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

(19) World Intellectual Property Organization
International Bureau

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

WIPO | PCT

(10) International Publication Number
WO 2014/197340 A1

(51) International Patent Classification:
G06F 9/44 (2006.01)

(21) International Application Number:
PCT/US2014/040416

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

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/832,841 8 June 2013 (08.06.2013) US
14/291,688 30 May 2014 (30.05.2014) US

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

(72) Inventors: **MAGAHERN, Charles**; c/o Apple Inc., I Infinite Loop, Cupertino, CA 95014 (US). **HUANG, Szu-Wen**; c/o Apple Inc., I Infinite Loop, Cupertino, CA 95014 (US). **PAULSON, Nicholas, J.**; c/o Apple Inc., I Infinite Loop, Cupertino, CA 95014 (US). **ALSINA, Thomas, M.**; c/o Apple Inc., I Infinite Loop, MS 302-3APP, Cupertino, CA (US).

(74) Agents: **WILLIAMS, Gary, S.** et al.; Morgan Lewis & Bockius LLP, 2 Palo Alto Square, 3000 El Camino Real, Suite 700, Palo Alto, CA 94306 (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,

[Continued on next page]

(54) Title: DEVICE AND METHOD FOR GENERATING USER INTERFACES FROM A TEMPLATE

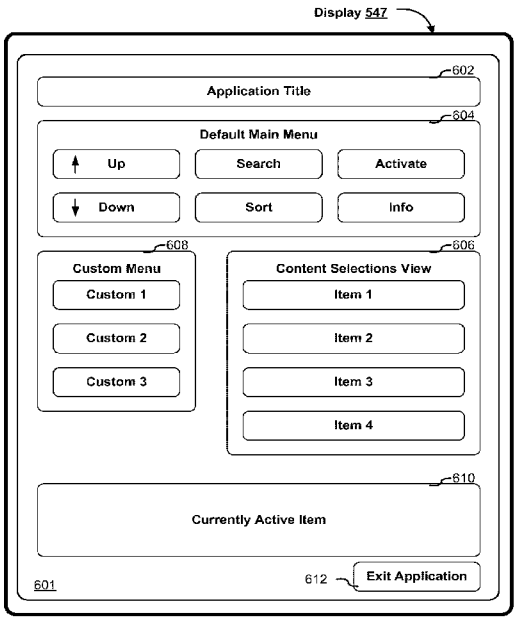


Figure 6A

(57) Abstract: An electronic device is configured to receive a first request to display a user interface of a first third-party application on a respective display that is in communication with the device. The device is further configured to, in response to receiving the request, obtain a first user-interface template configured to be used by a plurality of third-party applications, and request, from the first application, one or more values for populating the first template. The device is also configured to receive, from the first application, while the first application is running on the device, a first set of values for populating the first template; populate the first template with the first set of values; generate a first user interface for the first application using the first template populated with the first set of values; and send information to the respective display that enables the first user interface to be displayed.

WO 2014/197340 A1

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

DEVICE AND METHOD FOR GENERATING USER INTERFACES FROM A TEMPLATE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Patent Application No. 61/832,841, filed June 8, 2013, and is related to the following applications: U.S. Provisional Application Serial No. 61/793,924, filed March 15, 2013, entitled "Voice and Touch User Interface"; U.S. Application Serial No. 13/032,614, filed February 22, 2011, entitled "Pushing a Graphical User Interface to a Remote Device with Display Rules Provided by the Remote Device"; U.S. Application Serial No. 12/683,218, filed January 6, 2010, entitled "Pushing a User Interface to a Remote Device"; U.S. Application Serial No. 12/119,960, filed May 13, 2008, entitled "Pushing a User Interface to a Remote Device"; U.S. Application Serial No. 13/175,581, filed July 1, 2011, entitled "Pushing a User Interface to a Remote Device"; U.S. Application Serial No. 13/161,339, filed June 15, 2011, entitled "Pushing a Graphical User Interface to a Remote Device with Display Rules Provided by the Remote Device"; U.S. Application Serial No. 13/250,947, filed September 30, 2011, entitled "Automatically Adapting User Interfaces for Hands-Free Interaction"; U.S. Application Serial No. 12/987,982, filed January 10, 2011, entitled "Intelligent Automated Assistant"; U.S. Provisional Application Serial No. 61/295,774, filed January 18, 2010, entitled "Intelligent Automated Assistant"; U.S. Provisional Application Serial No. 61/493,201, filed June 3, 2011, entitled "Generating and Processing Data Items that Represent Tasks to Perform"; U.S. Provisional Application Serial No. 61/657,744, filed June 9, 2012, entitled "Automatically Adapting User Interface for Hands-Free Interaction"; U.S. Application Serial No. 12/207,316, filed September 9, 2008, entitled "Radio with Personal DJ"; U.S. Provisional Application Serial No. 61/727,554, filed November 16, 2012, entitled "System and Method for Negotiating Control of a Shared Audio or Visual Resource"; U.S. Application Serial No. 61/832,818, filed June 8, 2013, entitled "Mapping Application with Several User Interfaces,"; U.S. Application Serial No. 13/913,428, filed June 8, 2013, entitled "Application Gateway for Providing Different User Interfaces for Limited Distraction and Non-Limited Distraction Contexts,"; U.S. Provisional Application Serial No. 61/832,842, filed June 8, 2013, entitled "Device, Method, and Graphical User Interface for Synchronizing Two of More Displays," which applications are incorporated herein by reference in their entirety.

TECHNICAL FIELD

[0002] This relates generally to electronic devices for generating user interfaces, including, but not limited to, providing user interfaces for third-party applications.

BACKGROUND

[0003] The use of touch-sensitive surfaces as input devices for computers and other electronic computing devices has increased significantly in recent years. Exemplary touch-sensitive surfaces include touch pads and touch screen displays. Such surfaces are widely used to manipulate user interface objects on a first display associated with a first electronic device.

[0004] Exemplary manipulations include adjusting the position and/or size of one or more user interface objects, activating buttons or opening files/applications represented by user interface objects, or otherwise manipulating user interfaces. Exemplary user interface objects include digital images, video, text, icons, control elements such as buttons and other graphics.

[0005] Devices sometimes have first-party (or native) device software (e.g., an operating system for the device and applications developed by or at the direction of the same entity that developed the operating system) and third-party software that includes applications developed separately from the first-party (or native) device software. Third party applications are sometimes loaded on a device to provide additional functionality to the device.

SUMMARY

[0006] But the user interfaces of third-party applications are often custom and/or application-specific. Additionally, application-specific user interfaces sometimes include words or controls that are too small to be safely used while driving a vehicle. Moreover, different custom and/or application-specific user interfaces for different applications sometimes put controls with similar functions in different locations, so that users will have to take additional time and attention to locate and determine the function of these different controls. Thus, having different custom and/or application-specific user interfaces for different applications is cumbersome and inefficient. In addition, in some circumstances, these user interfaces take longer than necessary for a user manipulate, thereby wasting energy. This latter consideration is particularly important in battery-operated devices.

[0007] Accordingly, there is a need for electronic devices with faster, more efficient methods for providing access to third-party applications and eliminating driver distraction when using third-party applications. Generating user interfaces from a template for use with third-party applications that increase the legibility and ease of use of user interfaces for interacting with third-party applications eliminates driver distraction and improves user efficiency. Such methods optionally complement or replace conventional methods for generating user interfaces and providing access to third-party applications. Such methods reduce the cognitive burden on a user and produce a more efficient human-machine interface. For battery-operated devices, such methods conserve power and increase the time between battery charges.

[0008] The above deficiencies and other problems associated with user interfaces for electronic devices with touch-sensitive surfaces are reduced or eliminated by the disclosed devices. In some embodiments, the device is a desktop computer. In some embodiments, the device is portable (e.g., a notebook computer, tablet computer, or handheld device). In some embodiments, the device has a touchpad. In some embodiments, the device has a touch-sensitive display (also known as a "touch screen" or "touch screen display"). In some embodiments, the device has a graphical user interface (GUI), one or more processors, memory and one or more modules, programs or sets of instructions stored in the memory for performing multiple functions. In some embodiments, the user interacts with the GUI primarily through finger contacts and gestures on the touch-sensitive surface. In some embodiments, the functions optionally include image editing, drawing, presenting, word processing, website creating, disk authoring, spreadsheet making, game playing, telephoning, video conferencing, e-mailing, instant messaging, workout support, digital photographing, digital videoing, web browsing, digital music playing, and/or digital video playing. Executable instructions for performing these functions are, optionally, included in a non-transitory computer readable storage medium or other computer program product configured for execution by one or more processors.

[0009] In accordance with some embodiments, a method is performed at an electronic device with one or more processors and memory. The method includes receiving a first display request to display a user interface of a first third-party application on a respective display that is in communication with the electronic device. In response to receiving the first display request, the method includes: obtaining a first user-interface template that is configured to be used by a plurality of third-party applications; and requesting, from the

first third-party application, one or more values for populating the first user-interface template. The method further includes receiving, from the first third-party application, while the first third-party application is running on the electronic device, a first set of values for populating the first user-interface template; populating the first user- interface template with the first set of values received from the first third-party application; generating a first user interface for the first third-party application using the first user- interface template populated with the first set of values; and sending information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display.

[0010] In accordance with some embodiments, an electronic device includes memory and a processing unit coupled to the memory. The processing unit is configured to receive a first display request to display a user interface of a first third-party application on a respective display unit that is in communication with the electronic device. The processing unit is further configured to, in response to receiving the first display request: obtain a first user-interface template that is configured to be used by a plurality of third-party applications; and request, from the first third-party application, one or more values for populating the first user-interface template. The processing unit is also configured to receive, from the first third-party application, while the first third-party application is running on the electronic device, a first set of values for populating the first user-interface template; populate the first user-interface template with the first set of values received from the first third-party application; generate a first user interface for the first third-party application using the first user-interface template populated with the first set of values; and send information to the respective display unit that enables the first user interface for the first third-party application to be displayed on the respective display unit.

[0011] In accordance with some embodiments, an electronic device includes one or more processors, memory, and one or more programs; the one or more programs are stored in the memory and configured to be executed by the one or more processors and the one or more programs include instructions for performing the operations of any of the methods described herein. In accordance with some embodiments, a non-transitory computer readable storage medium has stored therein instructions which when executed by an electronic device, cause the device to perform the operations of any of the methods described herein. In accordance with some embodiments, an electronic device includes a means for performing the operations of any of the methods described herein. In accordance

with some embodiments, an information processing apparatus, for use in a first electronic device, includes means for performing the operations of any of the methods described herein.

[0012] Thus, electronic devices with displays are provided with faster, more efficient methods for providing user interfaces for third-party applications, thereby increasing the effectiveness, efficiency, and user satisfaction with such devices. Such methods and interfaces may complement or replace conventional methods for generating user interfaces.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] For a better understanding of the various described embodiments, reference should be made to the Description of Embodiments below, in conjunction with the following drawings in which like reference numerals refer to corresponding parts throughout the figures.

[0014] Figure 1A is a block diagram illustrating a portable multifunction device with a touch-sensitive display in accordance with some embodiments.

[0015] Figure 1B is a block diagram illustrating exemplary components for event handling in accordance with some embodiments.

[0016] Figure 2 illustrates a portable multifunction device having a touch screen in accordance with some embodiments.

[0017] Figure 3 is a block diagram of an exemplary multifunction device with a display and a touch-sensitive surface in accordance with some embodiments.

[0018] Figure 4A illustrates an exemplary user interface for a menu of applications on a portable multifunction device in accordance with some embodiments.

[0019] Figure 4B illustrates an exemplary user interface for a multifunction device with a touch-sensitive surface that is separate from the display in accordance with some embodiments.

[0020] Figure 5 is a block diagram illustrating an operating environment in which a portable multifunction device communicates with an external information presentation system and/or a server in accordance with some embodiments.

[0021] Figures 6A-6C illustrate exemplary user interfaces generated from a template in accordance with some embodiments.

[0022] Figures 7A-7I are flow diagrams illustrating a method of generating user interfaces from a template in accordance with some embodiments.

[0023] Figure 8 is a functional block diagram of an electronic device in accordance with some embodiments.

DESCRIPTION OF EMBODIMENTS

[0024] Many electronic devices use graphical user interfaces. Many of the graphical user interfaces are custom designed, or are specific to an individual application. The methods and devices described below improve on these methods by generating user interfaces from a user-interface template for use with third-party applications. In some embodiments, the user-interface template is a generic or application-agnostic user-interface template that is used by a plurality of different applications. These user interfaces are displayed on a respective display. Using the same generic user interface template for a number of different third-party applications helps to standardize the user interfaces of the third-party applications. This standardization makes the third-party applications more intuitive, and easier and more convenient to interact with, while still providing the user with access to the functionality of the third-party applications.

[0025] These templates are particularly helpful in situations where a device is providing access to the third-party application on an information presentation system that is external to the device, such as a smart phone providing access to a third-party application on the smartphone via a touch screen display in a car. The information presentation system optionally has a display with one of a plurality of different sets of operational properties (e.g., display dimensions, control types, etc.). In some embodiments, the respective display is a display in a vehicle information display system that is visible from a driver seat of a vehicle, so as to improve the ease of use of the third-party application by the driver and/or passengers in the vehicle.

[0026] Below, Figures 1A-1B, 2, and 3 provide a description of exemplary devices. Figures 4A-4B and 6A-6C illustrate exemplary user interfaces generated from a template. Figures 7A-7I are flow diagrams illustrating a method of providing user interfaces for third-party applications. Figure 8 is a functional block diagram of an electronic device. The user interfaces in Figures 6A-6C are used to illustrate the processes in Figures 7A-7I.

EXEMPLARY DEVICES

[0027] Reference will now be made in detail to embodiments, examples of which are illustrated in the accompanying drawings. In the following detailed description, numerous

specific details are set forth in order to provide a thorough understanding of the various described embodiments. However, it will be apparent to one of ordinary skill in the art that the various described embodiments may be practiced without these specific details. In other instances, well-known methods, procedures, components, circuits, and networks have not been described in detail so as not to unnecessarily obscure aspects of the embodiments.

[0028] It will also be understood that, although the terms first, second, etc. are, in some instances, used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first contact could be termed a second contact, and, similarly, a second contact could be termed a first contact, without departing from the scope of the various described embodiments. The first contact and the second contact are both contacts, but they are not the same contact.

[0029] The terminology used in the description of the various described embodiments herein is for the purpose of describing particular embodiments only and is not intended to be limiting. As used in the description of the various described embodiments 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.

[0030] As used herein, the term "if" is, optionally, 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" is, optionally, 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.

[0031] Embodiments of electronic devices, user interfaces for such devices, and associated processes for using such devices are described. In some embodiments, the device is a portable communications device, such as a mobile telephone, that also contains other functions, such as PDA and/or music player functions. Exemplary embodiments of portable

multifunction devices include, without limitation, the iPhone®, iPod Touch®, and iPad® devices from Apple Inc. of Cupertino, California. Other portable electronic devices, such as laptops or tablet computers with touch-sensitive surfaces (e.g., touch screen displays and/or touch pads), are, optionally, used. It should also be understood that, in some embodiments, the device is not a portable communications device, but is a desktop computer with a touch-sensitive surface (e.g., a touch screen display and/or a touch pad).

[0032] In the discussion that follows, an electronic device that includes a display and a touch-sensitive surface is described. It should be understood, however, that the electronic device optionally includes one or more other physical user-interface devices, such as a physical keyboard, a mouse and/or a joystick.

[0033] The device typically supports a variety of applications, such as one or more of the following: a drawing application, a presentation application, a word processing application, a website creation application, a disk authoring application, a spreadsheet application, a gaming application, a telephone application, a video conferencing application, an e-mail application, an instant messaging application, a workout support application, a photo management application, a digital camera application, a digital video camera application, a web browsing application, a digital music player application, and/or a digital video player application.

[0034] The various applications that are executed on the device optionally use at least one common physical user-interface device, such as the touch-sensitive surface. One or more functions of the touch-sensitive surface as well as corresponding information displayed on the device are, optionally, adjusted and/or varied from one application to the next and/or within a respective application. In this way, a common physical architecture (such as the touch-sensitive surface) of the device optionally supports the variety of applications with user interfaces that are intuitive and transparent to the user.

[0035] Attention is now directed toward embodiments of portable devices with touch-sensitive displays. Figure 1A is a block diagram illustrating portable multifunction device 100 with touch-sensitive displays 112 in accordance with some embodiments. Touch-sensitive display 112 is sometimes called a "touch screen" for convenience, and is sometimes known as or called a touch-sensitive display system. Device 100 includes memory 102 (which optionally includes one or more computer readable storage mediums), memory controller 122, one or more processing units (CPU's) 120, peripherals interface 118, RF circuitry 108, audio circuitry 110, speaker 111, microphone 113, input/output (I/O)

subsystem 106, other input or control devices 116, and external port 124. Device 100 optionally includes one or more optical sensors 164. Device 100 optionally includes one or more intensity sensors 165 for detecting intensity of contacts on device 100 (e.g., a touch-sensitive surface such as touch-sensitive display system 112 of device 100). Device 100 optionally includes one or more tactile output generators 167 for generating tactile outputs on device 100 (e.g., generating tactile outputs on a touch-sensitive surface such as touch-sensitive display system 112 of device 100 or touchpad 355 of device 300). These components optionally communicate over one or more communication buses or signal lines 103.

