relevant to the user input. In some examples, the contextual information also includes software and hardware states of the user device at the time the user request is received, and/or information related to the surrounding environment of the user at the time that the user request was received. In some examples, I/O processing module **728** can also send follow-up questions to, and receive answers from, the user regarding the user request. When a user request is received by I/O processing module **728** and the user request can include speech input, I/O processing module **728** can forward the speech input to STT processing module **730** (or speech recognizer) for speech-to-text conversions.

[0207] STT processing module **730** can include one or more ASR systems. The one or more ASR systems can process the speech input that is received through I/O processing module **728** to produce a recognition result. Each ASR system can include a front-end speech pre-processor. The front-end speech pre-processor can extract representative features from the speech input. For example, the front-end speech pre-processor can perform a Fourier transform on the speech input to extract spectral features that characterize the speech input as a sequence of representative multi-dimensional vectors. Further, each ASR system can include one or more speech recognition models (e.g., acoustic models and/or language models) and can implement one or more speech recognition engines. Examples of speech recognition models can include Hidden Markov Models, Gaussian-Mixture Models, Deep Neural Network Models, n-gram language models, and other statistical models. Examples of speech recognition engines can include the dynamic time warping based engines and weighted finite-state transducers (WFST) based engines. The one or more speech recognition models and the one or more speech recognition engines can be used to process the extracted representative features of the front-end speech pre-processor to produce intermediate recognitions results (e.g., phonemes, phonemic strings, and sub-words), and ultimately, text recognition results (e.g., words, word strings, or sequence of tokens). In some examples, the speech input can be processed at least partially by a third-party service or on the user's device (e.g., device **104**, **200**, **400**, or **600**) to produce the recognition result. Once STT processing module **730** produces recognition results containing a text string (e.g., words, or sequence of words, or sequence of tokens), the recognition result can be passed to natural language processing module **732** for intent deduction.

[0208] More details on the speech-to-text processing are described in U.S. Utility application Ser. No. 13/236,942 for "Consolidating Speech Recognition Results," filed on Sep. 20, 2011, the entire disclosure of which is incorporated herein by reference.

[0209] In some examples, STT processing module **730** can include and/or access a vocabulary of recognizable words via phonetic alphabet conversion module **731**. Each vocabulary word can be associated with one or more candidate pronunciations of the word represented in a speech recognition phonetic alphabet. In particular, the vocabulary of recognizable words can include a word that is associated with a plurality of candidate pronunciations. For example, the vocabulary may include the word "tomato" that is associated with the candidate pronunciations of tə'meɪɾoʊ and tə'mɑtoʊ. Further, vocabulary words can be associated with custom candidate pronunciations that are based on previous speech inputs from the user. Such custom candidate pronunciations can be stored in STT processing module **730** and can be associated with a particular user via the user's profile on the device. In some examples, the candidate pronunciations for words can be determined based on the spelling of the word and one or more linguistic and/or phonetic rules. In some examples, the candidate pronunciations can be manually generated, e.g., based on known canonical pronunciations.

[0210] In some examples, the candidate pronunciations can be ranked based on the commonness of the candidate pronunciation. For example, the candidate pronunciation tə'meɪɾoʊ can be ranked higher than tə'mɑtoʊ, because the former is a more commonly used pronunciation (e.g., among all users, for users in a particular geographical region, or for any other appropriate subset of users). In some examples, candidate pronunciations can be ranked based on whether the candidate pronunciation is a custom candidate pronunciation associated with the user. For example, custom candidate pronunciations can be ranked higher than canonical candidate pronunciations. This can be useful for recognizing proper nouns having a unique pronunciation that deviates from canonical pronunciation. In some examples, candidate pronunciations can be associated with one or more speech characteristics, such as geographic origin, nationality, or ethnicity. For example, the candidate pronunciation tə'meɪɾoʊ can be associated with the United States, whereas the candidate pronunciation tə'mɑtoʊ can be associated with Great Britain. Further, the rank of the candidate pronunciation can be based on one or more characteristics (e.g., geographic origin, nationality, ethnicity, etc.) of the user stored in the user's profile on the device. For example, it can be determined from the user's profile that the user is associated with the United States. Based on the user being associated with the United States, the candidate pronunciation tə'meɪɾoʊ (associated with the United States) can be ranked higher than the candidate pronunciation tə'mɑtoʊ (associated with Great Britain). In some examples, one of the ranked candidate pronunciations can be selected as a predicted pronunciation (e.g., the most likely pronunciation).

[0211] When a speech input is received, STT processing module **730** can be used to determine the phonemes corresponding to the speech input (e.g., using an acoustic model), and then attempt to determine words that match the phonemes (e.g., using a language model). For example, if STT processing module **730** can first identify the sequence of phonemes tə'meɪɾoʊ corresponding to a portion of the speech input, it can then determine, based on vocabulary index **744**, that this sequence corresponds to the word "tomato."

[0212] In some examples, STT processing module **730** can use approximate matching techniques to determine words in an utterance. Thus, for example, the STT processing module **730** can determine that the sequence of phonemes tə'meɪɾoʊ corresponds to the word "tomato," even if that particular sequence of phonemes is not one of the candidate sequence of phonemes for that word.

[0213] In some examples, natural language processing module **732** can be configured to receive metadata associated with the speech input. The metadata can indicate whether to perform natural language processing on the speech input (or the sequence of words or tokens corresponding to the speech input). If the metadata indicates that natural language processing is to be performed, then the natural language processing module can receive the sequence of words or tokens from the STT processing module to perform natural language processing. However, if the metadata indicates that natural language process is not to be performed, then the natural language processing module can be disabled and the sequence of words or tokens (e.g., text string) from the STT processing module can be outputted from the digital assistant. In some examples, the metadata can further identify one or more domains corresponding to the user request. Based on the one or more domains, the natural language processor can disable domains in ontology **760** other than the one or more domains. In this way, natural language processing is constrained to the one or more domains in ontology **760**. In particular, the structure query (described below) can be generated using the one or more domains and not the other domains in the ontology.

[0214] Natural language processing module **732** ("natural language processor") of the digital assistant can take the sequence of words or tokens ("token sequence") generated by STT processing module **730**, and attempt to associate the token sequence with one or more "actionable intents" recognized by the digital assistant. An "actionable intent" can represent a task that can be performed by the digital assistant, and can have an associated task flow implemented in task flow models **754**. The associated task flow can be a series of programmed actions and steps that the digital assistant takes in order to perform the task. The scope of a digital assistant's capabilities can be dependent on the number and variety of task flows that have been implemented and stored in task flow models **754**, or in other words, on the number and variety of "actionable intents" that the digital assistant recognizes. The effectiveness of the digital assistant, however, can also be dependent on the assistant's ability to infer the correct "actionable intent(s)" from the user request expressed in natural language.

[0215] In some examples, in addition to the sequence of words or tokens obtained from STT processing module **730**, natural language processing module **732** can also receive contextual information associated with the user request, e.g., from I/O processing module **728**. The natural language processing module **732** can optionally use the contextual information to clarify, supplement, and/or further define the information contained in the token sequence received from STT processing module **730**. The contextual information can include, for example, user preferences, hardware, and/or software states of the user device, sensor information collected before, during, or shortly after the user request, prior interactions (e.g., dialogue) between the digital assistant and the user, and the like. As described herein, contextual information can be dynamic, and can change with time, location, content of the dialogue, and other factors.

[0216] In some examples, the natural language processing can be based on, e.g., ontology **760**. Ontology **760** can be a hierarchical structure containing many nodes, each node representing either an "actionable intent" or a "property" relevant to one or more of the "actionable intents" or other "properties." As noted above, an "actionable intent" can represent a task that the digital assistant is capable of performing, i.e., it is "actionable" or can be acted on. A "property" can represent a parameter associated with an actionable intent or a sub-aspect of another property. A linkage between an actionable intent node and a property node in ontology **760** can define how a parameter represented by the property node pertains to the task represented by the actionable intent node.

[0217] In some examples, ontology **760** can be made up of actionable intent nodes and property nodes. Within ontology **760**, each actionable intent node can be linked to one or more property nodes either directly or through one or more intermediate property nodes. Similarly, each property node can be linked to one or more actionable intent nodes either directly or through one or more intermediate property nodes. For example, as shown in FIG. **7**C, ontology **760** can include a "restaurant reservation" node (i.e., an actionable intent node). Property nodes "restaurant," "date/time" (for the reservation), and "party size" can each be directly linked to the actionable intent node (i.e., the "restaurant reservation" node).