[0036] 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).

[0037] 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.

[0038] It should be appreciated that device 100 is only one example of a portable multifunction device, and that device 100 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 Figure 1A are implemented in hardware, software, or a combination of hardware and software, including one or more signal processing and/or application specific integrated circuits.

[0039] Memory 102 optionally includes high-speed random access memory and optionally also includes non-volatile memory, such as one or more magnetic disk storage devices, flash memory devices, or other non-volatile solid-state memory devices. Access to

memory 102 by other components of device 100, such as CPU 120 and the peripherals interface 118, is, optionally, controlled by memory controller 122.

[0040] Peripherals interface 118 can be used to couple input and output peripherals of the device to CPU 120 and memory 102. The one or more processors 120 run or execute various software programs and/or sets of instructions stored in memory 102 to perform various functions for device 100 and to process data.

[0041] In some embodiments, peripherals interface 118, CPU 120, and memory controller 122 are, optionally, implemented on a single chip, such as chip 104. In some other embodiments, they are, optionally, implemented on separate chips.

[0042] RF (radio frequency) circuitry 108 receives and sends RF signals, also called electromagnetic signals. RF circuitry 108 converts electrical signals to/from electromagnetic signals and communicates with communications networks and other communications devices via the electromagnetic signals. RF circuitry 108 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 108 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 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, Wireless Fidelity (Wi-Fi) (e.g., IEEE 802.11a, IEEE 802.11b, IEEE 802.11g and/or IEEE 802.11n), 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.

[0043] Audio circuitry 110, speaker 111, and microphone 113 provide an audio interface between a user and device 100. Audio circuitry 110 receives audio data from peripherals interface 118, converts the audio data to an electrical signal, and transmits the electrical signal to speaker 111. Speaker 111 converts the electrical signal to human-audible sound waves. Audio circuitry 110 also receives electrical signals converted by microphone 113 from sound waves. Audio circuitry 110 converts the electrical signal to audio data and transmits the audio data to peripherals interface 118 for processing. Audio data is, optionally, retrieved from and/or transmitted to memory 102 and/or RF circuitry 108 by peripherals interface 118. In some embodiments, audio circuitry 110 also includes a headset jack (e.g., 212, Figure 2). The headset jack provides an interface between audio circuitry 110 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).

[0044] I/O subsystem 106 couples input/output peripherals on device 100, such as touch screen 112 and other input control devices 116, to peripherals interface 118. I/O subsystem 106 optionally includes display controller 156, optical sensor controller 158, intensity sensor controller 159, haptic feedback controller 161 and one or more input controllers 160 for other input or control devices. The one or more input controllers 160 receive/send electrical signals from/to other input or control devices 116. The other input control devices 116 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) 160 are, optionally, coupled to any (or none) of the following: a keyboard, infrared port, USB port, and a pointer device such as a mouse. The one or more buttons (e.g., 208, Figure 2) optionally include an up/down button for volume control of speaker 111 and/or microphone 113. The one or more buttons optionally include a push button (e.g., 206, Figure 2).

[0045] Touch-sensitive display 112 provides an input interface and an output interface between the device and a user. Display controller 156 receives and/or sends electrical signals from/to touch screen 112. Touch screen 112 displays visual output to the user. The visual output optionally includes graphics, text, icons, video, and any combination

thereof(collectively termed "graphics"). In some embodiments, some or all of the visual output corresponds to user-interface objects.

[0046] Touch screen 112 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 112 and display controller 156 (along with any associated modules and/or sets of instructions in memory 102) detect contact (and any movement or breaking of the contact) on touch screen 112 and converts 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 112. In an exemplary embodiment, a point of contact between touch screen 112 and the user corresponds to a finger of the user.

[0047] Touch screen 112 optionally uses LCD (liquid crystal display) technology, LPD (light emitting polymer display) technology, or LED (light emitting diode) technology, although other display technologies are used in other embodiments. Touch screen 112 and display controller 156 optionally 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 112. In an exemplary embodiment, projected mutual capacitance sensing technology is used, such as that found in the iPhone®, iPod Touch®, and iPad® from Apple Inc. of Cupertino, California.

[0048] Touch screen 112 optionally has a video resolution in excess of 100 dpi. In some embodiments, the touch screen has a video resolution of approximately 160 dpi. The user optionally makes contact with touch screen 112 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.

[0049] In some embodiments, in addition to the touch screen, device 100 optionally includes a touchpad (not shown) for activating or deactivating particular functions. In some embodiments, the touchpad is a touch-sensitive area of the device that, unlike the touch screen, does not display visual output. The touchpad is, optionally, a touch-sensitive surface

that is separate from touch screen 112 or an extension of the touch-sensitive surface formed by the touch screen.

[0050] Device 100 also includes power system 162 for powering the various components. Power system 162 optionally includes a power management system, one or more power sources (e.g., battery, alternating current (AC)), a recharging system, a power failure detection circuit, a power converter or inverter, a power status indicator (e.g., a light-emitting diode (LED)) and any other components associated with the generation, management and distribution of power in portable devices.

[0051] Device 100 optionally also includes one or more optical sensors 164. Figure 1A shows an optical sensor coupled to optical sensor controller 158 in I/O subsystem 106. Optical sensor 164 optionally includes charge-coupled device (CCD) or complementary metal-oxide semiconductor (CMOS) phototransistors. Optical sensor 164 receives light from the environment, projected through one or more lens, and converts the light to data representing an image. In conjunction with imaging module 143 (also called a camera module), optical sensor 164 optionally captures still images or video. In some embodiments, an optical sensor is located on the back of device 100, opposite touch screen display 112 on the front of the device, so that the touch screen display is enabled for use as a viewfinder for still and/or video image acquisition. In some embodiments, another optical sensor is located on the front of the device so that the user's image is, optionally, obtained for videoconferencing while the user views the other video conference participants on the touch screen display.

[0052] Device 100 optionally also includes one or more contact intensity sensors 165. Figure 1A shows a contact intensity sensor coupled to intensity sensor controller 159 in I/O subsystem 106. Contact intensity sensor 165 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 165 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 112). In some embodiments, at least one contact intensity sensor is located on the back of device 100, opposite touch screen display 112 which is located on the front of device 100.

[0053] Device 100 optionally also includes one or more proximity sensors 166. Figure 1A shows proximity sensor 166 coupled to peripherals interface 118. Alternately, proximity sensor 166 is coupled to input controller 160 in I/O subsystem 106. In some embodiments, the proximity sensor turns off and disables touch screen 112 when the multifunction device is placed near the user's ear (e.g., when the user is making a phone call).

[0054] Device 100 optionally also includes one or more tactile output generators 167. Figure 1A shows a tactile output generator coupled to haptic feedback controller 161 in I/O subsystem 106. Tactile output generator 167 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 165 receives tactile feedback generation instructions from haptic feedback module 133 and generates tactile outputs on device 100 that are capable of being sensed by a user of device 100. 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 112) and, optionally, generates a tactile output by moving the touch-sensitive surface vertically (e.g., in/out of a surface of device 100) or laterally (e.g., back and forth in the same plane as a surface of device 100). In some embodiments, at least one tactile output generator sensor is located on the back of device 100, opposite touch screen display 112 which is located on the front of device 100.

[0055] Device 100 optionally also includes one or more accelerometers 168. Figure 1A shows accelerometer 168 coupled to peripherals interface 118. Alternately, accelerometer 168 is, optionally, coupled to an input controller 160 in I/O subsystem 106. 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 100 optionally includes, in addition to accelerometer(s) 168, 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 100.

[0056] In some embodiments, the software components stored in memory 102 include operating system 126, communication module (or set of instructions) 128, contact/motion module (or set of instructions) 130, graphics module (or set of instructions) 132, text input

module (or set of instructions) 134, Global Positioning System (GPS) module (or set of instructions) 135, and applications (or sets of instructions) 136. Furthermore, in some embodiments memory 102 stores device/global internal state 157, as shown in Figures 1A and 3. Device/global internal state 157 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 112; sensor state, including information obtained from the device's various sensors and input control devices 116; and location information concerning the device's location and/or attitude.

[0057] Operating system 126 (e.g., Darwin, RTXC, LINUX, UNIX, OS X, WINDOWS, or an embedded operating system such as VxWorks) includes various software components and/or drivers for controlling and managing general system tasks (e.g., memory management, storage device control, power management, etc.) and facilitates communication between various hardware and software components.

[0058] Communication module 128 facilitates communication with other devices over one or more external ports 124 and also includes various software components for handling data received by RF circuitry 108 and/or external port 124. External port 124 (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. In some embodiments, the external port is a multi-pin (e.g., 8-pin) connector that is the same as, or similar to and/or compatible with the 8-pin connector (e.g., Lightning connector) used on iPhone and iPod (trademark of Apple Inc.) devices.

[0059] Contact/motion module 130 optionally detects contact with touch screen 112 (in conjunction with display controller 156) and other touch sensitive devices (e.g., a touchpad or physical click wheel). Contact/motion module 130 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 130 receives contact data from the touch-sensitive surface. Determining movement of the point of contact, which is represented by a series of contact data, optionally includes determining speed (magnitude), velocity (magnitude and direction), and/or an acceleration (a change in magnitude and/or direction) of the point of contact. These operations are, optionally, applied to single contacts (e.g., one finger contacts) or to multiple simultaneous contacts (e.g., "multitouch"/multiple finger contacts). In some embodiments, contact/motion module 130 and display controller 156 detect contact on a touchpad.

[0060] In some embodiments, contact/motion module 130 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 100). For example, a mouse "click" threshold of a trackpad or touch screen display can be set to any of a large range of predefined thresholds 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).

[0061] Contact/motion module 130 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 (lift off) 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 (lift off) event.

[0062] Graphics module 132 includes various known software components for rendering and displaying graphics on touch screen 112 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.

[0063] In some embodiments, graphics module 132 stores data representing graphics to be used. Each graphic is, optionally, assigned a corresponding code. Graphics module 132 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 156.

[0064] Haptic feedback module 133 includes various software components for generating instructions used by tactile output generator(s) 167 to produce tactile outputs at one or more locations on device 100 in response to user interactions with device 100.

[0065] Text input module 134, which is, optionally, a component of graphics module 132, provides soft keyboards for entering text in various applications (e.g., contacts 137, e-mail140, IM 141, browser 147, and any other application that needs text input).

[0066] GPS module 135 determines the location of the device and provides this information for use in various applications (e.g., to telephone 138 for use in location-based dialing, to camera 143 as picture/video metadata, and to applications that provide location-based services such as weather widgets, local yellow page widgets, and map/navigation widgets).

[0067] Applications 136 optionally include the following modules (or sets of instructions), or a subset or superset thereof:

- contacts module 137 (sometimes called an address book or contact list);
- telephone module 138;
- video conferencing module 139;
- e-mail client module 140;
- instant messaging (IM) module 141;
- Internet radio module 142;
- camera module 143 for still and/or video images;
- image management module 144;
- browser module 147;
- calendar module 148;

- widget modules 149, which optionally include one or more of: weather widget 149-1, stocks widget 149-2, calculator widget 149-3, alarm clock widget 149-4, dictionary widget 149-5, and other widgets obtained by the user, as well as user-created widgets 149-6;
- digital personal assistant module 150;
- vehicle integration module 151;
- video and music player module 152, which is, optionally, made up of a video player module and a music player module;
- notes module 153;
- map module 154; and/or
- music library module 155.

[0068] Examples of other applications 136 that are, optionally, stored in memory 102 include other word processing applications, other image editing applications, drawing applications, presentation applications, JAVA-enabled applications, encryption, digital rights management, voice recognition, and voice replication.

[0069] In conjunction with touch screen 112, display controller 156, contact module 130, graphics module 132, and text input module 134, contacts module 137 are, optionally, used to manage an address book or contact list (e.g., stored in application internal state 192 of contacts module 137 in memory 102 or memory 370), 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 138, video conference 139, e-mail 140, or IM 141; and so forth.

[0070] In conjunction with RF circuitry 108, audio circuitry 110, speaker 111, microphone 113, touch screen 112, display controller 156, contact module 130, graphics module 132, and text input module 134, telephone module 138 are, optionally, used to enter a sequence of characters corresponding to a telephone number, access one or more telephone numbers in address book 137, 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 optionally uses any of a plurality of communications standards, protocols and technologies.

[0071] In conjunction with RF circuitry 108, audio circuitry 110, speaker 111, microphone 113, touch screen 112, display controller 156, optical sensor 164, optical sensor controller 158, contact module 130, graphics module 132, text input module 134, contact list 137, and telephone module 138, videoconferencing module 139 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.

[0072] In conjunction with RF circuitry 108, touch screen 112, display controller 156, contact module 130, graphics module 132, and text input module 134, e-mail client module 140 includes executable instructions to create, send, receive, and manage e-mail in response to user instructions. In conjunction with image management module 144, e-mail client module 140 makes it very easy to create and send e-mails with still or video images taken with camera module 143.

[0073] In conjunction with RF circuitry 108, touch screen 112, display controller 156, contact module 130, graphics module 132, and text input module 134, the instant messaging module 141 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 optionally include graphics, photos, audio files, video files and/or other attachments as are supported in a 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).

[0074] In conjunction with touch screen 112, display system controller 156, contact module 130, graphics module 132, audio circuitry 110, speaker 111, and RF circuitry 108, Internet radio module 142 is a prophetic third-party application that includes executable instructions to select and play Internet radio stations.

[0075] In conjunction with touch screen 112, display controller 156, optical sensor(s) 164, optical sensor controller 158, contact module 130, graphics module 132, and image management module 144, camera module 143 includes executable instructions to capture

still images or video (including a video stream) and store them into memory 102, modify characteristics of a still image or video, or delete a still image or video from memory 102.

[0076] In conjunction with touch screen 112, display controller 156, contact module 130, graphics module 132, text input module 134, and camera module 143, image management module 144 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.

[0077] In conjunction with RF circuitry 108, touch screen 112, display system controller 156, contact module 130, graphics module 132, and text input module 134, browser module 147 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.

[0078] In conjunction with RF circuitry 108, touch screen 112, display system controller 156, contact module 130, graphics module 132, text input module 134, e-mail client module 140, and browser module 147, calendar module 148 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.

[0079] In conjunction with RF circuitry 108, touch screen 112, display system controller 156, contact module 130, graphics module 132, text input module 134, and browser module 147, widget modules 149 are mini-applications that are, optionally, downloaded and used by a user (e.g., weather widget 149-1, stocks widget 149-2, calculator widget 149-3, alarm clock widget 149-4, and dictionary widget 149-5) or created by the user (e.g., user-created widget 149-6). 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).

[0080] In conjunction with RF circuitry 108, audio circuitry 110, speaker 111, microphone 113, touch screen 112, display controller 156, contact module 130, graphics module 132, and text input module 134, digital personal assistant module 150 records voice commands and sends information representative of the recorded voice commands to a server such as server 510 in Figure 5 for analysis, and responds to the voice commands based on a response from the server.

[0081] Vehicle integration module 151 includes executable instructions for one or more intermediation processes that control a vehicle information display system in a vehicle (e.g., a car, a truck, a van, etc.) that provides a user interface on a respective display of the vehicle information display system (e.g., display 547 of external information presentation system 540 in Figure 5), such as for a mapping application or a music application. The vehicle integration application converts information from third-party applications into content for display by the vehicle integration application on the respective display of the vehicle information display system.

[0082] In conjunction with touch screen 112, display system controller 156, contact module 130, graphics module 132, audio circuitry 110, speaker 111, RF circuitry 108, and browser module 147, video and music player module 152 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 112 or on an external, connected display via external port 124). In some embodiments, device 100 optionally includes the functionality of an MP3 player, such as an iPod (trademark of Apple Inc.).

[0083] In conjunction with touch screen 112, display controller 156, contact module 130, graphics module 132, and text input module 134, notes module 153 includes executable instructions to create and manage notes, to do lists, and the like in accordance with user instructions.

[0084] In conjunction with RF circuitry 108, touch screen 112, display system controller 156, contact module 130, graphics module 132, text input module 134, GPS module 135, and browser module 147, map module 154 are, optionally, 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.

[0085] In conjunction with touch screen 112, display system controller 156, contact module 130, graphics module 132, audio circuitry 110, speaker Ill, and RF circuitry 108, music library module 155 is a prophetic third-party application that includes executable instructions to select and play songs from a music library.

[0086] Each of the above identified modules and applications correspond 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 (i.e., sets of instructions) need not be implemented as separate software programs, procedures or modules, and thus various subsets of these modules are, optionally, combined or otherwise re-arranged in various embodiments. In some embodiments, memory 102 optionally stores a subset of the modules and data structures identified above. Furthermore, memory 102 optionally stores additional modules and data structures not described above.

[0087] In some embodiments, device 100 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 100, the number of physical input control devices (such as push buttons, dials, and the like) on device 100 is, optionally, reduced.

[0088] 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 100 to a main, home, or root menu from any user interface that is displayed on device 100. 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.

[0089] Figure 1B is a block diagram illustrating exemplary components for event handling in accordance with some embodiments. In some embodiments, memory 102 (in Figures 1A) or 370 (Figure 3) includes event sorter 170 (e.g., in operating system 126) and a respective application 136-1 (e.g., any of the aforementioned applications 137-13, 155, 380-390).