[0218] In addition, property nodes "cuisine," "price range," "phone number," and "location" can be sub-nodes of the property node "restaurant," and can each be linked to the "restaurant reservation" node (i.e., the actionable intent node) through the intermediate property node "restaurant." For another example, as shown in FIG. **7**C, ontology **760** can also include a "set reminder" node (i.e., another actionable intent node). Property nodes "date/time" (for setting the reminder) and "subject" (for the reminder) can each be linked to the "set reminder" node. Since the property "date/time" can be relevant to both the task of making a restaurant reservation and the task of setting a reminder, the property node "date/time" can be linked to both the "restaurant reservation" node and the "set reminder" node in ontology **760**.

[0219] An actionable intent node, along with its linked concept nodes, can be described as a "domain." In the present discussion, each domain can be associated with a respective actionable intent, and refers to the group of nodes (and the relationships there between) associated with the particular actionable intent. For example, ontology **760** shown in FIG. **7**C can include an example of restaurant reservation domain **762** and an example of reminder domain **764** within ontology **760**. The restaurant reservation domain includes the actionable intent node "restaurant reservation," property nodes "restaurant," "date/time," and "party size," and sub-property nodes "cuisine," "price range," "phone number," and "location." Reminder domain **764** can include the actionable intent node "set reminder," and property nodes "subject" and "date/time." In some examples, ontology **760** can be made up of many domains. Each domain can share one or more property nodes with one or more other domains. For example, the "date/time" property node can be associated with many different domains (e.g., a scheduling domain, a travel reservation domain, a movie ticket domain, etc.), in addition to restaurant reservation domain **762** and reminder domain **764**.

[0220] While FIG. **7**C illustrates two example domains within ontology **760**, other domains can include, for example, "find a movie," "initiate a phone call," "find directions," "schedule a meeting," "send a message," and "provide an answer to a question," "read a list," "providing navigation instructions," "provide instructions for a task" and so on. A "send a message" domain can be associated with a "send a message" actionable intent node, and may further include property nodes such as "recipient(s)," "message type," and "message body." The property node "recipient" can be further defined, for example, by the sub-property nodes such as "recipient name" and "message address."

[0221] In some examples, ontology **760** can include all the domains (and hence actionable intents) that the digital assistant is capable of understanding and acting upon. In some examples, ontology **760** can be modified, such as by adding or removing entire domains or nodes, or by modifying relationships between the nodes within the ontology **760**.

[0222] In some examples, nodes associated with multiple related actionable intents can be clustered under a "super domain" in ontology **760**. For example, a "travel" super-domain can include a cluster of property nodes and actionable intent nodes related to travel. The actionable intent nodes related to travel can include "airline reservation," "hotel reservation," "car rental," "get directions," "find points of interest," and so on. The actionable intent nodes under the same super domain (e.g., the "travel" super domain) can have many property nodes in common. For example, the actionable intent nodes for "airline reservation," "hotel reservation," "car rental," "get directions," and "find points of interest" can

share one or more of the property nodes "start location," "destination," "departure date/time," "arrival date/time," and "party size."

[0223] In some examples, each node in ontology **760** can be associated with a set of words and/or phrases that are relevant to the property or actionable intent represented by the node. The respective set of words and/or phrases associated with each node can be the so-called "vocabulary" associated with the node. The respective set of words and/or phrases associated with each node can be stored in vocabulary index **744** in association with the property or actionable intent represented by the node. For example, returning to FIG. **7**B, the vocabulary associated with the node for the property of "restaurant" can include words such as "food," "drinks," "cuisine," "hungry," "eat," "pizza," "fast food," "meal," and so on. For another example, the vocabulary associated with the node for the actionable intent of "initiate a phone call" can include words and phrases such as "call," "phone," "dial," "ring," "call this number," "make a call to," and so on. The vocabulary index **744** can optionally include words and phrases in different languages.

[0224] Natural language processing module **732** can receive the token sequence (e.g., a text string) from STT processing module **730**, and determine what nodes are implicated by the words in the token sequence. In some examples, if a word or phrase in the token sequence is found to be associated with one or more nodes in ontology **760** (via vocabulary index **744**), the word or phrase can "trigger" or "activate" those nodes. Based on the quantity and/or relative importance of the activated nodes, natural language processing module **732** can select one of the actionable intents as the task that the user intended the digital assistant to perform. In some examples, the domain that has the most "triggered" nodes can be selected. In some examples, the domain having the highest confidence value (e.g., based on the relative importance of its various triggered nodes) can be selected. In some examples, the domain can be selected based on a combination of the number and the importance of the triggered nodes. In some examples, additional factors are considered in selecting the node as well, such as whether the digital assistant has previously correctly interpreted a similar request from a user.

[0225] User data **748** can include user-specific information, such as user-specific vocabulary, user preferences, user address, user's default and secondary languages, user's contact list, and other short-term or long-term information for each user. In some examples, natural language processing module **732** can use the user-specific information to supplement the information contained in the user input to further define the user intent. For example, for a user request "invite my friends to my birthday party," natural language processing module **732** can be able to access user data **748** to determine who the "friends" are and when and where the "birthday party" would be held, rather than requiring the user to provide such information explicitly in his/her request.

[0226] Other details of searching an ontology based on a token string is described in U.S. Utility application Ser. No. 12/341,743 for "Method and Apparatus for Searching Using An Active Ontology," filed Dec. 22, 2008, the entire disclosure of which is incorporated herein by reference.

[0227] In some examples, once natural language processing module **732** identifies an actionable intent (or domain) based on the user request, natural language processing module **732** can generate a structured query to represent the identified actionable intent. In some examples, the structured query can include parameters for one or more nodes within the domain for the actionable intent, and at least some of the parameters are populated with the specific information and requirements specified in the user request. For example, the user may say "Make me a dinner reservation at a sushi place at 7." In this case, natural language processing module **732** can be able to correctly identify the actionable intent to be "restaurant reservation" based on the user input. According to the ontology, a structured query for a "restaurant reservation" domain may include parameters such as {Cuisine}, {Time}, {Date}, {Party Size}, and the like. In some examples, based on the speech input and the text derived from the speech input using STT processing module **730**, natural language processing module **732** can generate a partial structured query for the restaurant reservation domain, where the partial structured query includes the parameters {Cuisine="Sushi"} and {Time="7 pm"}. However, in this example, the user's utterance contains insufficient information to complete the structured query associated with the domain. Therefore, other necessary parameters such as {Party Size} and {Date} may not be specified in the structured query based on the information currently available. In some examples, natural language processing module **732** can populate some parameters of the structured query with received contextual information. For example, in some examples, if the user requested a sushi restaurant "near me," natural language processing module **732** can populate a {location} parameter in the structured query with GPS coordinates from the user device.

[0228] In some examples, natural language processing module **732** can pass the generated structured query (including any completed parameters) to task flow processing module **736** ("task flow processor"). Task flow processing module **736** can be configured to receive the structured query from natural language processing module **732**, complete the structured query, if necessary, and perform the actions required to "complete" the user's ultimate request. In some examples, the various procedures necessary to complete these tasks can be provided in task flow models **754**. In some examples, task flow models **754** can include procedures for obtaining additional information from the user and task flows for performing actions associated with the actionable intent.

[0229] As described above, in order to complete a structured query, task flow processing module **736** may need to initiate additional dialogue with the user in order to obtain additional information, and/or disambiguate potentially ambiguous utterances. When such interactions are necessary, task flow processing module **736** can invoke dialogue flow processing module **734** to engage in a dialogue with the user. In some examples, dialogue flow processing module **734** can determine how (and/or when) to ask the user for the additional information and receives and processes the user responses. The questions can be provided to and answers can be received from the users through I/O processing module **728**. In some examples, dialogue flow processing module **734** can present dialogue output to the user via audio and/or visual output, and receives input from the user via spoken or physical (e.g., clicking) responses. Continuing with the example above, when task flow processing module **736** invokes dialogue flow processing module **734** to determine the "party size" and "date" information for the structured query associated with the domain "restaurant reservation," dialogue flow processing module **734** can generate questions such as "For how many people?" and "On which day?" to pass to the user. Once

answers are received from the user, dialogue flow processing module **734** can then populate the structured query with the missing information, or pass the information to task flow processing module **736** to complete the missing information from the structured query.

[0230] Once task flow processing module **736** has completed the structured query for an actionable intent, task flow processing module **736** can proceed to perform the ultimate task associated with the actionable intent. Accordingly, task flow processing module **736** can execute the steps and instructions in the task flow model according to the specific parameters contained in the structured query. For example, the task flow model for the actionable intent of "restaurant reservation" can include steps and instructions for contacting a restaurant and actually requesting a reservation for a particular party size at a particular time. For example, using a structured query such as: {restaurant reservation, restaurant=ABC Café, date=3/12/2012, time=7 pm, party size=5}, task flow processing module **736** can perform the steps of: (1) logging onto a server of the ABC Café or a restaurant reservation system such as OPENTABLE®, (2) entering the date, time, and party size information in a form on the website, (3) submitting the form, and (4) making a calendar entry for the reservation in the user's calendar.

[0231] In some examples, task flow processing module **736** can employ the assistance of service processing module **738** ("service processing module") to complete a task requested in the user input or to provide an informational answer requested in the user input. For example, service processing module **738** can act on behalf of task flow processing module **736** to make a phone call, set a calendar entry, invoke a map search, invoke or interact with other user applications installed on the user device, and invoke or interact with third-party services (e.g., a restaurant reservation portal, a social networking website, a banking portal, etc.). In some examples, the protocols and application programming interfaces (API) required by each service can be specified by a respective service model among service models **756**. Service processing module **738** can access the appropriate service model for a service and generate requests for the service in accordance with the protocols and APIs required by the service according to the service model.

[0232] For example, if a restaurant has enabled an online reservation service, the restaurant can submit a service model specifying the necessary parameters for making a reservation and the APIs for communicating the values of the necessary parameter to the online reservation service. When requested by task flow processing module **736**, service processing module **738** can establish a network connection with the online reservation service using the web address stored in the service model, and send the necessary parameters of the reservation (e.g., time, date, party size) to the online reservation interface in a format according to the API of the online reservation service.

[0233] In some examples, natural language processing module **732**, dialogue flow processing module **734**, and task flow processing module **736** can be used collectively and iteratively to infer and define the user's intent, obtain information to further clarify and refine the user intent, and finally generate a response (i.e., an output to the user, or the completion of a task) to fulfill the user's intent. The generated response can be a dialogue response to the speech input that at least partially fulfills the user's intent. Further, in some examples, the generated response can be output as a speech output. In these examples, the generated response can be sent to speech synthesis module **740** (e.g., speech synthesizer) where it can be processed to synthesize the dialogue response in speech form. In yet other examples, the generated response can be data content relevant to satisfying a user request in the speech input.

[0234] Speech synthesis module **740** can be configured to synthesize speech outputs for presentation to the user. Speech synthesis module **740** synthesizes speech outputs based on text provided by the digital assistant. For example, the generated dialogue response can be in the form of a text string. Speech synthesis module **740** can convert the text string to an audible speech output. Speech synthesis module **740** can use any appropriate speech synthesis technique in order to generate speech outputs from text, including, but not limited, to concatenative synthesis, unit selection synthesis, diphone synthesis, domain-specific synthesis, formant synthesis, articulatory synthesis, hidden Markov model (HMM) based synthesis, and sinewave synthesis. In some examples, speech synthesis module **740** can be configured to synthesize individual words based on phonemic strings corresponding to the words. For example, a phonemic string can be associated with a word in the generated dialogue response. The phonemic string can be stored in metadata associated with the word. Speech synthesis model **740** can be configured to directly process the phonemic string in the metadata to synthesize the word in speech form.

[0235] In some examples, instead of (or in addition to) using speech synthesis module **740**, speech synthesis can be performed on a remote device (e.g., the server system **108**), and the synthesized speech can be sent to the user device for output to the user. For example, this can occur in some implementations where outputs for a digital assistant are generated at a server system. And because server systems generally have more processing power or resources than a user device, it can be possible to obtain higher quality speech outputs than would be practical with client-side synthesis.

[0236] Additional details on digital assistants can be found in the U.S. Utility application Ser. No. 12/987,982, entitled "Intelligent Automated Assistant," filed Jan. 10, 2011, and U.S. Utility application Ser. No. 13/251,088, entitled "Generating and Processing Task Items That Represent Tasks to Perform," filed Sep. 30, 2011, the entire disclosures of which are incorporated herein by reference.

[0237] 4. Process for Structured Dictation Using Digital Assistants

[0238] FIGS. **8**A-D illustrate process **800** for operating a digital assistant according to various examples. More specifically, process **800** can be implemented to perform structured dictation using a digital assistant. Process **800** can be performed using one or more electronic devices implementing a digital assistant. In some examples, process **800** can be performed using a client-server system (e.g., system **100**) implementing a digital assistant. In particular, as shown in FIGS. **8**A-D, blocks **801-804**, **828-840**, **844**, and **846** of process **800** can be performed by a client device (e.g., devices **104**, **200**, **400**, or **600** of FIGS. **1**, **2**A, **4**, **6**A-B, respectively), and blocks **818-826** and **842** of process **800** can be performed by a server system (e.g., DA server **106** described above with reference to FIG. **1**). It should be recognized however that in other examples, the individual blocks of process **800** can be distributed in any appropriate manner among one or more computers, systems, or electronic devices. For instances, in some examples, process **800** can be performed entirely on an elec-

tronic device (e.g., devices **104**, **200**, **400**, or **600**.) Further, while the following discussion describes process **800** as being performed by a digital assistant system (e.g., system **100** and/or digital assistant system **700**), it should be recognized that the process or any particular part of the process is not limited to performance by any particular device, combination of devices, or implementation.

[0239] At block **801**, a speech input can be received at an electronic device (e.g., device **104**, **200**, **400**, or **600**). In particular, the speech input can be received from a user of the electronic device. The speech input can be received in the course of, or as part of, an interaction with a digital assistant implemented on the electronic device. In some examples, the speech input can be received via a microphone (e.g., microphone **213**) of the electronic device. The received speech input can be processed (e.g., using audio circuitry **210** or processor(s) **220**) and converted into a representative form such as, for example, an electronic signal (digital or analog) or one or more audio files.

[0240] The speech input can be associated with an application (e.g., applications **236**) on the electronic device. In particular, the speech input can be directed to an input field of the application, such as, for example, the recipient field of a messages/email application or the search field of a maps application. For example, process **800** can include displaying an application on the display system (e.g., touch sensitive display system **212**) of the electronic device. The displayed application can include a graphical user interface having one or more text input fields. A selection of the one or more input fields can be received on the electronic device. While the input field is selected, instructions for invoking the digital assistant can be received. The speech input can be received while the digital assistant is invoked and while the one or more input fields are selected. Thus, in this example, the speech input is directed to the one or more selected input fields of the application.

[0241] The speech input can represent a user request. In some examples, the speech input can contain the user request. Further, in some examples, the speech input may not explicitly contain a user request, but rather the user request can be implied based on the context in which the speech input is received. In particular, the user request can be implied based on the input field and/or application to which the speech input is directed. For example, while focus is currently directed to the recipient field of the messaging or email application the speech input can include the word "mom". In this example, the speech input can represent a user request to obtain contact information of the user's mother. In another example, while focus is currently directed to the search field of the maps application the speech input can include the word "restaurant". In this example, the speech input can represent a user request to obtain location information of restaurants in a particular geographic area.

[0242] The speech input can be in natural language form and can include one or more ambiguous terms. For instance, the term "mom" can be an ambiguous term because it can be interpreted as a request to search for the contact information of a person whose name contains the string "mom" or as a request to search for the contact information of a person identified as being the mother of someone. Further, it may be ambiguous as to whose mother is being referenced in the speech input. In another example, the speech input can include the phrase "Best restaurants in San Francisco." In this example, the term "best" can be ambiguous because it can be interpreted as a request to search for restaurants with the term "best" in the search string or as a request to search for restaurants having a user rating that exceeds a certain threshold.

[0243] It should be recognized that in some examples, text input can be received at block **801** instead of speech input. For example, text input representing a user request can be received from the user via a suitable user interface on the electronic device. Examples of suitable user interfaces for receiving text input can include a virtual keyboard, a handwriting recognition interface, a gesture recognition interface, and the like. In these examples, the language input can be transmitted at block **804** and received at block **806** in lieu of the speech input. Further, in these examples, block **808** can be omitted since the input is already in text form.

[0244] At block **802**, metadata associated with the speech input can be generated. The metadata can include information relevant to processing the speech input. For example, the metadata can indicate whether natural language processing is to be performed on the speech input. Further, if natural language processing is to be performed, the metadata can be generated to identify one or more domains of an ontology that correspond to the user request. The one or more identified domains can subsequently be used (e.g., at block **818**) to perform natural language processing on the speech input. The information contained in the metadata can be in the form of data files, attributes, or tags.

[0245] The metadata can be generated based on the context in which the speech input is received. In particular, the metadata can be generated based on the application and/or input field associated with the speech input. In some examples, one or more input fields of an application displayed on the electronic device can be selected by the user to cause the focus of the application to be within the one or more input fields. The speech input can then be received while the focus of the application is within the one or more input fields. In these examples, it can be determined that the speech input is directed to the one or more input fields of the application and thus the metadata can be generated to indicate that natural language processing is to be performed on the speech input. Further, the metadata can be generated to identify one or more domains corresponding to the one or more input fields of the application. In this way, natural language processing (e.g., at block **818**) can be constrained to the one or more domains, which enables natural language processing to be performed more accurately and efficiently.