[0090] Event sorter 170 receives event information and determines the application 136-1 and application view 191 of application 136-1 to which to deliver the event information. Event sorter 170 includes event monitor 171 and event dispatcher module 174. In some embodiments, application 136-1 includes application internal state 192, which indicates the current application view(s) displayed on touch sensitive display 112 when the application is active or executing. In some embodiments, device/global internal state 157 is used by event sorter 170 to determine which application(s) is (are) currently active, and application internal state 192 is used by event sorter 170 to determine application views 191 to which to deliver event information.

[0091] In some embodiments, application internal state 192 includes additional information, such as one or more of: resume information to be used when application 136-1 resumes execution, user interface state information that indicates information being displayed or that is ready for display by application 136-1, a state queue for enabling the user to go back to a prior state or view of application 136-1, and a redo/undo queue of previous actions taken by the user.

[0092] Event monitor 171 receives event information from peripherals interface 118. Event information includes information about a sub-event (e.g., a user touch on touch-sensitive display 112, as part of a multi-touch gesture). Peripherals interface 118 transmits information it receives from I/O subsystem 106 or a sensor, such as proximity sensor 166, accelerometer(s) 168, and/or microphone 113 (through audio circuitry 110). Information that peripherals interface 118 receives from I/O subsystem 106 includes information from touch-sensitive display 112 or a touch-sensitive surface.

[0093] In some embodiments, event monitor 171 sends requests to the peripherals interface 118 at predetermined intervals. In response, peripherals interface 118 transmits event information. In other embodiments, peripheral interface 118 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).

[0094] In some embodiments, event sorter 170 also includes a hit view determination module 172 and/or an active event recognizer determination module 173.

[0095] Hit view determination module 172 provides software procedures for determining where a sub-event has taken place within one or more views, when touch sensitive display 112 displays more than one view. Views are made up of controls and other elements that a user can see on the display.

[0096] 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 optionally correspond to programmatic levels within a programmatic or view hierarchy of the application. For example, the lowest level view in which a touch is detected is, optionally, called the hit view, and the set of events that are recognized as proper inputs are, optionally, determined based, at least in part, on the hit view of the initial touch that begins a touch-based gesture.

[0097] Hit view determination module 172 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 172 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 (i.e., 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, 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.

[0098] Active event recognizer determination module 173 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 173 determines that only the hit view should receive a particular sequence of sub-events. In other embodiments, active event recognizer determination module 173 determines that all views that include the physic allocation 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.

[0099] Event dispatcher module 174 dispatches the event information to an event recognizer (e.g., event recognizer 180). In embodiments including active event recognizer determination module 173, event dispatcher module 174 delivers the event information to an event recognizer determined by active event recognizer determination module 173. In some embodiments, event dispatcher module 174 stores in an event queue the event information, which is retrieved by a respective event receiver module 182.

[0100] In some embodiments, operating system 126 includes event sorter 170. Alternatively, application 136-1 includes event sorter 170. In yet other embodiments, event sorter 170 is a stand-alone module, or a part of another module stored in memory 102, such as contact/motion module 130.

[0101] In some embodiments, application 136-1 includes a plurality of event handlers 190 and one or more application views 191, each of which includes instructions for handling touch events that occur within a respective view of the application's user interface. Each application view 191 of the application 136-1 includes one or more event recognizers 180. Typically, a respective application view 191 includes a plurality of event

recognizers 180. In other embodiments, one or more of event recognizers 180 are part of a separate module, such as a user interface kit (not shown) or a higher level object from which application 136-1 inherits methods and other properties. In some embodiments, a respective event handler 190 includes one or more of: data updater 176, object updater 177, GUI updater 178, and/or event data 179 received from event sorter 170. Event handler 190 optionally utilizes or calls data updater 176, object updater 177 or GUI updater 178 to update the application internal state 192. Alternatively, one or more of the application views 191 includes one or more respective event handlers 190. Also, in some embodiments, one or more of data updater 176, object updater 177, and GUI updater 178 are included in a respective application view 191.

[0102] A respective event recognizer 180 receives event information (e.g., event data 179) from event sorter 170, and identifies an event from the event information. Event recognizer 180 includes event receiver 182 and event comparator 184. In some embodiments, event recognizer 180 also includes at least a subset of: metadata 183, and event delivery instructions 188 (which optionally include sub-event delivery instructions).

[0103] Event receiver 182 receives event information from event sorter 170. 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 optionally also includes speed and direction of the sub-event. In some embodiments, events include rotation of the device from one orientation to another (e.g., from a portrait orientation to a landscape orientation, or vice versa), and the event information includes corresponding information about the current orientation (also called device attitude) of the device.

[0104] Event comparator 184 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 184 includes event definitions 186. Event definitions 186 contain definitions of events (e.g., predefined sequences of sub-events), for example, event 1 (187-1), event 2 (187-2), and others. In some embodiments, sub-events in an event 187 include, for example, touch begin, touch end, touch movement, touch cancellation, and multiple touching. In one example, the definition for event 1 (187-1) 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 lift-off (touch end) for a predetermined phase, a second touch (touch begin) on the displayed object for a predetermined phase, and a second lift-off (touch end) for a predetermined phase. In another example, the definition for event 2 (187-2) 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 112, and lift-off of the touch (touch end). In some embodiments, the event also includes information for one or more associated event handlers 190.

[0105] In some embodiments, event definition 187 includes a definition of an event for a respective user-interface object. In some embodiments, event comparator 184 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 112, when a touch is detected on touch-sensitive display 112, event comparator 184 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 190, the event comparator uses the result of the hit test to determine which event handler 190 should be activated. For example, event comparator 184 selects an event handler associated with the sub-event and the object triggering the hit test.

[0106] In some embodiments, the definition for a respective event 187 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.

[0107] When a respective event recognizer 180 determines that the series of sub-events do not match any of the events in event definitions 186, the respective event recognizer 180 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.

[0108] In some embodiments, a respective event recognizer 180 includes metadata 183 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 183 includes configurable properties, flags, and/or lists that

indicate how event recognizers interact, or are enabled to interact, with one another. In some embodiments, metadata 183 includes configurable properties, flags, and/or lists that indicate whether sub-events are delivered to varying levels in the view or programmatic hierarchy.

[0109] In some embodiments, a respective event recognizer 180 activates event handler 190 associated with an event when one or more particular sub-events of an event are recognized. In some embodiments, a respective event recognizer 180 delivers event information associated with the event to event handler 190. Activating an event handler 190 is distinct from sending (and deferred sending) sub-events to a respective hit view. In some embodiments, event recognizer 180 throws a flag associated with the recognized event, and event handler 190 associated with the flag catches the flag and performs a predefined process.

[0110] In some embodiments, event delivery instructions 188 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.

[0111] In some embodiments, data updater 176 creates and updates data used in application 136-1. For example, data updater 176 updates the telephone number used in contacts module 137, or stores a video file used in video player module 145. In some embodiments, object updater 177 creates and updates objects used in application 136-1. For example, object updater 176 creates a new user-interface object or updates the position of a user-interface object. GUI updater 178 updates the GUI. For example, GUI updater 178 prepares display information and sends it to graphics module 132 for display on a touch- sensitive display.

[0112] In some embodiments, event handler(s) 190 includes or has access to data updater 176, object updater 177, and GUI updater 178. In some embodiments, data updater 176, object updater 177, and GUI updater 178 are included in a single module of a respective application 136-1 or application view 191. In other embodiments, they are included in two or more software modules.

[0113] 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 100 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 touch-pads; 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.

[0114] Figure 2 illustrates a portable multifunction device 100 having a touch screen 112 in accordance with some embodiments. The touch screen optionally displays one or more graphics within user interface (UI) 200. 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 202 (not drawn to scale in the figure) or one or more styluses 203 (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 100. 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.

[0115] Device 100 optionally also includes one or more physical buttons, such as "home" or menu button 204. As described previously, menu button 204 is, optionally, used to navigate to any application 136 in a set of applications that are, optionally executed on device 100. Alternatively, in some embodiments, the menu button is implemented as a soft key in a GUI displayed on touch screen 112.

[0116] In one embodiment, device 100 includes touch screen 112, menu button 204, push button 206 for powering the device on/off and locking the device, volume adjustment button(s) 208, Subscriber Identity Module (SIM) card slot 210, head set jack 212, and docking/charging external port 124. Push button 206 is, optionally, used to tum 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 100 also accepts verbal input for activation or deactivation of some functions through microphone 113. Device 100 also, optionally, includes one or more contact intensity sensors 165 for detecting intensity of contacts on touch screen 112 and/or one or more tactile output generators 167 for generating tactile outputs for a user of device 100.

[0117] Figure 3 is a block diagram of an exemplary multifunction device with a display and a touch-sensitive surface in accordance with some embodiments. Device 300 need not be portable. In some embodiments, device 300 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 300 typically includes one or more processing units (CPU's) 310, one or more network or other communications interfaces 360, memory 370, and one or more communication buses 320 for interconnecting these components. Communication buses 320 optionally include circuitry (sometimes called a chipset) that interconnects and controls communications between system components. Device 300 includes input/output (I/O) interface 330 comprising display 340, which is typically a touch screen display (e.g., touch screen display 112). I/O interface 330 also optionally includes a keyboard and/or mouse (or other pointing device) 350 and touchpad 355, tactile output generator 357 for generating tactile outputs on device 300 (e.g., similar to tactile output generator(s) 167 described above with reference to Figure 1A), sensors 359 (e.g., optical, acceleration, proximity, touch-sensitive, and/or contact intensity sensors similar to contact intensity sensor(s) 165 described above with reference to Figure 1A). Memory 370 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 370 optionally includes one or more storage devices remotely located from CPU(s) 310. In some embodiments, memory 370 stores programs, modules, and data structures analogous to the programs, modules, and data structures stored in memory 102 of portable multifunction device 100 (Figure 1A), or a subset thereof. Furthermore, memory 370 optionally stores additional programs, modules, and data structures not present in memory 102 of portable multifunction device 100. For example, memory 370 of device 300 optionally stores drawing module 380, presentation module 382, word processing module

384, website creation module 386, disk authoring module 388, and/or spreadsheet module 390, while memory 102 of portable multifunction device 100 (Figure 1A) optionally does not store these modules.

[0118] Each of the above identified elements in Figure 3 is, optionally, 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 (i.e., sets of instructions) need not be implemented as separate software programs, procedures or modules, and thus various subsets of these modules are, optionally, combined or otherwise re-arranged in various embodiments. In some embodiments, memory 370 optionally stores a subset of the modules and data structures identified above. Furthermore, memory 370 optionally stores additional modules and data structures not described above.

[0119] Attention is now directed towards embodiments of user interfaces ("UI") that is, optionally, implemented on portable multifunction device 100.

[0120] Figure 4A illustrates an exemplary user interface for a menu of applications on portable multifunction device 100 in accordance with some embodiments. Similar user interfaces are, optionally, implemented on device 300. In some embodiments, user interface 400 includes the following elements, or a subset or superset thereof:

- Signal strength indicator(s) 402 for wireless communication(s), such as cellular and Wi-Fi signals;
- Time 404;
- Bluetooth indicator 405;
- Battery status indicator 406;
- Tray 408 with icons for frequently used applications, such as:
    - Icon 416 for telephone module 138, labeled "Phone," which optionally includes an indicator 414 of the number of missed calls or voicemail messages;
    - Icon 418 for e-mail client module 140, labeled "Mail," which optionally includes an indicator 410 of the number of unread e-mails;
    - Icon 420 for browser module 147, labeled "Browser;" and
    - Icon 422 for video and music player module 152, also referred to as iPod (trademark of Apple Inc.) module 152, labeled "iPod;" and
- Icons for other applications, such as:
    - Icon 424 for IM module 141, labeled "Text;"

- Icon 426 for calendar module 148, labeled "Calendar;"
- Icon 428 for image management module 144, labeled "Photos;"
- Icon 430 for camera module 143, labeled "Camera;"
- Icon 432 for music library module 155, labeled "Music Library"
- Icon 434 for stocks widget 149-2, labeled "Stocks;"
- Icon 436 for map module 154, labeled "Map;"
- Icon 438 for weather widget 149-1, labeled "Weather;"
- Icon 440 for alarm clock widget 149-4, labeled "Clock;"
- Icon 442 for Internet radio module 142, labeled "Internet Radio;"
- Icon 444 for notes module 153, labeled ''Notes;" and
- Icon 446 for a settings application or module, which provides access to settings for device 100 and its various applications 136.

[0121] It should be noted that the icon labels illustrated in Figure 4A are merely exemplary. For example, icon 422 for video and music player module 152 are 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.

[0122] Figure 4B illustrates an exemplary user interface on a device (e.g., device 300, Figure 3) with a touch-sensitive surface 451 (e.g., a tablet or touchpad 355, Figure 3) that is separate from the display 450 (e.g., touch screen display 112). Device 300 also, optionally, includes one or more contact intensity sensors (e.g., one or more of sensors 357) for detecting intensity of contacts on touch-sensitive surface 451 and/or one or more tactile output generators 359 for generating tactile outputs for a user of device 300.

[0123] Although some of the examples which follow will be given with reference to inputs on touch screen display 112 (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 Figure 4B. In some embodiments the touch sensitive surface (e.g., 451 in Figure 4B) has a primary axis (e.g., 452 in Figure 4B) that corresponds to a primary axis (e.g., 453 in Figure 4B) on the display (e.g., 450). In accordance with these embodiments, the device detects contacts (e.g., 460 and 462 in Figure 4B) with the touch-sensitive surface 451 at locations that correspond to respective

locations on the display (e.g., in Figure 4B, 460 corresponds to 468 and 462 corresponds to 470). In this way, user inputs (e.g., contacts 460 and 462, and movements thereof) detected by the device on the touch-sensitive surface (e.g., 451 in Figure 4B) are used by the device to manipulate the user interface on the display (e.g., 450 in Figure 4B) 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.

[0124] 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.

[0125] Figure 5 illustrates a block diagram of an operating environment 500 in accordance with some embodiments. Operating environment 500 includes server 510, one or more communications networks 505, portable multifunction device 100, and external information presentation system 540. In some embodiments, external information presentation system 540 is implemented in a vehicle.

[0126] Server 510 typically includes one or more processing units (CPUs) 512 for executing modules, programs and/or instructions stored in memory 524 and thereby performing processing operations, one or more network or other communications interfaces 520, memory 524, and one or more communication buses 522 for interconnecting these components. Communication buses 522 optionally include circuitry (sometimes called a chipset) that interconnects and controls communications between system components. Memory 524 includes high-speed random access memory, such as DRAM, SRAM, DDR RAM or other random access solid state memory devices, and may include 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 524 optionally includes one or more storage devices remotely located from the CPU(s) 512. Memory 524, or alternately the non-volatile memory device(s) within memory 524, comprises a non-transitory computer readable storage medium. In some embodiments, memory 524, or the computer readable storage medium of memory 524 stores the following programs, modules, and data structures, or a subset thereof:

- an operating system 526 that includes procedures for handling various basic system services and for performing hardware dependent tasks; and
- a network communication module 528 that is used for connecting (wired or wireless) server 510 to other computing devices via the one or more communication network interfaces 520 and one or more communication networks 505, such as the Internet, other wide area networks, local area networks, metropolitan area networks, and so on.

[0127] Portable multifunction device 100 typically includes the components described with reference to Figures 1A-1B and/or 3.

[0128] External information presentation system 540 typically includes one or more processing units (CPUs) 542 for executing modules, programs and/or instructions stored in memory 554 and thereby performing processing operations, one or more network or other communications interfaces 550, memory 554, and one or more communication buses 552 for interconnecting these components. External information presentation system 540 optionally includes a user interface 546 comprising a display device 547 and controls 548 (e.g., mechanical affordances, buttons or knobs, a touch-sensitive surface such as a touchscreen display, or other input sources). In some embodiments, display 547 is a touch screen display that is capable of receiving user touch inputs. Communication buses 552 optionally include circuitry (sometimes called a chipset) that interconnects and controls communications between system components. Memory 554 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 554 optionally includes one or more storage devices remotely located from the CPU(s) 542. Memory 552, or alternately the non-volatile memory device(s) within memory 552, comprises a non-transitory computer readable storage medium. In some embodiments, memory 552, or the computer

readable storage medium of memory 552 stores the following programs, modules, and data structures, or a subset thereof:

- an operating system 556 that includes procedures for handling various basic system services and for performing hardware dependent tasks; and
- a network communication module 558 that is used for connecting (wired or wireless) server 540 to other computing devices via the one or more communication network interfaces 550 and one or more communication networks 505, such as the Internet, other wide area networks, local area networks, metropolitan area networks, and so on.

[0129] In some embodiments, device 100 drives display 547 of system 540. For example, device 100 sends a video signal to system 540, and CPU 542 of system 540 renders the video signal on display 547. In some embodiments, device 100 sends a video signal directly to display 547 and CPU 542 is not used to render the video signal (e.g., device 100 uses display 547 as an auxiliary display). In some embodiments, the user interface displayed on touch screen 112 of device 100 is synchronized with the user interface displayed on display 547 of system 540, and, in some other embodiments, the user interface displayed on touch screen 112 of device 100 is not synchronized with the user interface displayed on display 547 of system 540. In some embodiments, system 540 sends information corresponding to a user input (e.g., a user touch input on display 547 or a user input via controls 548) to device 100, and device 100 updates the user interface displayed on touch screen 112 of device 100 in accordance with the received information.

USER INTERFACES AND ASSOCIATED PROCESSES

[0130] Attention is now directed towards embodiments of user interfaces ("UI") and associated processes that may be implemented on an electronic device with a display and a touch-sensitive surface, such as device 300 or portable multifunction device 100.

[0131] Figures 6A-6C illustrate exemplary user interfaces generated from a template for use with third-party applications, in accordance with some embodiments. The user interfaces in these figures are used to illustrate the processes described below, including the processes in Figures 7A-7I.

[0132] Figure 6A illustrates user-interface template 601 displayed on a display 547 of external information presentation system 540. In some embodiments, system 540 is included in a vehicle (e.g., in the dashboard or steering wheel of the vehicle). In some

embodiments, display 547 is a touch screen display configured to receive user touch input. Figure 6A further illustrates a plurality of user interface objects (sometimes herein called "affordances" or "selectable user interface objects") including application title bar 602, default main menu 604, custom menu 608, content selections view 606, active item bar 610, and exit button 612. In some embodiments, the user-interface template contains fewer affordances than shown in template 601, to further simplify the user interface and prevent driver distraction.

[0133] Figure 6B illustrates an example of a user interface generated from user-interface template 601 (e.g., for use with an internet radio application). Figure 6B illustrates application title bar 602 displaying the title of the current application (e.g., Internet Radio App), main menu 604 with default controls (e.g., up, down, search, sort, etc.), custom menu 608 (e.g., Favorites menu) with application-specific controls (e.g., filter favorites, add current to favorites that correspond to application-specific functions), and exit button 612. Figure 6B further illustrates content selections view 606 (e.g., station selections) displaying selection options for the current application (e.g., smooth jazz station 103, Korean pop station 105, etc.) and active item bar 610 displaying information regarding the currently active item (e.g., hard rock station 101).

[0134] Figure 6C illustrates an example of a user interface generated from user-interface template 601 (e.g., for use with a music library application) that is different from the user interface generated from template 601 that is shown in Figure 6B. Figure 6C illustrates application title bar 602 displaying the title of the current application (e.g., Music Library App), main menu 604 with default controls (e.g., up, down, search, sort, etc.), custom menu 608 (e.g., User Playlists menu) with application-specific controls (e.g., playlist 1, playlist 2, etc.), and exit button 612. Figure 6C further illustrates content selections view 606 (e.g., song library) displaying selection options for the current application (e.g., stored song 103, stored song 105, etc.) and active item bar 610 displaying information regarding the currently active item (e.g., stored song 101).

[0135] As shown in Figures 6B-6C, a common template (e.g., the template 601 shown in Figure 6A) is, optionally, used to generate different user interfaces for different third-party applications (e.g., the user interface for the internet radio application shown in Figure 6B is different from the user interface for the music library application shown in Figure 6C) by inserting information retrieved from the different third-party applications into user-interface template 601. The different user interfaces shown in Figures 6B and

6C, however, share similarities that make them easier to use (e.g., affordances for selecting playable content are displayed in content selections view 606 in both of the user interfaces in Figures 6B and 6C and information about the currently playing media is displayed in the active item bar 610 in both of the user interfaces in Figures 6B and 6C). Additionally, the template optionally enforces a minimum text size to ensure that text displayed in the user interfaces in Figures 6B and 6C is easy to read.

[0136] Figures 7A-7I are flow diagrams illustrating a method 700 of providing user interfaces for third-party applications in accordance with some embodiments. The method 700 is performed at an electronic device (e.g., device 300, Figure 3, or portable multifunction device 100, Figure 1A). In some embodiments, the electronic device includes a display and a touch-sensitive surface. In some embodiments, the display is a touch screen display and the touch-sensitive surface is on the display. In some embodiments, the display is separate from the touch-sensitive surface. Some operations in method 700 are, optionally, combined and/or the order of some operations is, optionally, changed.

[0137] As described below, the method 700 provides user interfaces for third-party applications that reduce the cognitive burden on a user when interacting with third party applications, thereby creating a more efficient human-machine interface. For battery-operated electronic devices, enabling a user to interact with the user interfaces faster and more efficiently conserves power and increases the time between battery charges.

[0138] The device receives (702) a first display request to display a user interface of a first third-party application on a respective display that is in communication (e.g., via a wired or wireless connection) with the electronic device. For example, the device receives an input that corresponds to activation of an application launch icon for the first third-party application from a control associated with the respective display or a control associated with the electronic device (e.g., a tap gesture on Internet radio icon 442 (Figure 4A) for Internet radio application 142 (Figure 1A or 3), or a tap gesture on Music Library icon 432 (Figure 4A) for Music Library application 155 (Figure 1A or 3). In some embodiments, the respective display is a display of the device (e.g., touchscreen 112 of portable multifunction device 100). In some embodiments, the respective display is a display that is separate from the device (e.g., display 547 of external information presentation system 540 in Figure 5).

[0139] In response to receiving the first display request (704), the device obtains (706) a first user-interface template that is configured (or designed) to be used by a plurality of third-party applications; and the device requests (710), from the first third-party application, one or more values for populating the first user-interface template. In some embodiments, the first user-interface template is a generic or application-agnostic (or independent) user-interface template that is used by a plurality of different applications. For example, in some embodiments, the device obtains user-interface template 601 shown in Figure 6A.

[0140] In some embodiments, the respective display is (708) a display in a vehicle information display system that is visible from a driver seat of a vehicle; and the first user-interface template is one of a plurality of predetermined user-interface templates that are configured to be used to generate user interfaces for the third-party application on the respective display that prevents distraction of the driver (e.g., the plurality of predetermined templates have large font and a few easily selectable options that can be operated by a driver without distracting the driver from driving safely). Thus, in some embodiments, using these predetermined user-interface templates enables the device to provide the user with access to the functionality of one or more third party applications while maintaining some control over the appearance of the user interface that is displayed on the respective display, so as to ensure that the user interface on the respective display is simple and intuitive to use.

[0141] In some embodiments, values at the first third-party application are arranged (711) in a tree structure that includes a plurality of nodes that have predetermined properties and a tree relationship (e.g., a parent relationship or a child relationship) with one or more other nodes in the tree structure. In some embodiments, the request for one or more values for populating the first user-interface template includes one or more of: a request to return values that correspond to a list of child nodes of a respective node of the plurality of nodes (e.g., for a node that corresponds to an artist, the child nodes correspond to songs by the artist or albums by the artist); a request to return values that correspond to a list of properties of a respective node of the plurality of nodes; a request to provide content that is represented by the respective node (e.g., a request to provide audio content that corresponds to an audio file, an audio stream, a video file, a video stream and/or a playlist or radio station that corresponds to a plurality of distinct units of content); a request to provide a list of nodes that match a search query (e.g., a request to provide a list

of artists, albums, songs, radio stations, audio books, podcasts or the like that match a set of one or more search terms and, optionally, logical operators connecting the search terms); and a request to provide a list of actions specific to the first third-party application that can be taken while providing access to content associated with the first third-party application (e.g., "like" "skip" "add to favorites" "flag" or other application-specific operations). In some embodiments, the properties of a respective node include one or more of: artwork for the respective node, an identifier for the respective node, a duration of media content associated with the respective node, a title of the respective node (a title of a currently playing track or radio station), a subtitle of the respective node (e.g., an artist of a currently playing song or a track of a currently playing radio station), a playable flag indicating whether or not the respective node is playable, a played flag indicating whether or not the respective node has already been played, and/or a container flag indicating whether or not the respective node is a container that has child nodes.

[0142] The device receives (712), from the first third-party application, while the first third-party application is running on the electronic device, a first set of values for populating the first user-interface template (e.g., receiving a list of names of internet streaming radio stations). For example, Figure 6B shows user-interface template 601 populated with data from an application titled "Internet Radio App."

[0143] In some embodiments, one or more of the values in the first set of values are retrieved (713) from a remote source (e.g., a server in communication with the first third-party application such as sever 510 in Figure 5) by the first third-party application in response to the request for one or more values for populating the first user-interface template. For example, when a streaming internet radio station has been selected for playing, the first third-party application initiates sending to a remote server a request to start streaming the streaming internet radio station and requests a name of a song and artist and corresponding album artwork for a song to be played on the streaming internet radio station. The first third-party application receives these values and transmits these values to an intermediary process for incorporation into a "now playing" user interface template that the intermediary process will transmit to the respective display.

[0144] In some embodiments, the device includes (714) a digital personal assistant process with voice recognition capabilities (e.g., a digital personal assistant 150 that records voice commands and sends information representative of the recorded voice commands to a server such as server 510 in Figure 5 for analysis, and responds to the

voice commands based on a response from the server). In some embodiments, the device determines (715) the capabilities of the first third-party application based on responses from the first third-party application to a plurality of requests to the third-party application for values associated with populating user-interface templates for user interfaces for the respective display (e.g., the digital personal assistant process traverses a tree structure of metadata for the first third- party application and maps out the various options and metadata that are accessible via the first third-party application). In some embodiments, the device generates (716) an index of the capabilities of the first third-party application (e.g., based on the map of the various options and metadata that are accessible via the first third-party application). In some embodiments, the device provides (718) voice access to the capabilities of the first third- party application via the digital personal assistant process (e.g., enabling a user to use voice commands to request display of a list of the songs by a particular artist, display of a list of internet streaming radio stations that are available for playing, playback of a particular song by a particular artist, or searching through a media presentation application for media that matches a verbally stated search query). For example, for a streaming internet radio application, the personal assistant would traverse and index the tree structure of the metadata for available streaming internet radio stations and other operations that can be performed by the streaming internet radio application. Then, the user is enabled to provide verbal commands to the digital personal assistant to start playing a radio station, stop playing a radio station, create a new radio station based on a current song, skip a song or other operation using voice commands directed to the digital personal assistant (e.g., "Assistant, please play Artist X radio station," "Assistant, please skip this song," "Assistant please list all radio stations created in the last week," or "Assistant, please mark this song as a favorite.")

[0145] In some embodiments, the device detects (720) that a predetermined condition has been met. In response to detecting that the predetermined condition has been met, the device re-determines (721) the capabilities of the first third-party application and re-generates the index of the capabilities of the first third-party application (e.g., the various options that are accessible via the first third-party application are re-indexed: when the device is placed in communication with the respective display (such as display 547 of external information presentation system 540 (Figure 5)), when the device is connected to a vehicle information display system that includes the respective display, and/or when the first third-party application is launched). For example, for a streaming internet radio

application, the availability of streaming internet radio stations is, optionally, determined based on the availability of an internet connection for receiving the streaming internet radio station; similarly, if the user creates/subscribes to a new radio station or deletes/unsubscribes to an old radio station, the metadata associated with the streaming internet radio application changes (e.g., which radio stations are nodes in the tree structure and/or which radio stations are playable). In order to address the changing metadata at the streaming internet radio application, the device periodically re-indexes the metadata at the streaming internet radio application (e.g., when the streaming internet radio application is launched or when the device is connected to a vehicle information display system). In some embodiments, the third-party application proactively informs the digital personal assistant that some portion of the metadata has been changed or updated and prompts the digital personal assistant to re-index that portion (or all) of the metadata of the third-party application. The digital personal assistant re-indexes that portion (or all) of the metadata of the third-party application in response to receiving the information that the portion of the metadata has been changed or the next time that the third-party application is launched.

[0146] The device populates (722) the first user-interface template with the first set of values received from the first third-party application. For example, in accordance with these embodiments, Figure 6B shows user-interface template 601 populated with data from a third-party application titled "Internet Radio App" (e.g., station selections, favorites, current station, etc.) to generate the user interface for the third-party application that is displayed on Display 547 in Figure 6C.

[0147] In some embodiments, the first user-interface template is used by a plurality of different third-party applications including (724) the first third-party application and a second third-party application. In some embodiments, populating the first user-interface template with values received from a respective third-party application includes inserting values identified by the respective third-party application as belonging to a first class of values in a predefined region at a predefined font size in a user interface that is provided to the respective display. For the first user-interface template, the first third-party application identifies a first type of information as the first class of values, and the second third-party application identifies a second type of information, different from the first type of information, as the first class of values. For example, Figure 6B shows user-interface template 601 populated with values related to internet radio stations and Figure 6C shows

user-interface template 601 populated with values related to stored songs. In accordance with these embodiments, Figures 6B and 6C show user-interface templates with the same font sizes.

[0148] In some embodiments, for the first user-interface template: the first third- party application identifies (726) the names of remotely-generated content stations (e.g., streaming internet radio stations) as the first class of values; and the second third-party application identifies the names of predetermined playlists as the first class of values. In some embodiments, a predetermined playlist is different from a streaming internet radio station because a predetermined playlist has a set length and the contents of the playlist are set at the time that the user selects the playlist for playing, whereas a streaming internet radio station is a continuous stream of music of indeterminate length that continues for as long as the user is listening and includes content selected remotely (e.g., automatically by a computer in accordance with music-selection algorithms or manually by a disc jockey, radio host or other person independent of actions by a user of the device). For example in Figures 6B-6C, the device identifies the names of streaming internet radio stations as the first class of values (e.g., descriptions of playable media content items) for the Internet radio application (e.g., 142, Figure 1A) in Figure 6B and the device identifies the names of songs as the first class of values for the music library application (e.g., 155, Figure 1A) in Figure 6C.

[0149] In some embodiments, for the first user-interface template: the first third- party application identifies (728) the names of content that can be streamed to the device (e.g., on-demand streaming of music or other media content over a distributed data network such as the Internet) as the first class of values; and the second third-party application identifies the names of content stored on the device (e.g., language learning programs, audio books, music, podcasts or the like) as the first class of values.

[0150] The device generates (730) a first user interface for the first third-party application using the first user-interface template populated with the first set of values. For example, Figure 6B illustrates a user interface generated from user-interface template 601 and values received from an internet radio application. In accordance with this example, Figure 6B shows default main menu 604 including an info button (e.g., to return values that correspond to a list of properties of a respective node), an activate button (e.g., to provide content that is represented by the respective node), a search button (e.g., to provide a list of nodes that match a search query), and custom menu 608 with custom

buttons (e.g., to provide a list of actions specific to the first third-party application that can be taken while providing access to content associated with the first third-party application).

[0151] In some embodiments, the device includes (732) a plurality of processes including a first third-party application process for running the first third-party application and a set of one or more intermediation processes (e.g., an application for integrating third- party applications with a vehicle information display system such as a navigation or entertainment system or other external information display system, like external information presentation system 540 in Figure 5) for relaying information between the respective display and the first-third party application process. In some embodiments, the request of the one or more values for populating the first user-interface template is generated by an intermediation process from the set of one or more intermediation processes; and the first user interface is generated by an intermediation process from the set of one or more intermediation processes. For example, a smart phone (e.g., portable multifunction device 100 in Figure 5) has a vehicle integration application 151(e.g., an application including one or more intermediation processes) that controls a vehicle information display system in the vehicle (e.g., a car, a truck, a van, etc.) that provides a user interface on a respective display of the vehicle information display system (e.g., display 547 of external information presentation system 540 in Figure 5), such as for a mapping application or a music application. The smart phone also has one or more third-party applications such as a separate podcasting application or a streaming internet radio application and the vehicle integration application converts information from the third-party applications into content for display by the vehicle integration application on the respective display of the vehicle information display system.

[0152] In some embodiments, the first third-party application is (734) a media presentation application (e.g., Internet radio 142, Figure 1A). In some embodiments, the device includes a first-party application that is a media presentation application (e.g., video & music player module 152, Figure 1A); the first party application includes a first-party user interface that has a plurality of selectable affordances for requesting performance of media presentation operations and one or more content presentation regions (e.g., a region for displaying a list of available media) at respective locations in the first-party user interface; and generating the first user interface for the first third-party application includes generating a user interface that includes selectable affordances and

one or more content presentation regions that correspond in function and location to the selectable affordances and content presentation regions of the first-party user interface. Thus, in some embodiments, the first user-interface template is a template that conforms information from third-party applications into a standardized user interface that has similar controls, content regions, and appearance to a first-party media presentation application so as to provide a consistent user interface that is familiar to the user and thus prevents driver distraction when displayed on a vehicle information display system that is visible from a driver seat of a vehicle.

[0153] The device sends (736), or otherwise provides, information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display (e.g., portable multifunction device 100 provides the information to display 547 of external information presentation system 540 in Figure 5).

[0154] In some embodiments, while the first user-interface for the first third-party application is displayed on the respective display, the device receives (738), from an input device for the respective display (e.g., a touch-sensitive surface that is co-incident with the respective display or a button or knob that is used for controlling the respective display) a respective input that corresponds to selection of an option in the first user interface (e.g., the selected option is an option to play, pause, seek, display a new menu or perform a third- party application specific operation such as "liking," "skipping" or "add to favorites"). In response to receiving the respective input, the device provides (740) information to the first third-party application running on the electronic device that enables the first third-party application to respond to selection of the option in the first user interface. In some embodiments, the device receives (742), from the first third-party application, the response to selection of the option from the first third-party application; and sends (744), or otherwise provides, information to the respective display that enables the response to selection of the option to be provided (e.g., displayed) to the user. For example, the user is presented with a list of names of streaming internet radio stations on the respective display, the user taps on a name of a respective streaming internet radio station (e.g., on a touchscreen display of external information presentation system 540 in Figure 5), an intermediation process at the device (e.g., portable multifunction device 100 in Figure 5) provides the first third-party application with information indicating that the respective streaming internet radio station has been selected, the first third-party application requests that a server (e.g., server 510 in Figure 5) provide artwork, a title,

subtitle and audio stream for the respective streaming internet radio station and then provides this information to the intermediation process (e.g., at portable multifunction device 100 in Figure 5) which generates a user interface based on the information and a user interface template and transmits the generated user interface to the respective display for display (e.g., on display 547 of external information presentation system 540 in Figure 5).