[0246] In a specific example, while focus is currently within the recipient field of a messages/email application displayed on the electronic device, speech input can be received. In this example, the metadata can be generated to indicate that natural language processing is to be performed on the speech input. Further, the metadata can be generated to identify a domain corresponding to contact information (e.g., information from the user's address book) as the focus is currently within a recipient field. Based on the metadata, natural language processing can subsequently be performed (e.g., at block **818**) using the domain corresponding to contact information.

[0247] In another example, while focus is currently within the search field of a maps application displayed on the electronic device, speech input of "best restaurants" can be received. The user can be browsing a particular geographic area on the maps application at the time the speech input is received. In this example, the metadata can be generated to indicate that natural language processing is to be performed

on the speech input. In addition, the metadata can be generated to identify a domain corresponding to location information. Further, in this example, the metadata can be generated to define one or more parameters associated with the identified domain. Specifically, the metadata can define the particular geographic area that the user was browsing. Based on the metadata, natural language processing can subsequently be performed (e.g., at block **818**) using the domain corresponding to location information. The particular geographic area defined in the metadata can also be used during natural language processing to generate the structured query (e.g., at block **818**) associated with the domain.

[0248] In some examples, the speech input can be directed to an input field of an application and it can be determined based on the input field that natural language processing need not be performed on the speech input. In one such example, the speech input can be directed to an input field corresponding to the body of an email message in the email application. In this example, it can be inferred based on the input field that the user intends to perform unstructured dictation to compose the body of the email message. In this example, the metadata can be generated to indicate that natural language process is not to be performed on the speech input. Instead, the metadata can be generated to indicate that the speech input is to be processed only to obtain a transcription of the speech input (e.g., speech-to-text processing).

[0249] Further, in some examples, the speech input can be received while the focus of the application is not within any particular input field of an application. In particular, the focus of the application displayed on the electronic device can be outside the one or more input fields while the speech input is received. Alternatively, the speech input may not be directed to any particular application or any particular input field of an application. In these examples, the metadata can be generated to indicate that natural language processing is to be performed on the speech input. Further, the metadata can be generated such that no particular domain corresponding to the user request is identified. In these examples, as described below in blocks **812** and **816**, natural language processing can be performed across a plurality of domains. Specifically, natural language processing can be performed (e.g., at block **816**) on the speech input to determine, among the plurality of domains, one or more domains implicated by the speech input. The one or more domains can subsequently be used (e.g., at block **818**) to generate a structured query that defines the user's intent.

[0250] In some examples, the metadata may not explicitly indicate whether to perform natural language processing on the speech input. Rather, in these examples, the metadata can be generated to identify the application and/or input field to which the speech input is directed. The natural language processor can subsequently determine (e.g., using look-up tables) whether to perform natural language processing based on the application and/or input field identified in the metadata. Similarly, the natural language processor can also identify one or more domains corresponding to the user request based on the application and/or input field identified in the metadata.

[0251] At block **804**, the speech input of block **801** and metadata of block **802** can be transmitted to a server (e.g., DA server **106**). The speech input and the metadata can be transmitted to the server using communication circuitry (e.g., RF Circuitry **208**) of the electronic device. The speech input and the metadata can be transmitted in any suitable form such an electronic signal or one or more data files. In some examples, the speech input and the metadata can be transmitted to the server via a direct communication connection, such as Bluetooth, NFC, BTLE, or the like, or via a wired or wireless network, such as a local Wi-Fi network. In one such example, the electronic device can establish a Wi-Fi connection to a communication network via a wireless router and transmit the speech input and the metadata to the server directly through a communication network (e.g., network(s) **110**). In other examples, the electronic device can utilize a second electronic device (e.g., user device **122**) as a proxy to transmit the speech input and the metadata to the server. In particular, the electronic device can establish a connection to the second electronic device and the second electronic device can be connected to the server via a communication network. The electronic device can thus transmit the representation of the speech input to the server via the second electronic device. A more detailed explanation of transmitting and receiving information to/from the server via the second electronic device is provided below.

[0252] At block **806**, the speech input and the metadata associated with the speech input can be received by the server from the electronic device. For example, with reference to FIG. **1**, the speech input and the metadata can be received at the I/O interface (e.g., I/O interface to DA client **112**) of DA server **106** from user device **104**.

[0253] At block **808**, a text string corresponding to the speech input can be determined. The speech input can be converted to text using conventional automatic speech recognition techniques. In particular, a STT processing module (e.g., STT processing module **730**) of the server can be used to process the speech input and determine the corresponding text string. The STT processing module can include one or more ASR systems (e.g., ASR system(s) **731**). The one or more ASR systems can extract representative features from the speech input and the extracted representative features can be processed using one or more speech recognition models and one or more speech recognition engines of the one or more ASR systems to produce intermediate recognitions results (e.g., phonemes, phonemic strings, and sub-words), and ultimately, text recognition results (e.g., words, word strings, or sequence of tokens). The text recognition results can include the text string corresponding to the speech input. The text string can be in the form of a sequence of words or a representation of a sequence of words (e.g., a sequence of tokens with each token representing a particular word).

[0254] At block **810**, it can be determined, based on the metadata, whether to perform natural language processing on the text string. Block **810** can be performed by the natural language processing module (e.g., natural language processing module **732**) of the server. As previously described, in some examples, the metadata can indicate (e.g., using one or more tags or attributes) whether to perform natural language processing on the text string. Based on such an indication, the server can determine whether to perform natural language processing on the text string. In some examples, the metadata can explicitly indicate whether to perform natural language processing on the text string. Specifically, the metadata can include an explicit natural language processing tag. The tag can indicate "yes" when natural language processing is to be performed on the text string and "no" when natural language processing is not to be performed on the text string. In other examples, the metadata can include other information that can be used to determine whether to perform natural language

processing. In particular, the metadata can include a tag or attribute that indicates the input field and/or application to which the speech input is directed. Based on the input field and/or application, the server can determine whether natural language processing is to be performed on the text string.

[0255] As shown in FIGS. **8**A-C, in response to determining based on the metadata that natural language processing is to be performed on the text string, blocks **812** through **840** can be performed.

[0256] At block **812**, it can be determined whether the metadata identifies one or more domains corresponding to the user request. The one or more domains can be one or more domains of an ontology (e.g., ontology **760**). The one or more domains can be identified by one or more tags or attributes in the metadata. In some examples, the one or more tags or attributes can explicitly identify one or more domains corresponding to the user request. In other examples, the one or more tags or attributes can implicitly identify one or more domains corresponding to the user request. Specifically, the one or more tags or attributes can identify the input field and/or application to which the speech input is directed. Based on the identified input field and/or application in the metadata, one or more domains corresponding to the user request can be identified (e.g., using a look-up table) at block **812**. In yet other examples, the one or more tags or attributes can define one or more topics (e.g., contacts, locations, movies, etc.) corresponding to the user input. The one or more topics can be based on the input field and/or application to which the speech input is directed. One or more domains associated with the user request can then be identified (e.g., using a look-up table) at block **812** based on the one or more topics defined in the metadata. Block **812** can be performed by the natural language processing module (e.g., natural language processing module **732**) of the server.

[0257] Block **814** can be performed in response to determining at block **812** that the metadata identifies one or more domains corresponding to the user request. At block **814**, domains other than the one or more domains identified at block **812** can be disabled. Block **812** can be performed by the natural language processing module (e.g., natural language processing module **732**) of the server. An ontology (e.g., ontology **760**) of the natural language processing module can include a plurality of domains that include the one or more domains identified at block **812** and other domains different from the one or more domains. These other domains can be disabled such that they do not participate in natural language processing of the text string. In this way, natural language processing of the text string can be constrained to only the one or more domains. The one or more domains identified at block **812** can be the most relevant domains of the plurality of domains for processing the speech input. Therefore, by constraining natural language processing to only the one or more domains, natural language processing can be performed more quickly, efficiently, and accurately.

[0258] Block **816** can be performed in response to determining at block **812** that the metadata does not identify one or more domains corresponding to the user request. At block **816**, one or more domains relevant to a user request can be determined. For example, the natural language processing module (e.g., natural language processing module **732**) of the server can process the text string to determine an actionable intent associated with the user request. In determining an actionable intent of the user, one or more domains relevant to the user request can be determined from the plurality of domains in the ontology.

[0259] The determination of the one or more domains relevant to the user request can be made based on the words and/or phrases in the text string. As described above, the ontology of the natural language processing module can include a plurality of domains. Each domain can be a domain of knowledge associated with a particular topic, concept, or field. Further, each domain can be associated with a set of words and/or phrases (e.g., stored in vocabulary index **744**) and with an actionable intent. For example, the plurality of domains can include a "contact information" domain that can be associated with a set a words and/or phrases that include "name", "nickname", "telephone number," "email," "contact," "address," and the like. Further, the "contact information" domain can be associated with the actionable intent of retrieving information related to the user's contacts. The natural language processing module can analyze the words and phrases contained in the text string and identify the one or more domains implicated by these words and phrases. For example, the relevant domain can be identified based on the number of words or phrases in the text string that are associated with the one or more domains relevant to the user request and based on the salience of those words or phrases with respect to the one or more domains relevant to the user request.