[0155] In some embodiments, a first-party intermediation process at the device intermediates communication between the respective display and a first third-party application by converting data from the first third-party application into a standardized, simplified, user interface that prevents distracted driving and by relaying inputs that correspond to user interaction with the user interface displayed on the respective display so that the first third-party application can respond to the inputs, even when the first third-party application is not provided with information indicative of a size, resolution and other characteristics of the respective display and does not have direct access to user interactions with the user interface displayed on the respective display. For example, the third-party application is not provided with times and locations of touch inputs but rather gets information about what information was requested by the user as filtered through the first- party intermediation process.

[0156] In some embodiments, while the first user interface for the first third-party application is displayed on the respective display (746), the device receives (748) information indicative of a selection of a respective option in the first user interface (e.g., selection of a container or a node with child nodes, such as an "artist" folder that includes songs by the selected artist or selection of a particular internet streaming radio station of a plurality of internet streaming radio stations). In some embodiments, in response to receiving the information indicative of the selection of the respective option in the first user interface (750): the device obtains (752) a second user-interface template, different from the first user-interface template (e.g., a template for displaying the contents of a container, such as a playlist template or a "now playing" user interface template); and requests (754), from the first third-party application, one or more values for populating the second user-interface template. In some embodiments, the device receives (756), from the first third-party application, while the first third-party application is running on the electronic device, a second set of values for populating the second user-interface template (e.g., names and playtime information for the songs by the selected artist for a playlist

template or a name of a currently playing radio station, a name of a currently playing song, artwork for the currently playing song for a "now playing" template); populates (758) the second user- interface template with the second set of values received from the first third-party application; generates (760) a second user interface for the first third-party application by using the second user-interface template populated with the second set of values (e.g., inserting the names and playtime information for the songs by the selected artist into the playlist template or inserting a name of a currently playing radio station, a name of a currently playing song, and artwork for the currently playing song into a "now playing" template that includes controls for controlling pausing/starting/stopping/fast forwarding/rewinding the currently playing song); and sends (762), or otherwise provides, information to the respective display that enables the second user interface for the first third-party application to be displayed on the respective display.

[0157] In some embodiments, after sending the information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display (764), the device receives (766) a second display request to display a user interface for a second third-party application, different from the first third-party application, on the respective display (e.g., receiving activation of an application launch icon for the second third-party application using a control associated with the respective display or a control associated with the electronic device). For example, while the user was initially playing audio from an Internet streaming radio application 142 (e.g., using the user interface for the streaming internet radio application shown in Figure 6B), the user switches to playing audio from a separate podcast application or an audio book application. In some embodiments, in response to receiving the second display request, the device requests (768), from the second third-party application, one or more values for populating the first user- interface template; receives (770), from the second third-party application, while the second third-party application is running on the electronic device, a third set of values for populating the first user-interface template; populates (772) the first user-interface template with the third set of values received from the second third-party application; generates (774) a second user interface for the second third-party application using the first user-interface template populated with the third set of values; and sends (776) (or otherwise provides) information to the respective display that enables the second user interface for the second third-party application to be displayed on the respective display (e.g., the user interface for the music library application 155 shown in Figure 6C).

Thus, in some embodiments, the device repeats the process of requesting values from a third party application to fill out the first user-interface template when a different third party application is requested that uses the same first user-interface template. This process is repeated in this situation because the first user-interface template is designed or configured to be used by a plurality of different third-party applications.

[0158] In some embodiments, after sending the information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display (764): the device receives (778), from the first third-party application, an updated first set of values for populating the first user-interface template. In some embodiments, in response to receiving the updated first set of values for populating the first user-interface template (780), the device populates (782) the first user-interface template with the updated first set of values received from the first third-party application; generates (784) an updated first user interface for the first third-party application using the first user- interface template populated with the updated first set of values; and provides (786) information to the respective display that enables the updated first user interface for the first third-party application to be displayed on the respective display. In some embodiments, the updated first set of values is generated based on events that occur at the first third-party application (e.g., the update to the user interface for the first third-party application is initiated by the first third-party application rather than being initiated by an intermediation process). For example, the third-party application is a streaming media application and the data connection of the device ceases to function. In response to detecting that the data connection has ceased to function, the third-party application transmits information to an intermediation process at the device indicating that streaming content that was previously marked as "playable" (e.g., when there was a working data connection) has changed in status so that it is "not playable" (e.g., because the data connection is no longer available). In some embodiments, when content changes from being "playable" to "not playable," the first user interface for the first third-party application is updated to indicate that the content is "not playable" (e.g., by removing the content from a list of playable content or changing an appearance of a representation of the content to indicate that the content is not currently playable by reducing a contrast of the representation or displaying a "not playable" icon or symbol next to the representation of the content). For example, in Figure 6B, if the device were to cease to have access to the streaming internet radio stations shown in content selections view 606, the device

would regenerate the user interface shown in Figure 6B to reflect the unavailability of the streaming internet radio stations (e.g., by displaying a not- playable icon adjacent to the names of the unavailable stations, by reducing the contrast of ("graying out") the unavailable stations or by removing the stations from the content selections view 606).

[0159] It should be understood that the particular order in which the operations in Figures 7A-7I have been described is merely an example and is not intended to indicate that the described order is the only order in which the operations could be performed. One of ordinary skill in the art would recognize various ways to reorder the operations described herein.

[0160] In accordance with some embodiments, Figure 8 shows a functional block diagram of a first electronic device 800 configured in accordance with the principles of the various described embodiments. The functional blocks of the device are, optionally, implemented by hardware, software, or a combination of hardware and software to carry out the principles of the various described embodiments. It is understood by persons of skill in the art that the functional blocks described in Figure 8 are, optionally, combined or separated into sub-blocks to implement the principles of the various described embodiments. Therefore, the description herein optionally supports any possible combination or separation or further definition of the functional blocks described herein.

[0161] As shown in Figure 8, electronic device 800 includes memory coupled to processing unit 802. In some embodiments, electronic device 800 also includes communications interface unit 804 coupled to processing unit 802 and configured to communicate with respective display unit 806. In some embodiments, processing unit 802 includes receiving unit 807, obtaining unit 808, requesting unit 810, populating unit 812, interface generating unit 814, sending unit 816, information providing unit 818, determining unit 820, index generating unit 822, voice providing unit 824, detecting unit 826, and digital assistant unit 828. Processing unit 802 is configured to receive (e.g., with receiving unit 807 via communications interface unit 804) a first display request to display a user interface of a first third-party application on a respective display unit (e.g., with respective display unit 806) that is in communication with electronic device 800. Processing unit 802 is further configured to, in response to receiving the first display request: obtain (e.g., with obtaining unit 808) a first user-interface template that is configured to be used by a plurality of third- party applications; and request (e.g., with requesting unit 810), from the first third-party application, one or more values for

populating the first user-interface template. Processing unit 802 is further configured to receive (e.g., with receiving unit 807 or communications interface unit 804), from the first third-party application, while the first third-party application is running on the electronic device, a first set of values for populating the first user-interface template; populate (e.g., with populating unit 812) the first user-interface template with the first set of values received from the first third-party application; generate (e.g., with interface generating unit 814) a first user interface for the first third-party application using the first user-interface template populated with the :first set of values; and send (e.g., with sending unit 816) information to the respective display unit that enables the first user interface for the first third-party application to be displayed on the respective display unit (e.g., respective display unit 806).

[0162] In some embodiments, processing unit 802 is further configured to: while the first user-interface for the first third-party application is displayed on the respective display unit (e.g., respective display unit 806), receive (e.g., with receiving unit 807 via communications interface unit 804), from an input device for the respective display unit a respective input that corresponds to selection of an option in the first user interface; and, in response to receiving the respective input, provide (e.g., with information providing unit 818) information to the first third-party application running on the electronic device that enables the first third-party application to respond to selection of the option in the first user interface. In some embodiments, processing unit 802 is further configured to receive (e.g., with receiving unit 807 via communications interface unit 804), from the first third-party application, the response to selection of the option from the first third-party application; and send (e.g., with sending unit 816) information to the respective display unit that enables the response to selection of the option to be provided to the user.

[0163] In some embodiments, processing unit 802 is further configured to, while the first user interface for the first third-party application is displayed on the respective display unit (e.g., respective display unit 806): receive (e.g., with receiving unit 807 via communications interface unit 804) information indicative of a selection of a respective option in the first user interface; and, in response to receiving the information indicative of the selection of the respective option in the first user interface: obtain (e.g., with obtaining unit 808) a second user-interface template, different from the first user-interface template; and request (e.g., with requesting unit 810), from the first third-party application, one or more values for populating the second user-interface template. In some

embodiments, processing unit 802 is further configured to receive (e.g., with receiving unit 807 via communications interface unit 804), from the first third-party application, while the first third-party application is running on the electronic device, a second set of values for populating the second user-interface template; populate (e.g., with populating unit 812) the second user-interface template with the second set of values received from the first third- party application; generate (e.g., with interface generating unit 814) a second user interface for the first third-party application by using the second user-interface template populated with the second set of values; and send (e.g., with sending unit 816) information to the respective display unit that enables the second user interface for the first third-party application to be displayed on the respective display unit (e.g., respective display unit 806).

[0164] In some embodiments, processing unit 802 is further configured to, after sending (e.g., with sending unit 816) the information to the respective display unit that enables the first user interface for the first third-party application to be displayed on the respective display unit (e.g., respective display unit 806) receive (e.g., with receiving unit 807 or communications interface unit 804) a second display request to display a user interface for a second third-party application, different from the first third-party application, on the respective display unit (e.g., respective display unit 806). In some embodiments, processing unit 802 is further configured to, in response to receiving the second display request, request (e.g., with requesting unit 810), from the second third-party application, one or more values for populating the first user-interface template; receive (e.g., with receiving unit 807 via communications interface unit 804), from the second third-party application, while the second third-party application is running on the electronic device, a third set of values for populating the first user-interface template; populate (e.g., with populating unit 812) the first user-interface template with the third set of values received from the second third-party application; generate (e.g., with interface generating unit 814) a second user interface for the second third-party application using the first user-interface template populated with the third set of values; and send (e.g., with sending unit 816) information to the respective display unit that enables the second user interface for the second third-party application to be displayed on the respective display unit (e.g., respective display unit 806).

[0165] In some embodiments, the device includes a plurality of processes including a first third-party application process for running the first third-party application and a set of

one or more intermediation processes for relaying information between the respective display unit and the first-third party application process; the request of the one or more values for populating the first user-interface template is generated by an intermediation process from the set of one or more intermediation processes; and the first user interface is generated by an intermediation process from the set of one or more intermediation processes.

[0166] In some embodiments, one or more of the values in the first set of values are retrieved from a remote source by the first third-party application in response to the request for one or more values for populating the first user-interface template.

[0167] In some embodiments, the respective display unit (e.g., respective display unit 806) is a display unit in a vehicle information display system that is visible from a driver seat of a vehicle; and the first user-interface template is one of a plurality of predetermined user-interface templates that are configured to be used to generate user interfaces for the third-party application on the respective display unit that prevents distraction of the driver.

[0168] In some embodiments, the first user-interface template is used by a plurality of different third-party applications including the first third-party application and a second third-party application; populating (e.g., with populating unit 812) the first user-interface template with values received from a respective third-party application includes inserting values identified by the respective third-party application as belonging to a first class of values in a predefined region at a predefined font size in a user interface that is provided to the respective display unit; and, for the first user-interface template: the first third-party application identifies a first type of information as the first class of values; and the second third-party application identifies a second type of information, different from the first type of information, as the first class of values.

[0169] In some embodiments, for the first user-interface template: the first third-party application identifies the names of remotely-generated content stations as the first class of values; and the second third-party application identifies the names of predetermined playlists as the first class of values.

[0170] In some embodiments, for the first user-interface template: the first third-party application identifies the names of content that can be streamed to the device as the first class of values; and the second third-party application identifies the names of content stored on the device as the first class of values.

[0171] In some embodiments, values at the first third-party application are arranged in a tree structure that includes a plurality of nodes that have predetermined properties and a tree relationship with one or more other nodes in the tree structure; and the request (e.g., with requesting unit 810) for one or more values for populating the first user-interface template includes one or more of: a request to return values that correspond to a list of child nodes of a respective node of the plurality of nodes; a request to return values that correspond to a list of properties of a respective node of the plurality of nodes; a request to provide content that is represented by the respective node; a request to provide a list of nodes that match a search query; and a request to provide a list of actions specific to the first third-party application that can be taken while providing access to content associated with the first third-party application.

[0172] In some embodiments, the first third-party application is a media presentation application. In some embodiments, device 800 includes a first-party application that is a media presentation application; the first party application includes a first-party user interface that has a plurality of selectable affordances for requesting performance of media presentation operations and one or more content presentation regions at respective locations in the first-party user interface; and generating (e.g., with interface generating unit 814) the first user interface for the first third-party application includes generating a user interface that includes selectable affordances and one or more content presentation regions that correspond in function and location to the selectable affordances and content presentation regions of the first-party user interface.

[0173] In some embodiments, processing unit 802 is further configured to, after sending the information to the respective display unit that enables the first user interface for the first third-party application to be displayed on the respective display unit (e.g., respective display unit 806): receive (e.g., with receiving unit 807 via communications interface unit 804), from the first third-party application, an updated first set of values for populating the first user-interface template; and, in response to receiving the updated first set of values for populating the first user-interface template: populate (e.g., with populating unit 812) the first user-interface template with the updated first set of values received from the first third-party application; generate (e.g., with interface generating unit 814) an updated first user interface for the first third-party application using the first user-interface template populated with the updated first set of values; and provide (e.g., with information providing unit 818) information to the respective display unit that

enables the updated first user interface for the first third-party application to be displayed on the respective display unit.

[0174] In some embodiments, device 800 includes a digital personal assistant process (e.g., digital assistant unit 828) with voice recognition capabilities. In some embodiments, processing unit 802 is further configured to: determine (e.g., with determining unit 820) the capabilities of the first third-party application based on responses from the first third-party application to a plurality of requests to the third-party application for values associated with populating user-interface templates for user interfaces for the respective display unit; generate (e.g., with index generating unit 822) an index of the capabilities of the first third-party application; and provide (e.g., with voice providing unit 824) voice access to the capabilities of the first third-party application via the digital personal assistant process (e.g., with digital assistant unit 828).

[0175] In some embodiments, processing unit 802 is further configured to: detect (e.g., with detecting unit 826) that a predetermined condition has been met; and, in response to detecting that the predetermined condition has been met: re-determine (e.g., with determining unit 820) the capabilities of the first third-party application; and re-generate (e.g., with index generating unit 822) the index of the capabilities of the first third-party application.

[0176] The operations in the information processing methods described above are, optionally, implemented by running one or more functional modules in information processing apparatus such as general purpose processors (e.g., as described above with respect to Figures 1A and 3) or application specific chips.

[0177] The operations described above with reference to Figures 7A-7I are, optionally, implemented by components depicted in Figures 1A-1B or Figure 8. For example, receiving operations 702 and 712, requesting operation 710, populating operation 722, generating operation 730, and sending operation 736 are, optionally, implemented by event sorter 170, event recognizer 180, and event handler 190. Event monitor 171 in event sorter 170 detects a contact on touch-sensitive display 112, and event dispatcher module 174 delivers the event information to application 136-1. A respective event recognizer 180 of application 136-1 compares the event information to respective event definitions 186, and determines whether a first contact at a first location on the touch-sensitive surface (or whether orientation of the device) corresponds to a predefined event or sub-event, such as selection of an object on a user interface, or

rotation of the device from one orientation to another. When a respective predefined event or sub-event is detected, event recognizer 180 activates an event handler 190 associated with the detection of the event or sub-event. Event handler 190 optionally uses or calls data updater 176 or object updater 177 to update the application internal state 192. In some embodiments, event handler 190 accesses a respective GUI updater 178 to update what is displayed by the application. Similarly, it would be clear to a person having ordinary skill in the art how other processes can be implemented based on the components depicted in Figures 1A-1B.

[0178] 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 invention and its practical applications, to thereby enable others skilled in the art to best use the invention and various described embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A method, comprising:

at an electronic device with one or more processors and memory:

receiving a first display request to display a user interface of a first third-party application on a respective display that is in communication with the electronic device;

in response to receiving the first display request:

obtaining a first user-interface template that is configured to be used by a plurality of third-party applications; and

requesting, from the first third-party application, one or more values for populating the first user-interface template;

receiving, from the first third-party application, while the first third-party application is running on the electronic device, a first set of values for populating the first user-interface template;

populating the first user-interface template with the first set of values received from the first third-party application;

generating a first user interface for the first third-party application using the first user-interface template populated with the first set of values; and

sending information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display.

2. The method of claim 1, including:

while the first user-interface for the first third-party application is displayed on the respective display, receiving, from an input device for the respective display a respective input that corresponds to selection of an option in the first user interface; and

in response to receiving the respective input, providing information to the first third-party application running on the electronic device that enables the first third-party application to respond to selection of the option in the first user interface;

receiving, from the first third-party application, the response to selection of the option from the first third-party application; and

sending information to the respective display that enables the response to selection of the option to be provided to the user.

3. The method of any of claims 1-2, including, while the first user interface for the first third-party application is displayed on the respective display:

receiving information indicative of a selection of a respective option in the first user interface; and

in response to receiving the information indicative of the selection of the respective option in the first user interface:

obtaining a second user-interface template, different from the first user-interface template; and

requesting, from the first third-party application, one or more values for populating the second user-interface template;

receiving, from the first third-party application, while the first third-party application is running on the electronic device, a second set of values for populating the second user-interface template;

populating the second user-interface template with the second set of values received from the first third-party application;

generating a second user interface for the first third-party application by using the second user-interface template populated with the second set of values; and

sending information to the respective display that enables the second user interface for the first third-party application to be displayed on the respective display.

4. The method of any of claims 1-3, including, after sending the information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display:

receiving a second display request to display a user interface for a second third-party application, different from the first third-party application, on the respective display;

in response to receiving the second display request, requesting, from the second third-party application, one or more values for populating the first user-interface template;

receiving, from the second third-party application, while the second third-party application is running on the electronic device, a third set of values for populating the first user-interface template;

populating the first user-interface template with the third set of values received from the second third-party application;

generating a second user interface for the second third-party application using the first user-interface template populated with the third set of values; and

sending information to the respective display that enables the second user interface for the second third-party application to be displayed on the respective display.

5. The method of any of claims 1-4, wherein:

the device includes a plurality of processes including a first third-party application process for running the first third-party application and a set of one or more intermediation processes for relaying information between the respective display and the first-third party application process;

the request for the one or more values for populating the first user-interface template is generated by an intermediation process from the set of one or more intermediation processes; and

the first user interface is generated by an intermediation process from the set of one or more intermediation processes.

6. The method of any of claims 1-5, wherein one or more of the values in the first set of values are retrieved from a remote source by the first third-party application in response to the request for one or more values for populating the first user-interface template.

7. The method of any of claims 1-6, wherein:

the respective display is a display in a vehicle information display system that is visible from a driver seat of a vehicle; and

the first user-interface template is one of a plurality of predetermined user-interface templates that are configured to be used to generate user interfaces for the first third-party application on the respective display that prevents distraction of the driver.

8. The method of any of claims 1-7, wherein:

the first user-interface template is used by a plurality of different third-party applications including the first third-party application and a second third-party application;

populating the first user-interface template with values received from a respective third-party application includes inserting values identified by the respective third-party application as belonging to a first class of values in a predefined region at a predefined font size in a user interface that is provided to the respective display; and

for the first user-interface template:

the first third-party application identifies a first type of information as the first class of values; and

the second third-party application identifies a second type of information, different from the first type of information, as the first class of values.

9. The method of claim 8, wherein, for the first user-interface template:

the first third-party application identifies the names of remotely-generated content stations as the first class of values; and

the second third-party application identifies the names of predetermined playlists as the first class of values.

10. The method of claim 8, wherein, for the first user-interface template:

the first third-party application identifies the names of content that can be streamed to the device as the first class of values; and

the second third-party application identifies the names of content stored on the device as the first class of values.

11. The method of any of claims 1-10, wherein:

values at the first third-party application are arranged in a tree structure that includes a plurality of nodes that have predetermined properties and a tree relationship with one or more other nodes in the tree structure; and

the request for one or more values for populating the first user-interface template includes one or more of:

a request to return values that correspond to a list of child nodes of a respective node of the plurality of nodes;

a request to return values that correspond to a list of properties of a respective node of the plurality of nodes;

a request to provide content that is represented by the respective node;

a request to provide a list of nodes that match a search query; and

a request to provide a list of actions specific to the first third-party application that can be taken while providing access to content associated with the first third-party application.

12. The method of any of claims 1-11, wherein:

the first third-party application is a media presentation application;

the device includes a first-party application that is a media presentation application;

the first party application includes a first-party user interface that has a plurality of selectable affordances for requesting performance of media presentation operations and one or more content presentation regions at respective locations in the first-party user interface; and

generating the first user interface for the first third-party application includes generating a user interface that includes selectable affordances and one or more content presentation regions that correspond in function and location to the selectable affordances and content presentation regions of the first-party user interface.

13. The method of any of claims 1-12, after sending the information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display:

receiving, from the first third-party application, an updated first set of values for populating the first user-interface template; and

in response to receiving the updated first set of values for populating the first user-interface template:

populating the first user-interface template with the updated first set of values received from the first third-party application;

generating an updated first user interface for the first third-party application using the first user-interface template populated with the updated first set of values; and

providing information to the respective display that enables the updated first user interface for the first third-party application to be displayed on the respective display.

14. The method of any of claims 1-13, wherein:

the device includes a digital personal assistant process with voice recognition capabilities; and

the method includes:

determining the capabilities of the first third-party application based on responses from the first third-party application to a plurality of requests to the first third-

party application for values associated with populating user-interface templates for user interfaces for the respective display;

generating an index of the capabilities of the first third-party application; and

providing voice access to the capabilities of the first third-party application via the digital personal assistant process.

15. The method of claim 14, including:

detecting that a predetermined condition has been met; and

in response to detecting that the predetermined condition has been met:

re-determining the capabilities of the first third-party application; and

re-generating the index of the capabilities of the first third-party application.

16. An electronic device, comprising:

one or more processors;

memory; and

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

receiving a first display request to display a user interface of a first third-party application on a respective display that is in communication with the electronic device;

in response to receiving the first display request:

obtaining a first user-interface template that is configured to be used by a plurality of third-party applications; and

requesting, from the first third-party application, one or more values for populating the first user-interface template;

receiving, from the first third-party application, while the first third-party application is running on the electronic device, a first set of values for populating the first user-interface template;

populating the first user-interface template with the first set of values received from the first third-party application;

generating a first user interface for the first third-party application using the first user-interface template populated with the first set of values; and

sending information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display.

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

receive a first display request to display a user interface of a first third-party application on a respective display that is in communication with the electronic device;

in response to receiving the first display request:

obtain a first user-interface template that is configured to be used by a plurality of third-party applications; and

request, from the first third-party application, one or more values for populating the first user-interface template;

receive, from the first third-party application, while the first third-party application is running on the electronic device, a first set of values for populating the first user-interface template;

populate the first user-interface template with the first set of values received from the first third-party application;

generate a first user interface for the first third-party application using the first user-interface template populated with the first set of values; and

send information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display.

18. An electronic device, comprising:

means for receiving a first display request to display a user interface of a first third-party application on a respective display that is in communication with the electronic device;

means, responsive to receiving the first display request, for:

obtaining a first user-interface template that is configured to be used by a plurality of third-party applications; and

requesting, from the first third-party application, one or more values for populating the first user-interface template;

means for receiving, from the first third-party application, while the first third-party application is running on the electronic device, a first set of values for populating the first user-interface template;

means for populating the first user-interface template with the first set of values received from the first third-party application;

means for generating a first user interface for the first third-party application using the first user-interface template populated with the first set of values; and

means for sending information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display.

19. An information processing apparatus for use in an electronic device, comprising:

means for receiving a first display request to display a user interface of a first third-party application on a respective display that is in communication with the electronic device;

means, responsive to receiving the first display request, for:

obtaining a first user-interface template that is configured to be used by a plurality of third-party applications; and

requesting, from the first third-party application, one or more values for populating the first user-interface template;

means for receiving, from the first third-party application, while the first third-party application is running on the electronic device, a first set of values for populating the first user-interface template;

means for populating the first user-interface template with the first set of values received from the first third-party application;

means for generating a first user interface for the first third-party application using the first user-interface template populated with the first set of values; and

means for sending information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display.

20. An electronic device, comprising:

one or more processors;

memory; and

one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more programs including instructions for performing any of the methods of claims 1-15.

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

22. An electronic device, comprising:
means for performing any of the methods of claims 1-15.

23. An information processing apparatus for use in an electronic device, comprising:
means for performing any of the methods of claims 1-15.

24. An electronic device, comprising:
memory; and
a processing unit coupled to the memory, the processing unit configured to:
receive a first display request to display a user interface of a first third-party application on a respective display unit that is in communication with the electronic device;
in response to receiving the first display request:
obtain a first user-interface template that is configured to be used by a plurality of third-party applications; and
request, from the first third-party application, one or more values for populating the first user-interface template;
receive, from the first third-party application, while the first third-party application is running on the electronic device, a first set of values for populating the first user-interface template;
populate the first user-interface template with the first set of values received from the first third-party application;
generate a first user interface for the first third-party application using the first user-interface template populated with the first set of values; and
send information to the respective display unit that enables the first user interface for the first third-party application to be displayed on the respective display unit.

25. The electronic device of claim 24, wherein the processing unit is further configured to:

while the first user-interface for the first third-party application is displayed on the respective display unit, receive, from an input device for the respective display unit a respective input that corresponds to selection of an option in the first user interface; and

in response to receiving the respective input, provide information to the first third-party application running on the electronic device that enables the first third-party application to respond to selection of the option in the first user interface;

receive, from the first third-party application, the response to selection of the option from the first third-party application; and

send information to the respective display unit that enables the response to selection of the option to be provided to the user.

26. The electronic device of any of claims 24-25, wherein the processing unit is further configured to, while the first user interface for the first third-party application is displayed on the respective display unit:

receive information indicative of a selection of a respective option in the first user interface; and

in response to receiving the information indicative of the selection of the respective option in the first user interface:

obtain a second user-interface template, different from the first user-interface template; and

request, from the first third-party application, one or more values for populating the second user-interface template;

receive, from the first third-party application, while the first third-party application is running on the electronic device, a second set of values for populating the second user-interface template;

populate the second user-interface template with the second set of values received from the first third-party application;

generate a second user interface for the first third-party application by using the second user-interface template populated with the second set of values; and

send information to the respective display unit that enables the second user interface for the first third-party application to be displayed on the respective display unit.

27. The electronic device of any of claims 24-26, wherein the processing unit is further configured to, after sending the information to the respective display unit that enables the first user interface for the first third-party application to be displayed on the respective display unit:

receive a second display request to display a user interface for a second third-party application, different from the first third-party application, on the respective display unit;

in response to receiving the second display request, request, from the second third-party application, one or more values for populating the first user-interface template;

receive, from the second third-party application, while the second third-party application is running on the electronic device, a third set of values for populating the first user-interface template;

populate the first user-interface template with the third set of values received from the second third-party application;

generate a second user interface for the second third-party application using the first user-interface template populated with the third set of values; and

send information to the respective display unit that enables the second user interface for the second third-party application to be displayed on the respective display unit.

28. The electronic device of any of claims 24-27, wherein:

the device includes a plurality of processes including a first third-party application process for running the first third-party application and a set of one or more intermediation processes for relaying information between the respective display unit and the first-third party application process;

the request of the one or more values for populating the first user-interface template is generated by an intermediation process from the set of one or more intermediation processes; and

the first user interface is generated by an intermediation process from the set of one or more intermediation processes.

29. The electronic device of any of claims 24-28, wherein one or more of the values in the first set of values are retrieved from a remote source by the first third-party application in response to the request for one or more values for populating the first user-interface template.

30. The electronic device of any of claims 24-29, wherein:

the respective display unit is a display unit in a vehicle information display system that is visible from a driver seat of a vehicle; and

the first user-interface template is one of a plurality of predetermined user-interface templates that are configured to be used to generate user interfaces for the third-party application on the respective display unit that prevents distraction of the driver.

31. The electronic device of any of claims 24-30, wherein:

the first user-interface template is used by a plurality of different third-party applications including the first third-party application and a second third-party application;

populating the first user-interface template with values received from a respective third-party application includes inserting values identified by the respective third-party application as belonging to a first class of values in a predefined region at a predefined font size in a user interface that is provided to the respective display unit; and

for the first user-interface template:

the first third-party application identifies a first type of information as the first class of values; and

the second third-party application identifies a second type of information, different from the first type of information, as the first class of values.

32. The electronic device of claim 31, wherein, for the first user-interface template:

the first third-party application identifies the names of remotely-generated content stations as the first class of values; and

the second third-party application identifies the names of predetermined playlists as the first class of values.

33. The electronic device of claim 31, wherein, for the first user-interface template:

the first third-party application identifies the names of content that can be streamed to the device as the first class of values; and

the second third-party application identifies the names of content stored on the device as the first class of values.

34. The electronic device of any of claims 24-33, wherein:

values at the first third-party application are arranged in a tree structure that includes a plurality of nodes that have predetermined properties and a tree relationship with one or more other nodes in the tree structure; and

the request for one or more values for populating the first user-interface template includes one or more of:

a request to return values that correspond to a list of child nodes of a respective node of the plurality of nodes;

a request to return values that correspond to a list of properties of a respective node of the plurality of nodes;

a request to provide content that is represented by the respective node;

a request to provide a list of nodes that match a search query; and

a request to provide a list of actions specific to the first third-party application that can be taken while providing access to content associated with the first third-party application.

35. The electronic device of any of claims 24-34, wherein:

the first third-party application is a media presentation application;

the device includes a first-party application that is a media presentation application;

the first party application includes a first-party user interface that has a plurality of selectable affordances for requesting performance of media presentation operations and one or more content presentation regions at respective locations in the first-party user interface; and

generating the first user interface for the first third-party application includes generating a user interface that includes selectable affordances and one or more content presentation regions that correspond in function and location to the selectable affordances and content presentation regions of the first-party user interface.

36. The electronic device of any of claims 24-35, wherein the processing unit is further configured to, after sending the information to the respective display unit that enables the first user interface for the first third-party application to be displayed on the respective display unit:

receive, from the first third-party application, an updated first set of values for populating the first user-interface template; and

in response to receiving the updated first set of values for populating the first user-interface template:

populate the first user-interface template with the updated first set of values received from the first third-party application;

generate an updated first user interface for the first third-party application using the first user-interface template populated with the updated first set of values; and

provide information to the respective display unit that enables the updated first user interface for the first third-party application to be displayed on the respective display unit.

37. The electronic device of any of claims 24-36, wherein:

the device includes a digital personal assistant process with voice recognition capabilities; and

the processing unit is further configured to:

determine the capabilities of the first third-party application based on responses from the first third-party application to a plurality of requests to the third-party application for values associated with populating user-interface templates for user interfaces for the respective display unit;

generate an index of the capabilities of the first third-party application; and

provide voice access to the capabilities of the first third-party application via the digital personal assistant process.

38. The electronic device of claim 37, wherein the processing unit is further configured to:

detect that a predetermined condition has been met; and

in response to detecting that the predetermined condition has been met:

re-determine the capabilities of the first third-party application; and

re-generate the index of the capabilities of the first third-party application.