[0260] At block **818**, a structured query based on the one or more domains can be generated. In particular, a structured query representing an actionable intent associated with the one or more domains can be generated using the text string and based on the one or more domains. Generating the structured query can be part of natural language processing of the text string. In some examples, natural language processing can be constrained to the one or more domains such that the one or more domains are the only domains of the ontology used to generate the structured query. Thus, in these examples, the other domains in the ontology are not used to generate the structured query. Block **818** can be performed by the natural language processing module (e.g., natural language processing module **732**) of the server.

[0261] Generating the structured query can include parsing the text string to identify information that is relevant to the actionable intent. In one example, the one or more domains can include the domain corresponding to "location information" and the text string can include "best restaurants in San Francisco." In this example, the text string can be parsed to identify the location of "San Francisco" that is relevant to the actionable intent of finding location information.

[0262] One or more parameters of the structured query can be populated at block **818** with the relevant information identified from parsing the text string. The one or more parameters can be associated with the one or more domains. For example, the parameter of {search region} associated with the "location information" domain can be populated with the parsed information of "San Francisco" from the text string. Further, as described above, one or more parameters of the structured query can be populated with information from the metadata received at block **806**. For example, the metadata can define the geographic area viewed by the user on the maps application at the time the speech input is received. The parameter of {search region} can then be populated with the geographic area defined in the metadata.

[0263] Further, in some examples, generating the structured query at block **818** can include deriving a programmatic representation of the user request based on the one or more parameters. The programmatic representation can indicate the action to be performed and can specify the relevant parameters defining the action. For instance, in an example where the text string includes "best restaurants in San Francisco," the programmatic representation can define a search query for the locations of restaurants within the search region of San Francisco and having a user rating greater than four stars.

[0264] At block **820**, a task flow associated with the structured query can be executed. For example, a task flow processing module (e.g., task flow processing module **736**) of the server can process the structured query (generated at block **818**) to determine a task flow associated with the structured query. The task flow can be determined based on one or more task flow models (e.g., task flow models **754**) associated with the actionable intent and based on the one or more parameters of the structured query. Further, the task flow can be determined based on the programmatic representation of the user request derived at block **818**. The task flow can include instructions for performing one or more tasks in furtherance of satisfying the user request.

[0265] In some examples, executing the task flow can include obtaining a result to satisfy the user request. The result can be obtained from within the server or from sources and services external to the server (e.g., external services **118**, etc.). In a specific example, executing the task flow can include searching a contact database for contact information that satisfies the search constraints defined in the structured query. The contact database can include contact information from the address book of the user on the electronic device. The contact database can be stored on the server where it is searched. Alternatively, the contact database can be stored on the electronic device and uploaded to the server to perform the search. In some cases, the contact databased can be stored at an external service and the search can be performed by the external service. The structured query can define the search for contact information of a specific individual. The contact information of that specific individual can thus be retrieved as a result of searching the contact database.

[0266] In another example, executing the task flow can include searching a location database in accordance with search constraints defined in the structured query. In particular, the structured query can define a search for location information of a restaurant in San Francisco having a user rating greater than four stars. The location information of such a restaurant (or restaurants) can thus be retrieved as a result of searching the location database.

[0267] At block **822**, it can be determined whether a result satisfying the user request is obtained from executing the task flow. In some examples, the determination can be made based on whether the task flow is successfully executed to completion. In particular, if the task flow is successfully executed to obtain a result, it can be determined that a result satisfying the user request is obtained. However, if the task flow is not successfully executed (e.g., an error message is obtained), it can be determined that a result satisfying the user request is not obtained. In other examples, the result obtained from executing the task flow can be evaluated according to a predefined criterion. It can be determined that the result satisfies the user request if the predefined criterion is met. Conversely, it can be determined that the result does not satisfy the user request if the predefined criterion is not met.

[0268] As shown in FIG. **8**B, in response to determining at block **822** that a result satisfying the user request is obtained, block **824** can be performed. At block **824**, data content containing the result can be outputted. The result in the data content can be the result obtained from executing the task flow at block **820**. The data content can be outputted to the electronic device via a communication network using the I/O interface (I/O interface to DA client **112**) of the server.

[0269] Block **826** can be performed in response to determining at block **822** that a result satisfying the user request is not obtained. At block **826**, data content containing the text string can be outputted. The text string can then be subsequently processed (e.g., by the electronic device at block **834**, described below) to satisfy the user request. The data content can be outputted to the electronic device via a communication network using the I/O interface (I/O interface to DA client **112**) of the server.

[0270] At block **828**, data content in furtherance of satisfying the user request can be received from the server. As described above, in response to determining at block **822** that a result satisfying the user request is obtained, the receive data content can contain the result obtained from executing the task flow at block **820**. Alternatively, in response to determining at block **822** that a result satisfying the user request is not obtained, the receive data content can contain the text string of block **806**. The data content can be received from the server using the communication circuitry (e.g., RF circuitry **208**) of the electronic device.

[0271] At block **830**, it can be determined whether the data content satisfies the user request. Block **830** can be similar to block **822**, except that block **830** is performed by the electronic device rather than the server. In some examples, the determination can be based on analyzing the data content for a result that satisfies the user request. In other examples, the determination can be based on an attribute or tag in the data content. In particular, if it is determined at block **822** that a result satisfying the user request is obtained, then the data content can include an attribute or tag indicating this determination. It can then be determined at block **830** that the received data content satisfies the user request based on this attribute or tag. However, if it is determined at block **822** that a result satisfying the user request is not obtained, then the data content can include an attribute or tag indicating this determination. It can then be determined at block **830** that the received data content does not satisfy the user request based on this attribute or tag.

[0272] Block **832** can be performed in response to determining at block **830** that the data content satisfies the user request. At block **832**, a final result based on the data content can be outputted on the electronic device. The final result can be based on the result obtained from executing the task flow at block **820**. In some examples, the final result can be outputted via any suitable interface (e.g., touch-sensitive display system **212**, speaker **211**, or tactile output generator **267**) of the electronic device. In particular, the final result can be outputted as a visual, sound, or tactile output from the electronic device in accordance with various attributes contained in the data content.

[0273] In some examples, the result obtained from executing the task flow at block **820** can be directly outputted on the electronic device as the final result. In other examples, the result obtained from executing the task flow at block **820** can be further processed to obtain the final result. The data content can include instructions to further process the result to obtain

the final result. Further, the data content can include formatting data, graphical user interface data, audio data, and the like that can be used to generate and output the final result.

[0274] In a specific example, the result obtained from executing the task flow at block **820** can be a reference identifier that identifies a particular contact in the address book of the user on the electronic device. The electronic device can use this reference identifier to retrieve, from the address book, contact information (e.g., the email address or phone number) of the individual corresponding to the reference identifier. The contact information can then be outputted as the final result on the electronic device. In particular, the final result can be displayed in the recipient field of the messages/email application as a name, email address, phone number or equivalent contact means.

[0275] In another example, the result obtained from executing the task flow at block **820** can be location information (e.g., Global Positioning System coordinates) corresponding to one or more highly rated restaurants in San Francisco. The electronic device can process this location information such that the information is displayed on a map. In this example, outputting the final result can include displaying a map of San Francisco that indicates the location(s) of the one or more highly rated restaurants.

[0276] In some examples, outputting the final result can require a selection from the user. In one such example, the speech input can represent a request for the contact information of a person named "Brian." The results obtained from searching a contact database on the user (e.g., at block **820**) can include the contact information of several individuals named "Brian." In this example, the contact information of these individuals named "Brian" can initially be displayed on the electronic device. The user can then make a selection from among the displayed contact information. The selection can be received from the user via a graphic user interface of the electronic device. The final result can be based on the selected contact information. In particular, the selected contact information of "Brian" can be displayed in the recipient field of the messages/email application. Accordingly, the final result can at least partially disambiguate one or more ambiguous terms in the speech input.

[0277] With reference to FIG. **8**C, blocks **834** through **840** can be performed in response to determining at block **830** that the data content does not satisfy the user request. At block **834**, a task can be performed using the text string to obtain a second result. In particular, because the server did not obtain a result that satisfies the user request, the text string can be further processed at block **834** by the electronic device in furtherance of satisfying the user request.

[0278] In some examples, the task can include searching a database on the electronic device using the text string. For example, the speech input can represent a user request to obtain the contact information of an individual named "Brian." The server may be unsuccessful in finding any contact information corresponding to "Brian" and thus, the text string "Brian" is returned to the electronic device (e.g., at block **826**). In this example, the electronic device can search the address book on the electronic device for "Brian" and can successfully retrieve the contact information of "Brian."