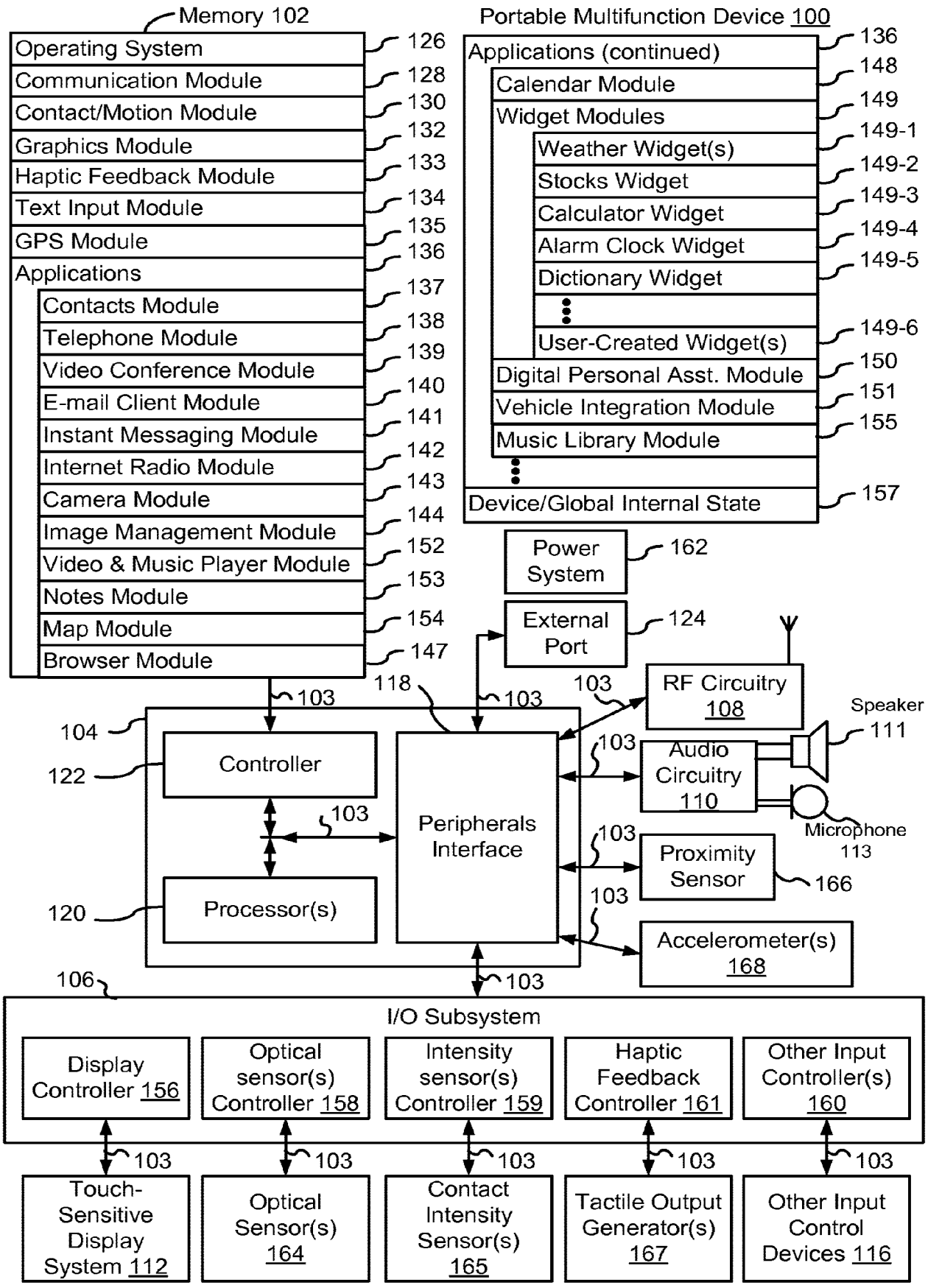

Memory 102
Operating System
126
Communication Module
128
Contact/Motion Module
130
Graphics Module
132
Haptic Feedback Module
133
Text Input Module
134
GPS Module
135
Applications
136
Contacts Module
137
Telephone Module
138
Video Conference Module
139
E-mail Client Module
140
Instant Messaging Module
141
Internet Radio Module
142
Camera Module
143
Image Management Module
144
Video & Music Player Module
152
Notes Module
153
Map Module
154
Browser Module
147
Portable Multifunction Device 100
Applications (continued)
136
Calendar Module
148
Widget Modules
149
Weather Widget(s)
149-1
Stocks Widget
149-2
Calculator Widget
149-3
Alarm Clock Widget
149-4
Dictionary Widget
149-5
User-Created Widget(s)
149-6
Digital Personal Asst. Module
150
Vehicle Integration Module
151
Music Library Module
155
Device/Global Internal State
157
Power System
162
External Port
124
RF Circuitry 108
Speaker
111
Audio Circuitry 110
Microphone 113
Proximity Sensor
166
Accelerometer(s) 168
Peripherals Interface
Controller
Processor(s)
104
122
120
118
103
106
I/O Subsystem
Display Controller 156
Optical sensor(s) Controller 158
Intensity sensor(s) Controller 159
Haptic Feedback Controller 161
Other Input Controller(s) 160
Touch-Sensitive Display System 112
Optical Sensor(s) 164
Contact Intensity Sensor(s) 165
Tactile Output Generator(s) 167
Other Input Control Devices 116

Figure 1A

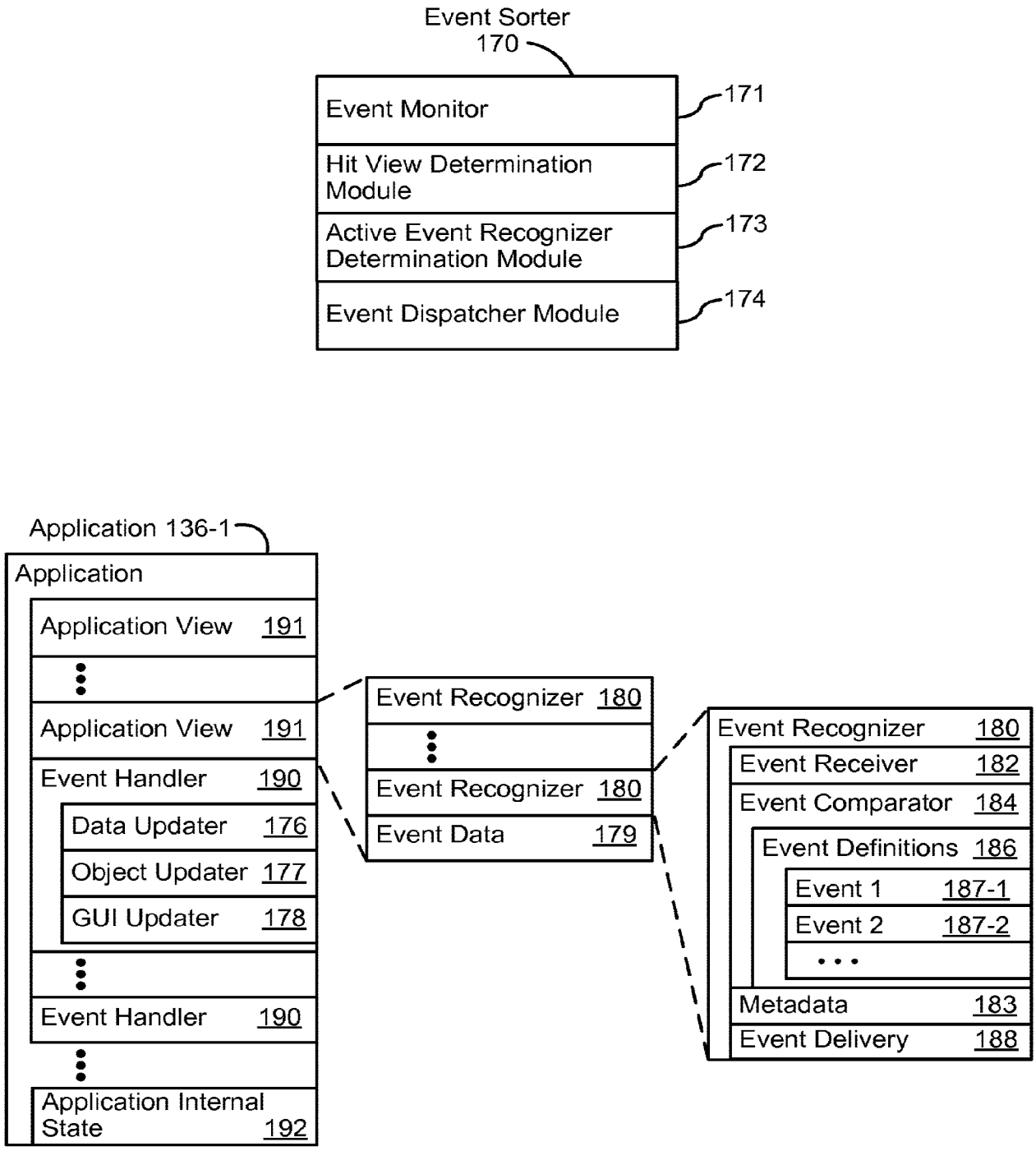

Event Sorter
170
Event Monitor
171
Hit View Determination Module
172
Active Event Recognizer Determination Module
173
Event Dispatcher Module
174
Application 136-1
Application
Application View 191
Application View 191
Event Handler 190
Data Updater 176
Object Updater 177
GUI Updater 178
Event Handler 190
Application Internal State 192
Event Recognizer 180
Event Recognizer 180
Event Data 179
Event Recognizer 180
Event Receiver 182
Event Comparator 184
Event Definitions 186
Event 1 187-1
Event 2 187-2
Metadata 183
Event Delivery 188

Figure 1B

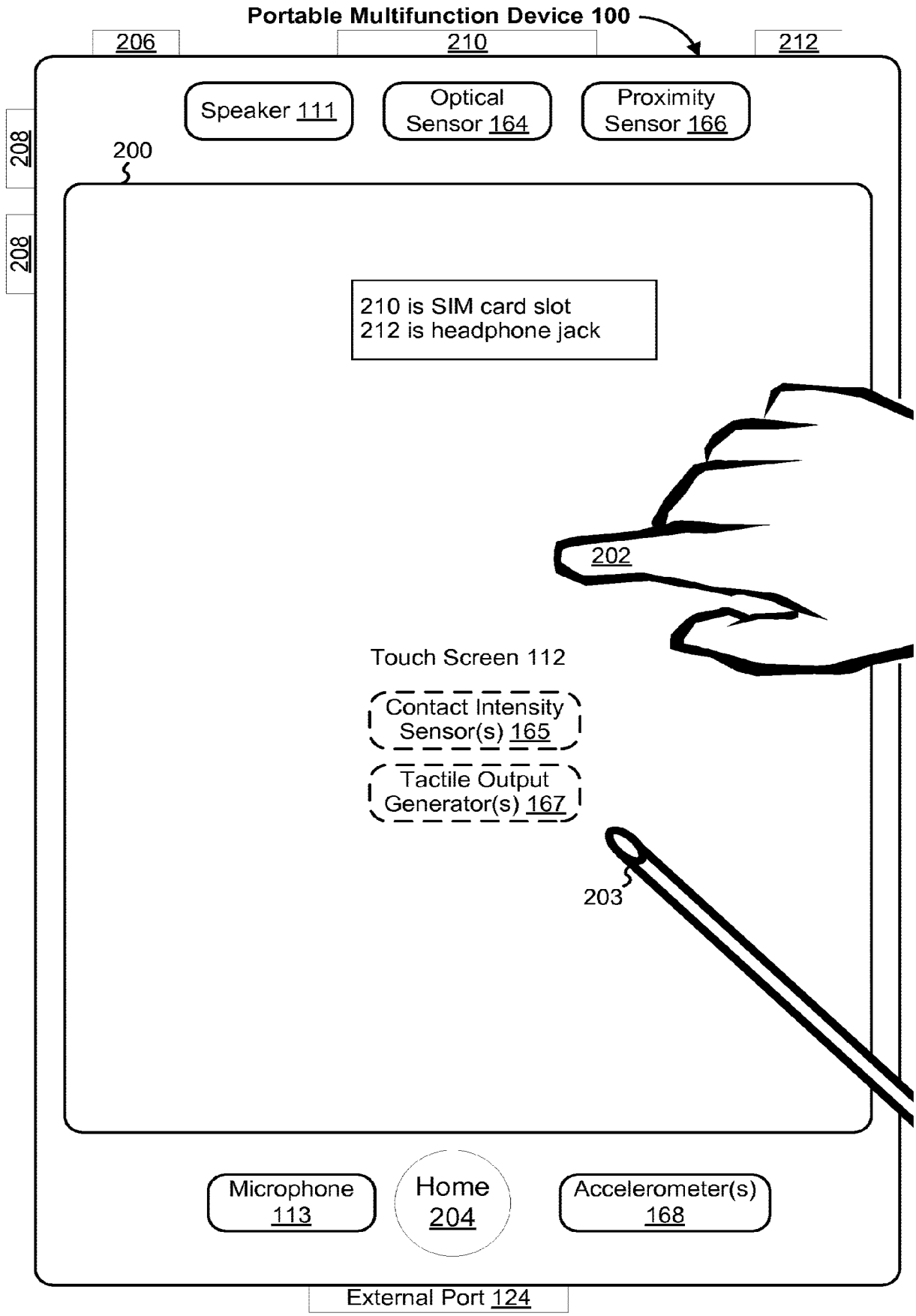

Portable Multifunction Device 100
206
210
212
Speaker 111
Optical Sensor 164
Proximity Sensor 166
208
200
208
210 is SIM card slot
212 is headphone jack
202
Touch Screen 112
Contact Intensity Sensor(s) 165
Tactile Output Generator(s) 167
203
Microphone 113
Home 204
Accelerometer(s) 168
External Port 124

Figure 2

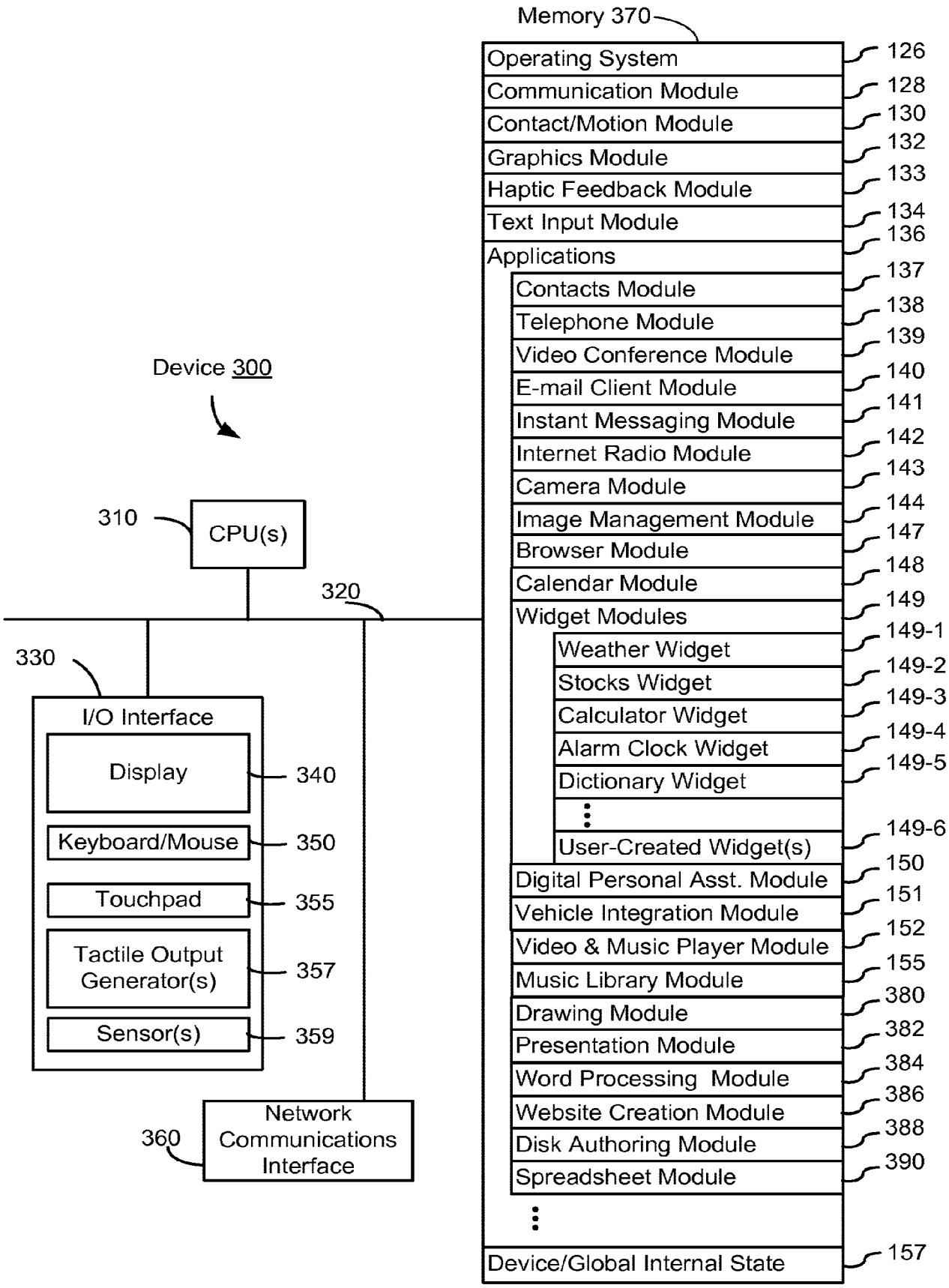

Memory 370
Operating System
126
Communication Module
128
Contact/Motion Module
130
Graphics Module
132
Haptic Feedback Module
133
Text Input Module
134
Applications
136
Contacts Module
137
Telephone Module
138
Video Conference Module
139
E-mail Client Module
140
Instant Messaging Module
141
Internet Radio Module
142
Camera Module
143
Image Management Module
144
Browser Module
147
Calendar Module
148
Widget Modules
149
Weather Widget
149-1
Stocks Widget
149-2
Calculator Widget
149-3
Alarm Clock Widget
149-4
Dictionary Widget
149-5
User-Created Widget(s)
149-6
Digital Personal Asst. Module
150
Vehicle Integration Module
151
Video & Music Player Module
152
Music Library Module
155
Drawing Module
380
Presentation Module
382
Word Processing Module
384
Website Creation Module
386
Disk Authoring Module
388
Spreadsheet Module
390
Device/Global Internal State
157
Device 300
310
CPU(s)
320
330
I/O Interface
Display
340
Keyboard/Mouse
350
Touchpad
355
Tactile Output Generator(s)
357
Sensor(s)
359
360
Network Communications Interface