[0279] In another example, the task can include performing, based on the text string, a search of data previously entered, received, or presented on the electronic device using the application. For example, the electronic device can search previous messages of email for contact information related to "Brian." Such a search can be effective in examples where the user has utilized an individual's contact information, but did not store that individual's contact information in the address book.

[0280] At block **836**, it can be determined whether the second result satisfies the user request. In some examples, it can be determined that the second result satisfies the user request if the task performed at block **834** is successfully completed and a second result is obtained. Alternatively, it can be determined that the second result does not satisfy the user request if the task performed at block **834** is not successfully completed (e.g., an error message is obtained) or a second result is not obtained. In other examples, the second result can be evaluated based on a predefined criterion to determine whether the user request is satisfied. In these examples, it can be determined that the second result satisfies the user request if the predefined criterion is satisfied. Conversely, it can be determined that the second result does not satisfy the user request if the predefined criterion is not satisfied.

[0281] As shown in FIG. **8**C, block **838** can be performed in response to determining at block **836** that the second result satisfies the user request. At block **838**, the second result can be outputted on the electronic device. The second result can be outputted in a similar manner as the final result described at block **832**. For example, the second result can be displayed on the electronic device. In particular, the second result can be displayed in one or more input fields of an application.

[0282] Block **840** can be performed in response to determining at block **836** that the second result does not satisfy the user request. At block **840**, the text string can be outputted on the electronic device. For example, the text string can be displayed on the electronic device. In particular, the text string can be displayed in one or more input fields of an application.

[0283] With reference to FIG. **8**D, in response to determining at block **810** that natural language processing is not to be performed on the text string, blocks **842** through **846** can be performed. At block **842**, the text string determined at block **808** can be outputted from the server. The text string can be outputted from the server to the electronic device via a communication network using the I/O interface (I/O interface to DA client **112**) of the server.

[0284] At block **844**, the text string can be received at the electronic device. The text string can be received from the server using the communication circuitry (e.g., RF circuitry **208**) of the electronic device.

[0285] At block **846**, the text string can be outputted from the electronic device. For example, the text string can be displayed on the electronic device. In particular, the text string can be displayed in one or more input fields of an application.

[0286] Although blocks **801** through **846** of process **800** are shown in a particular order in FIGS. **8**A-D, it should be appreciated that these blocks can be performed in any order. Further, it should be appreciated that in some cases, one or more blocks of processes **800** can be optional and additional blocks can also be performed.

[0287] Further, in some examples, at least some of the communication between the electronic device and the server in process **800** can occur via a second electronic device (e.g., user device **122**). The second electronic device can serve as a proxy for the electronic device. In some examples, prior to the second electronic device serving as a proxy for the electronic

device, the electronic device and the second electronic devices can be paired (e.g., Bluetooth pairing) to allow the devices to recognize each other and securely communicate with one another.

[0288] In some examples, the electronic device can determine whether a second electronic device is available to act as a proxy. If the second electronic device is available to act as a proxy, communication between the electronic device and the server can occur via the second electronic device. However, if the second electronic device is not available to act as a proxy, communication can occur directly between the electronic device and the server. For example, process **800** can include determining whether the electronic device is communicatively coupled to a second electronic device. In some examples, such a determination can occur prior to transmitting information between the electronic device and the server. In a specific example, the determination can occur prior to block **804** of process **800**. In response to determining that the electronic device is communicatively coupled to the second electronic device, the transmitting (or outputting) and receiving at any one of blocks **804**, **806**, **824**, **826**, **828**, **842**, and **844** can occur via the second electronic device. Conversely, in response to determining that the electronic device is not communicatively coupled to the second electronic device, the transmitting and receiving at blocks **804**, **806**, **824**, **826**, **828**, **842**, and **844** can occur directly between the electronic device and the server.

5. Additional Exemplary Electronic Devices

[0289] FIG. **9** shows a functional block diagram of electronic device **900** configured in accordance with the principles of the various described examples. The functional blocks of the device can be optionally implemented by hardware, software, or a combination of hardware and software to carry out the principles of the various described examples. It is understood by persons of skill in the art that the functional blocks described in FIG. **9** can be optionally combined or separated into sub-blocks to implement the principles of the various described examples. Therefore, the description herein optionally supports any possible combination, separation, or further definition of the functional blocks described herein.

[0290] As shown in FIG. **9**, electronic device **900** can include touch screen display unit **902** configured to display a graphical user interface, touch-sensitive surface unit **903** configured to receive contact input from the user, audio input unit **904** configured to receive speech input, speaker unit **906** configured to output audio, and RF unit **907** configured to transmit and receive data from external electronic devices. Electronic device **900** can further include processing unit **908** coupled to display unit **902**, touch-sensitive surface unit **903**, audio input unit **904**, speaker unit **906**, and RF unit **907**. In some examples, processing unit **908** can include receiving unit **910**, determining unit **912**, generating unit **914**, executing unit **916**, outputting unit **918**, transmitting unit **920**, displaying unit **922**, and performing unit **924**.

[0291] Processing unit **908** is configured to receive (e.g., with receiving unit **910**) via the audio input unit **904**, a speech input representing a user request. Processing unit **908** is further configured to receive (e.g., with receiving unit **910**) via RF unit **907**, metadata associated with the speech input. Processing unit **908** is further configured to determine (e.g., with determining unit **912**) a text string corresponding to the speech input. Processing unit **908** is further configured to determine (e.g., with determining unit **912**), based on the metadata, whether to perform natural language processing on the text string. In response to determining that natural language processing is to be performed on the text string, processing unit **908** is further configured to determine (e.g., with determining unit **912**) whether the metadata identifies one or more domains corresponding to the user request. In response to determining that the metadata identifies one or more domains corresponding to the user request, processing unit **908** is further configured to generate (e.g., with generating unit **914**), using the text string and based on the one or more domains, a structured query representing an actionable intent associated with the one or more domains. Processing unit **908** is further configured to execute (e.g., with executing unit **916**) a task flow associated with the structured query. Processing unit **908** is further configured to determine (e.g., with determining unit **912**) whether a result satisfying the user request is obtained from executing the task flow. In response to determining that a result satisfying the user request is obtained from executing the task flow, processing unit **908** is further configured to output (e.g., with outputting unit **918**) data content containing the result.

[0292] In some examples, the one or more domains are domains of an ontology and the ontology includes a plurality of other domains different from the one or more domains.

[0293] In some examples, in response to determining that the metadata identifies one or more domains associated with the user request, the plurality of other domains are disabled such that natural language processing is not performed on the text string using the plurality of other domains.

[0294] In some examples, in response to determining that the metadata identifies one or more domains associated with the user request, the structured query is not generated using the plurality of other domains.

[0295] In some examples, in response to determining that the metadata identifies one or more domains associated with the user request, the one or more domains are the only domains of the ontology used to generate the structured query.

[0296] In some examples, the speech input is associated with an input field of an application. In these examples, the one or more domains are identified by one or more attributes in the metadata. Further, in these examples, the one or more attributes are based on the input field and the application.

[0297] In some examples, whether natural language processing is to be performed on the text string is determined based on the one or more attributes.

[0298] In some examples, the one or more attributes define one or more topics corresponding to the input field of the application and the one or more domains associated with the user request are identified according to the one or more topics.

[0299] In some examples, the input field is a recipient field and the application is an electronic text-based communication application. The one or more attributes identify one or more domains associated with contact information. In these examples, executing the task flow includes searching a contact database in accordance with search constraints defined in the structured query. Further, the result identifies contact information to be populated into the recipient field.

[0300] In some examples, the speech input is received from a user and the speech input defines a recipient based on a relationship of the recipient to the user.

[0301] In some examples, the input field is a location search field and the application is a maps application. The one or more attributes identify one or more domains associated with

location information. In these examples, processing unit **908** is further configured to search (e.g., with executing unit **916**) a location database in accordance with search constraints defined in the structured query. Further, the result identifies location information to be presented using the maps application.

[0302] In some examples, the metadata defines a geographic area in which the searching of the location database is to be confined.

[0303] In some examples, in response to determining that natural language processing is not to be performed on the text string, processing unit **908** is further configured to output (e.g., with outputting unit **918**) the text string.

[0304] In some examples, in response to determining that the metadata does not identify one or more domains corresponding to the speech input, processing unit **908** is further configured to determine (e.g., with determining unit **912**), using the text string, a relevant domain associated with the speech input. Processing unit **908** is further configured to generate (e.g., with generating unit **914**), using the text string and based on the relevant domain, a second structured query representing an actionable intent associated with the relevant domain. Processing unit **908** is further configured to execute (e.g., with executing unit **916**) a second task flow associated with the second structured query to obtain a second result in furtherance of satisfying the user request. Processing unit **908** is further configured to output (e.g., with outputting unit **918**) second data content containing the second result.

[0305] In some examples, in response to determining that a result satisfying the user request is not obtained from executing the task flow, processing unit **908** is further configured to output (e.g., with outputting unit **918**) the text string.

[0306] In some examples, processing unit **908** is further configured to parse (e.g., with generating unit **914**) the text string based on the one or more domains to identify relevant information required for the actionable intent, to populate (e.g., with generating unit **914**) a parameter of the structured query with the relevant information, and to derive (e.g., with generating unit **914**) a programmatic representation of the user request based on the parameter. In these examples, the task flow is based on the programmatic representation of the user request.

[0307] In some examples, the metadata defines a second parameter of the structured query.

[0308] In some examples, the speech input includes one or more ambiguous terms and the result at least partially disambiguates the one or more ambiguous terms.

[0309] In some examples, the data content includes instructions for performing a task using the result.

[0310] In some examples, processing unit **908** is configured to receive (e.g., with receiving unit **910**) via audio input unit **904**, a speech input representing a user request. Processing unit **908** is further configured to transmit (e.g., with transmitting unit **920**), using RF unit **907**, metadata associated with the speech input to the server. The metadata identifies one or more domains of an ontology and the one or more domains correspond to the user request. Processing unit **908** is further configured to receive (e.g., with receiving unit **910**), via RF unit **907**, data content in furtherance of satisfying the user request. The data content is derived from the speech input. Processing unit **908** is further configured to determine (e.g., with determining unit **912**) whether the data content satisfies the user request. In response to determining that the data content satisfies the user request, processing unit **908** is further configured to output (e.g., with outputting unit **918**) a result based on the data content.

[0311] In some examples, the speech input is directed to an input field of an application on the electronic device.

[0312] In some examples, the one or more domains are based on the input field of the application.

[0313] In some examples, processing unit **908** is further configured to display (e.g., displaying unit **922**) on display unit **902** a graphical user interface of the application. The graphical user interface includes the input field. Processing unit **908** is further configured to receive (e.g., with receiving unit **910**) via touch-sensitive surface unit **903** a selection of the input field. Processing unit **908** is further configured to receive (e.g., with receiving unit **910**) via touch-sensitive surface unit **903**, instructions to invoke the digital assistant while the input field is selected, wherein the speech input is received while the digital assistant is invoked.

[0314] In some examples, the input field is a recipient field and the application is an electronic text-based communication application. In these examples, the one or more domains are associated with the user request of finding contact information.

[0315] In some examples, the speech input defines a recipient based on a relationship of the recipient to the user.

[0316] In some examples, the input field is a location search field and the application is a maps application. In these examples, the one or more domains are associated with the user request of finding location information.

[0317] In some examples, a map associated with the user request of finding location information is displayed on display unit **902** and the metadata defines a geographic area of the displayed map.

[0318] In some examples, processing unit **908** is further configured to display (e.g., with displaying unit **922**) on display unit **902** the result in the input field.

[0319] In some examples, the data content includes a plurality of values. In these examples, processing unit **908** is further configured to display (e.g., with displaying unit **922**) on display unit **902** the plurality of values on the electronic device. Further, processing unit **908** is further configured to receive (e.g., with receiving unit **910**) via touch screen display unit **902** a selection of a first value of the plurality of values. The result is based on the first value.

[0320] In some examples, the metadata indicates that natural language processing is to be performed on the speech input using the one or more domains.

[0321] In some examples, the metadata defines a parameter associated with the one or more domains.

[0322] In some examples, the ontology includes a plurality of other domains different from the one or more domains. The metadata indicates that the plurality of other domains are to be disabled such that natural language processing is not performed on the speech input using the plurality of other domains.

[0323] In some examples, the data content includes a text string representing the speech input. In response to determining that the data content does not satisfy the user request, processing unit **908** is further configured to perform (e.g., with performing unit **924**) a predefined task using the text string to obtain a second result. Processing unit **908** is further configured to determine (e.g., with determining unit **912**) whether the second result satisfies the user request. In response to determining that the second result satisfies the

user request, processing unit **908** is further configured to output (e.g., with outputting unit **918**) output the second result.

[0324] In some examples, processing unit **908** is further configured to perform (e.g., with performing unit **924**), based on the text string, a search of a database on the electronic device.

[0325] In some examples, processing unit **908** is further configured to perform (e.g., with performing unit **924**), based on the text string, a search of data previously entered, received, or presented on the electronic device using the application.

[0326] In some examples, in response to determining that the second result does not satisfy the user request, processing unit **908** is further configured to receive output (e.g., with outputting unit **918**) the text string.

[0327] In some examples, the data content is derived by performing natural language processing on the text string.

[0328] In some examples, the speech input includes one or more ambiguous terms and the result at least partially disambiguates the one or more ambiguous terms.

[0329] In some examples, the data content includes instructions for performing a task using the data content.

[0330] The operations described above with reference to FIGS. **8**A-D can be optionally implemented by components depicted in FIGS. **1**, **2**A, **4**, **6**A-B, **7**A, **9**, or **10**. For example, receiving operation **806**; determining operations **808**, **810**, **812**, **822**; generating operation **818**; executing operation **820**; and outputting operation **824** are, optionally implemented by I/O processing module **728**, STT processing module **730**, natural language processing module **732**, and task flow processing module **736**. I/O processing module **728** can received a speech input representing a user request and can receive metadata associated with the speech input. STT processing module **730** can determine a text string corresponding to the speech input. Natural language processing module **732** can determine, based on the metadata, whether to perform natural language processing on the text string. In response to determining that natural language processing is to be performed on the text string, natural language processing module **732** can further determine whether the metadata identifies one or more domains corresponding to the user request. In response to determining that the metadata identifies one or more domains corresponding to the user request, natural language processing module **732** can generate, using the text string and based on the one or more domains, a structured query representing an actionable intent associated with the one or more domains. Task flow processing module **736** can execute a task flow associated with the structured query. Task flow processing module **736** can determine whether a result satisfying the user request is obtained from executing the task flow. In response to determining that a result satisfying the user request is obtained from executing the task flow, I/O processing module **728** can output data content containing the result.

[0331] In another example, receiving operations **801**, **828**; transmitting operation **804**; determining operation **830**; outputting operation **832** are, optionally implemented by digital assistant client module **229** and applications **236**. Digital assistant client module **229** can receive from microphone **213** a speech input representing a user request. Digital assistant client module **229** can transmit, using RF circuitry **208**, the speech input and metadata associated with the speech input. The metadata can identify one or more domains of an ontology and the one or more domains correspond to the user request. Digital assistant client module can receive, using RF circuitry **208**, data content in furtherance of satisfying the user request. The data content can be derived from the speech input. Digital assistant client module **229** can determine whether the data content satisfies the user request. In response to determining that the data content satisfies the user request, Digital assistant client module **229** and applications **236** can output a result based on the data content.

[0332] FIG. **10** shows a functional block diagram of electronic device **1000** configured in accordance with the principles of the various described examples. The functional blocks of the device can be optionally implemented by hardware, software, or a combination of hardware and software to carry out the principles of the various described examples. It is understood by persons of skill in the art that the functional blocks described in FIG. **10** can be optionally combined or separated into sub-blocks to implement the principles of the various described examples. Therefore, the description herein optionally supports any possible combination, separation, or further definition of the functional blocks described herein.

[0333] As shown in FIG. **10**, electronic device **1000** can include display unit **1002** configured to display a user interface and receive input from the user, touch-sensitive surface unit **1003** configured to receive contact input from the user, audio input unit **1004** configured to receive speech input, speaker unit **1006** configured to output audio, and RF unit **1007** configured to transmit and receive data from external electronic devices. Electronic device **1000** can further include processing unit **1008** coupled to display unit **1002**, touch-sensitive surface unit **1003**, audio input unit **1004**, speaker unit **1006**, and RF unit **1007**. In some examples, processing unit **1008** can include displaying unit **1010**, receiving unit **1012**, determining unit **1014**, constraining unit **1016**, outputting unit **1018**, and performing unit **1020**.

[0334] Processing unit **1008** can be configured to display (e.g., with displaying unit **1010**) on display unit **1002**, an application comprising one or more text input fields. Processing unit **1008** can be further configured to receive (e.g., with receiving unit **1012**) via audio input unit **1004**, a speech input. Processing unit **1008** can be further configured to determine (e.g., with determining unit **1014**) a text string corresponding to the speech input. Processing unit **1008** can be further configured to determine (e.g., with determining unit **1014**) whether a focus of the application is within the one or more text input fields. In accordance with a determination that the focus of the application is within the one or more text input fields, processing unit **1008** can be further configured to constrain (e.g., with constraining unit **1016**) natural language processing of the text string to a domain of two or more domains. Processing unit **1008** can be further configured to output (e.g., with outputting unit **1018**) a result based on the domain. In accordance with a determination that the focus of the application is outside the one or more text input fields, processing unit **1008** can be further configured to perform (e.g., using performing unit **1020**) natural language processing of the text string across the two or more domains. Processing unit **1008** can be further configured to output (e.g., with outputting unit **1018**) on display unit **1002** or using speaker unit **1006**, a result based on the two or more domains.

[0335] In some examples, the application is an email application. In some examples, the text input field is a recipient field of the email application. In some examples, processing unit **1008** can be further configured to constrain (e.g., with constraining unit **1016**) the domain to contacts stored on the

electronic device. In some examples, the domain is associated with contact information stored on the electronic device.

[0336] In some examples, the application is a maps application. In some examples, the text input field is a search field of the maps application. In some examples, the domain is associated with location information.

[0337] The operations in the information processing methods described above are, optionally implemented by running one or more functional modules in information processing apparatus such as general purpose processors (e.g., as described above with respect to FIGS. **2**A, **4**, **6**, **7**A) or application specific chips.

[0338] The operations described above with reference to FIGS. **8**A-D can be optionally implemented by components depicted in FIGS. **1**, **2**A, **4**, **6**A-B, **7**A, **9**, or **10**. For example, displaying operation **801**; receiving operation **806**; determining operation **808**; constraining operation **814**; performing operation **816**; and outputting operation **832** are, optionally implemented by applications **724**, I/O processing module **728**, STT processing module **730**, natural language processing module **732**, and task flow processing module **736**. In particular, applications **724** can display an application comprising one or more text input fields. I/O processing module can receive the speech input and STT processing module **730** can convert the speech input to a text string. When the focus of the application is within the one or more text input fields, natural language processing module **732** can constrain natural language processing of the text string to a domain of two or more domains and task flow processing module **736** can output a result based on the domain. When the focus of the application is outside the one or more text input fields, natural language processing module **732** can perform natural language processing of the text string across the two or more domains and task flow processing module **736** can output a result based on the two or more domains. Similarly, it would be clear to a person having ordinary skill in the art how other processes can be implemented based on the components depicted in FIGS. **1**, **2**A, **4**, **6**A-B, and **7**A.

[0339] The foregoing description, for purpose of explanation, has been described with reference to specific embodiments. However, the illustrative discussions above are not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the techniques and their practical applications. Others skilled in the art are thereby enabled to best utilize the techniques and various embodiments with various modifications as are suited to the particular use contemplated.

[0340] Although the disclosure and examples have been fully described with reference to the accompanying drawings, it is to be noted that various changes and modifications will become apparent to those skilled in the art. Such changes and modifications are to be understood as being included within the scope of the disclosure and examples as defined by the claims.

What is claimed is:

1. A method for operating a digital assistant, the method comprising:

at an electronic device:

receiving a speech input representing a user request;

receiving metadata associated with the speech input;

determining a text string corresponding to the speech input;

determining, based on the metadata, whether to perform natural language processing on the text string;

in response to determining that natural language processing is to be performed on the text string:

determining whether the metadata identifies one or more domains corresponding to the user request;

in response to determining that the metadata identifies one or more domains corresponding to the user request:

generating, using the text string and based on the one or more domains, a structured query representing an actionable intent associated with the one or more domains;

executing a task flow associated with the structured query;

determining whether a result satisfying the user request is obtained from executing the task flow; and

in response to determining that a result satisfying the user request is obtained from executing the task flow, outputting data content containing the result.

2. The method of claim **1**, wherein the one or more domains are domains of an ontology, and wherein the ontology includes a plurality of other domains different from the one or more domains.

3. The method of claim **2**, wherein in response to determining that the metadata identifies one or more domains associated with the user request, the plurality of other domains are disabled such that natural language processing is not performed on the text string using the plurality of other domains.

4. The method of claim **2**, wherein in response to determining that the metadata identifies one or more domains associated with the user request, the structured query is not generated using the plurality of other domains.

5. The method of claim **2**, wherein in response to determining that the metadata identifies one or more domains associated with the user request, the one or more domains are the only domains of the ontology used to generate the structured query.

6. The method of claim **1**, wherein the speech input is associated with an input field of an application, wherein the one or more domains are identified by one or more attributes in the metadata, and wherein the one or more attributes are based on the input field and the application.

7. The method of claim **6**, wherein whether natural language processing is to be performed on the text string is determined based on the one or more attributes.

8. The method of claim **6**, wherein the one or more attributes define one or more topics corresponding to the input field of the application, and wherein the one or more domains associated with the user request are identified according to the one or more topics.

9. The method of claim **6**, wherein:

the input field is a recipient field;

the application is an electronic text-based communication application;

the one or more attributes identify one or more domains associated with contact information;

executing the task flow includes searching a contact database in accordance with search constraints defined in the structured query; and

the result identifies contact information to be populated into the recipient field.

10. The method of claim **9**, wherein the speech input is received from a user, and wherein the speech input defines a recipient based on a relationship of the recipient to the user.

11. The method of claim **6**, wherein:

the input field is a location search field;

the application is a maps application;

the one or more attributes identify one or more domains associated with location information;

executing the task flow includes searching a location database in accordance with search constraints defined in the structured query; and

the result identifies location information to be presented using the maps application.

12. The method of claim **11**, wherein the metadata defines a geographic area in which the searching of the location database is to be confined.

13. The method of claim **1**, further comprising:

in response to determining that natural language processing is not to be performed on the text string, outputting the text string.

14. The method of claim **1**, further comprising:

in response to determining that the metadata does not identify one or more domains corresponding to the speech input:

determining, using the text string, a relevant domain associated with the speech input;

generating, using the text string and based on the relevant domain, a second structured query representing an actionable intent associated with the relevant domain;

executing a second task flow associated with the second structured query to obtain a second result in furtherance of satisfying the user request; and

outputting second data content containing the second result.

15. The method of claim **1**, further comprising:

in response to determining that a result satisfying the user request is not obtained from executing the task flow, outputting the text string.

16. The method of claim **1**, wherein generating the structured query further comprises:

parsing the text string based on the one or more domains to identify relevant information required for the actionable intent;

populating a parameter of the structured query with the relevant information; and

deriving a programmatic representation of the user request based on the parameter, wherein the task flow is based on the programmatic representation of the user request.

17. The method of claim **1**, wherein the metadata defines a second parameter of the structured query.

18. The method of claim **1**, wherein the speech input includes one or more ambiguous terms, and wherein the result at least partially disambiguates the one or more ambiguous terms.

19. The method of claim **1**, wherein the data content includes instructions for performing a task using the result.

20. A method for operating a digital assistant, the method comprising:

at an electronic device with a display system and a microphone:

displaying, on the display system, an application comprising one or more text input fields;

receiving, via the microphone, a speech input;

determining a text string corresponding to the speech input;

determining whether a focus of the application is within the one or more text input fields;

in accordance with a determination that the focus of the application is within the one or more text input fields:

constraining natural language processing of the text string to a domain of two or more domains; and

outputting a result based on the domain; and

in accordance with a determination that the focus of the application is outside the one or more text input fields:

performing natural language processing of the text string across the two or more domains; and

outputting a result based on the two or more domains.

21. The method of claim **20**, wherein the application is an email application, and wherein the text input field is a recipient field of the email application.

22. The method of claim **21**, further comprising:

constraining the domain to contacts stored on the electronic device.

23. The method of claim **20**, wherein the application is a maps application.

24. A non-transitory computer-readable storage medium comprising computer-executable instructions, which when executed by one or more processors, causes the one or more processors to:

receive a speech input representing a user request;

receive metadata associated with the speech input;

determine a text string corresponding to the speech input;

determine, based on the metadata, whether to perform natural language processing on the text string;

in response to determining that natural language processing is to be performed on the text string:

determine whether the metadata identifies one or more domains corresponding to the user request;

in response to determining that the metadata identifies one or more domains corresponding to the user request:

generate, using the text string and based on the one or more domains, a structured query representing an actionable intent associated with the one or more domains;

execute a task flow associated with the structured query;

determine whether a result satisfying the user request is obtained from executing the task flow; and

in response to determining that a result satisfying the user request is obtained from executing the task flow, output data content containing the result.

25. A system comprising:

one or more processors;

memory storing computer-readable instructions, which when executed by the one or more processors, causes the one or more processors to:

receive a speech input representing a user request;

receive metadata associated with the speech input;

determine a text string corresponding to the speech input;

determine, based on the metadata, whether to perform natural language processing on the text string;

in response to determining that natural language processing is to be performed on the text string:

determine whether the metadata identifies one or more domains corresponding to the user request;

in response to determining that the metadata identifies one or more domains corresponding to the user request:

generate, using the text string and based on the one or more domains, a structured query representing an actionable intent associated with the one or more domains;

execute a task flow associated with the structured query;

determine whether a result satisfying the user request is obtained from executing the task flow; and

in response to determining that a result satisfying the user request is obtained from executing the task flow, output data content containing the result.

* * * * *