Figure 3

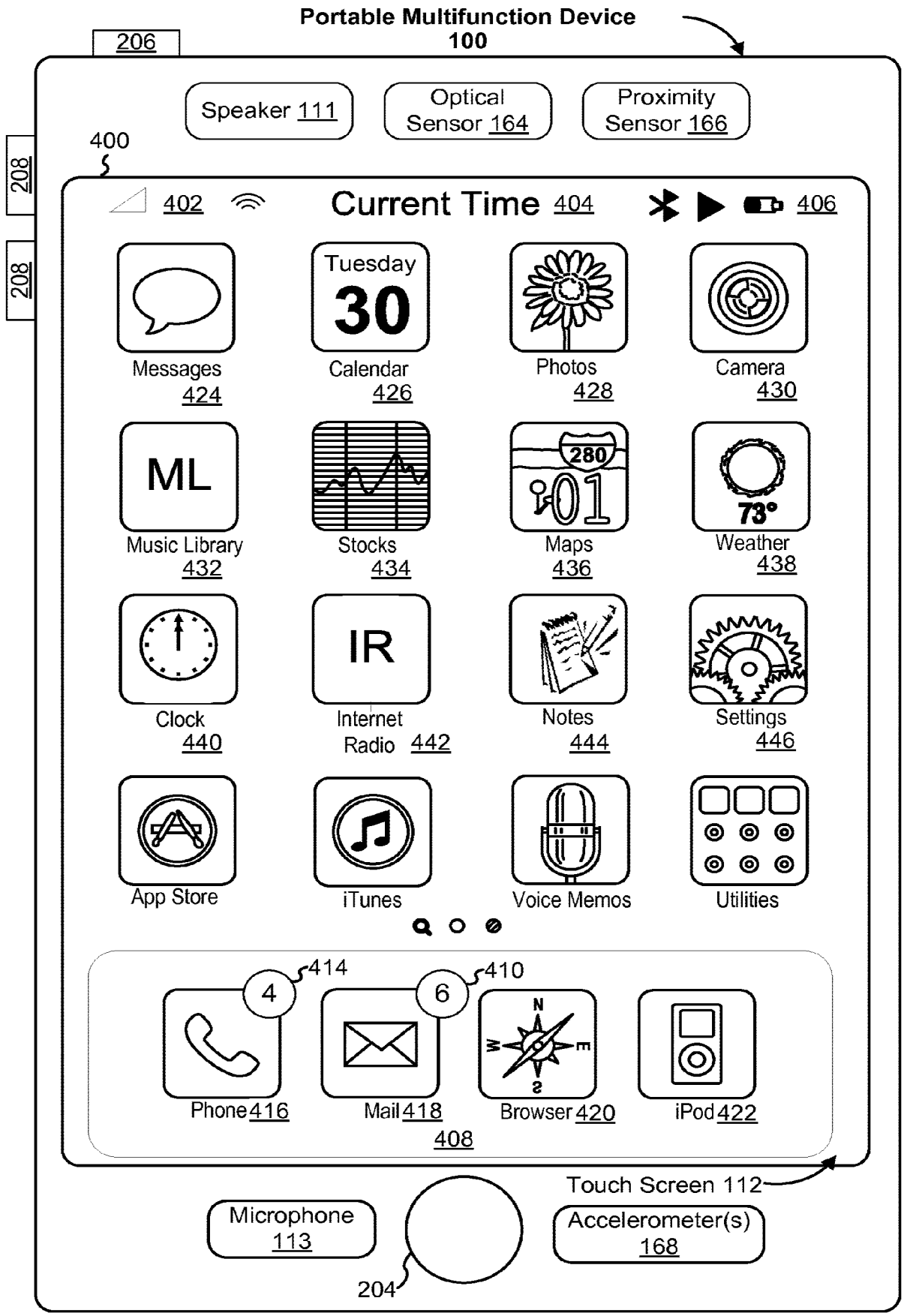
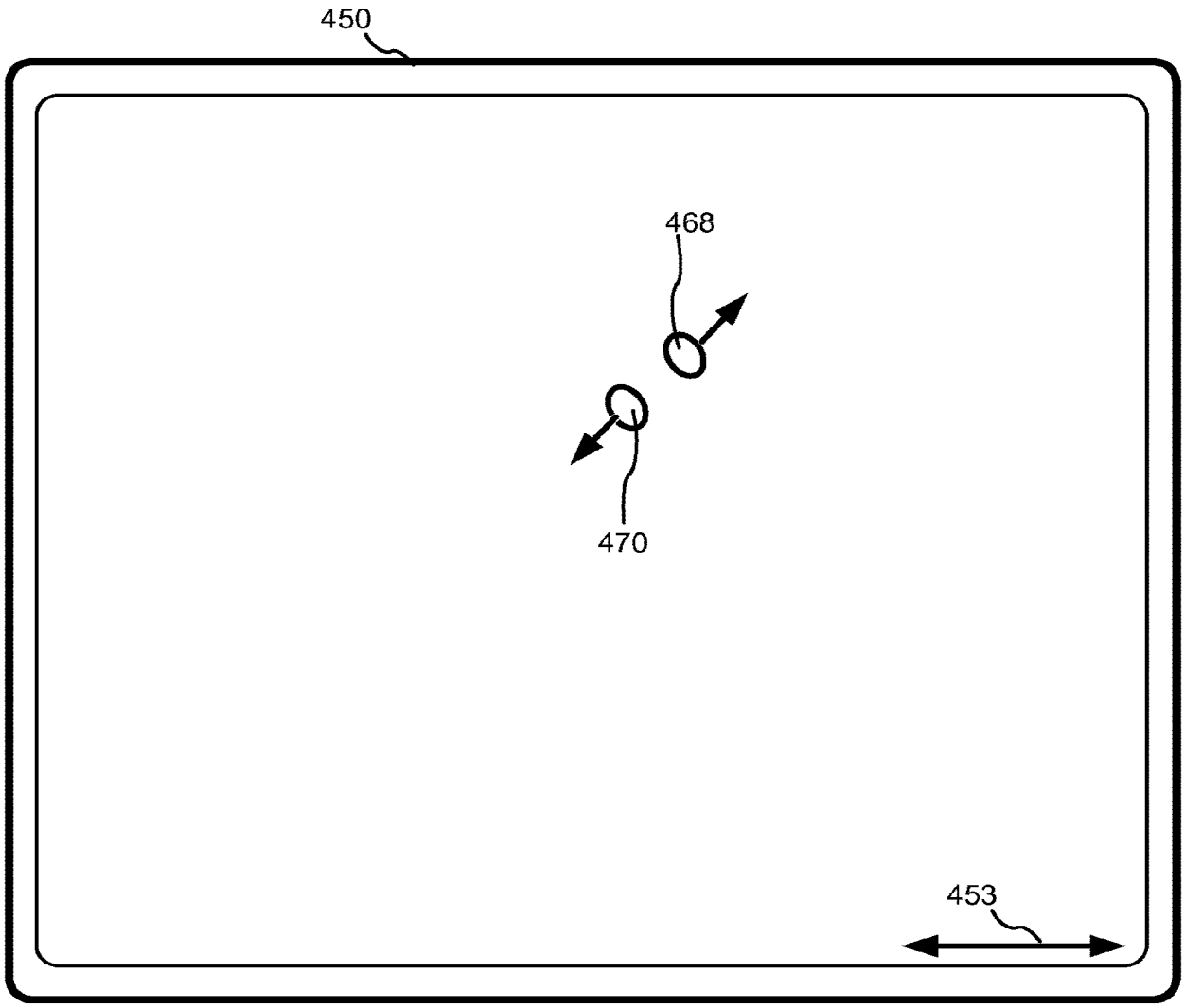

Portable Multifunction Device
100
206
Speaker 111
Optical Sensor 164
Proximity Sensor 166
400
208
208
402
Current Time 404
406
Messages 424
Tuesday
30
Calendar 426
Photos 428
Camera 430
ML
Music Library 432
Stocks 434
280
Maps 436
73°
Weather 438
Clock 440
IR
Internet Radio 442
Notes 444
Settings 446
App Store
iTunes
Voice Memos
Utilities
4
414
6
410
N
W
E
S
Phone 416
Mail 418
Browser 420
iPod 422
408
Touch Screen 112
Microphone 113
Accelerometer(s) 168
204

Figure 4A


450
468
470
453

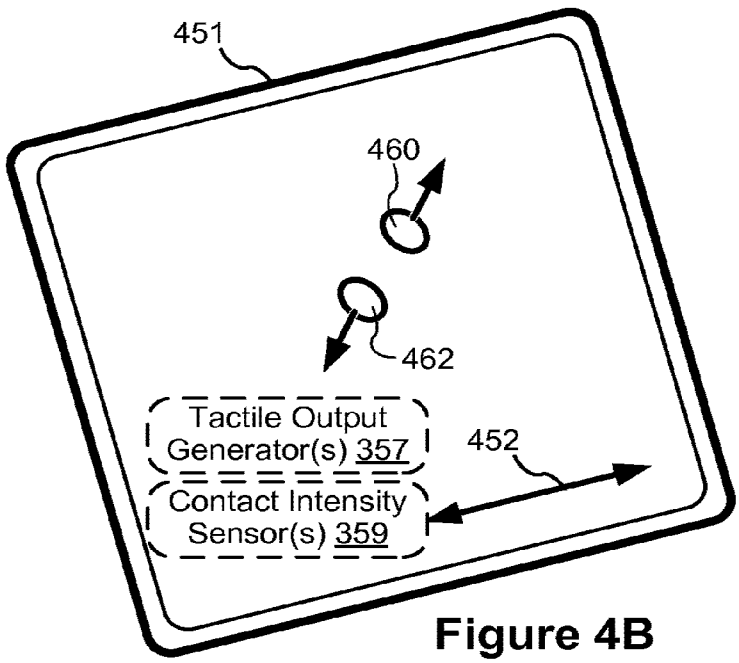

451
460
462
Tactile Output Generator(s) 357
452
Contact Intensity Sensor(s) 359

Figure 4B

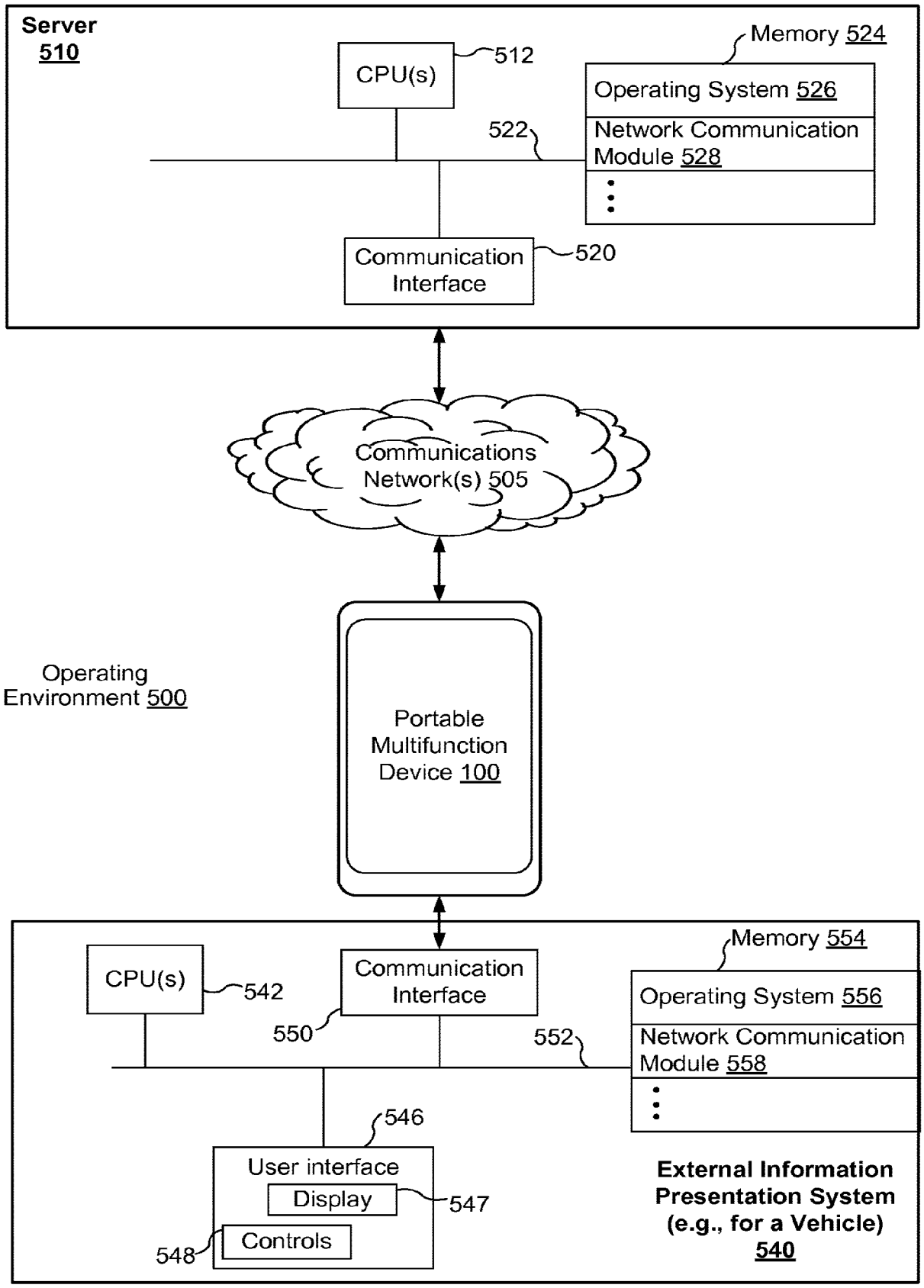

Server
510
CPU(s)
512
Memory 524
Operating System 526
Network Communication Module 528
522
Communication Interface
520
Communications Network(s) 505
Operating Environment 500
Portable Multifunction Device 100
CPU(s)
542
Communication Interface
550
Memory 554
Operating System 556
Network Communication Module 558
552
546
User interface
Display
547
Controls
548
External Information Presentation System (e.g., for a Vehicle)
540

Figure 5

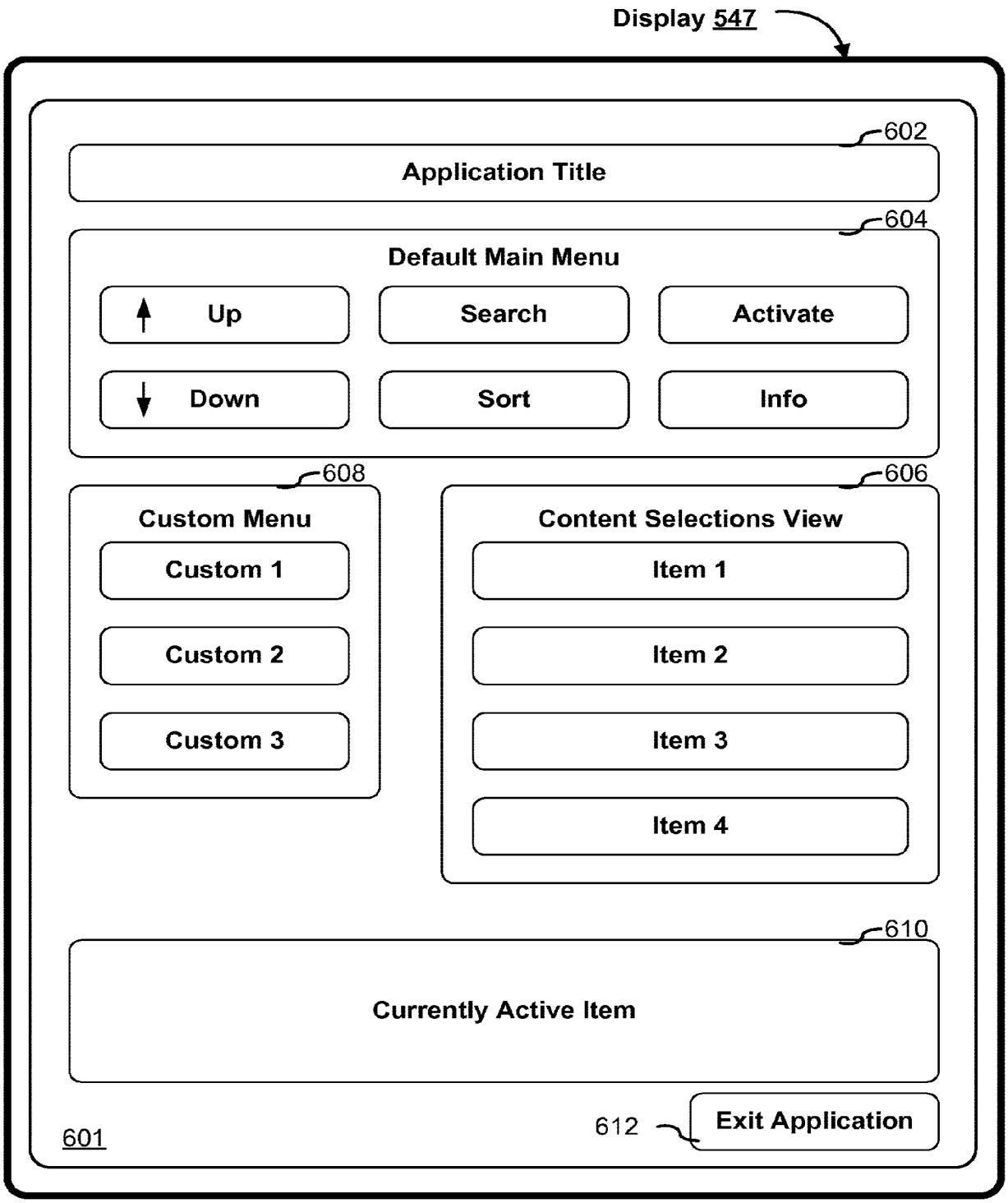

Display 547
602
Application Title
604
Default Main Menu
Up
Search
Activate
Down
Sort
Info
608
Custom Menu
Custom 1
Custom 2
Custom 3
606
Content Selections View
Item 1
Item 2
Item 3
Item 4
610
Currently Active Item
601
612
Exit Application

Figure 6A

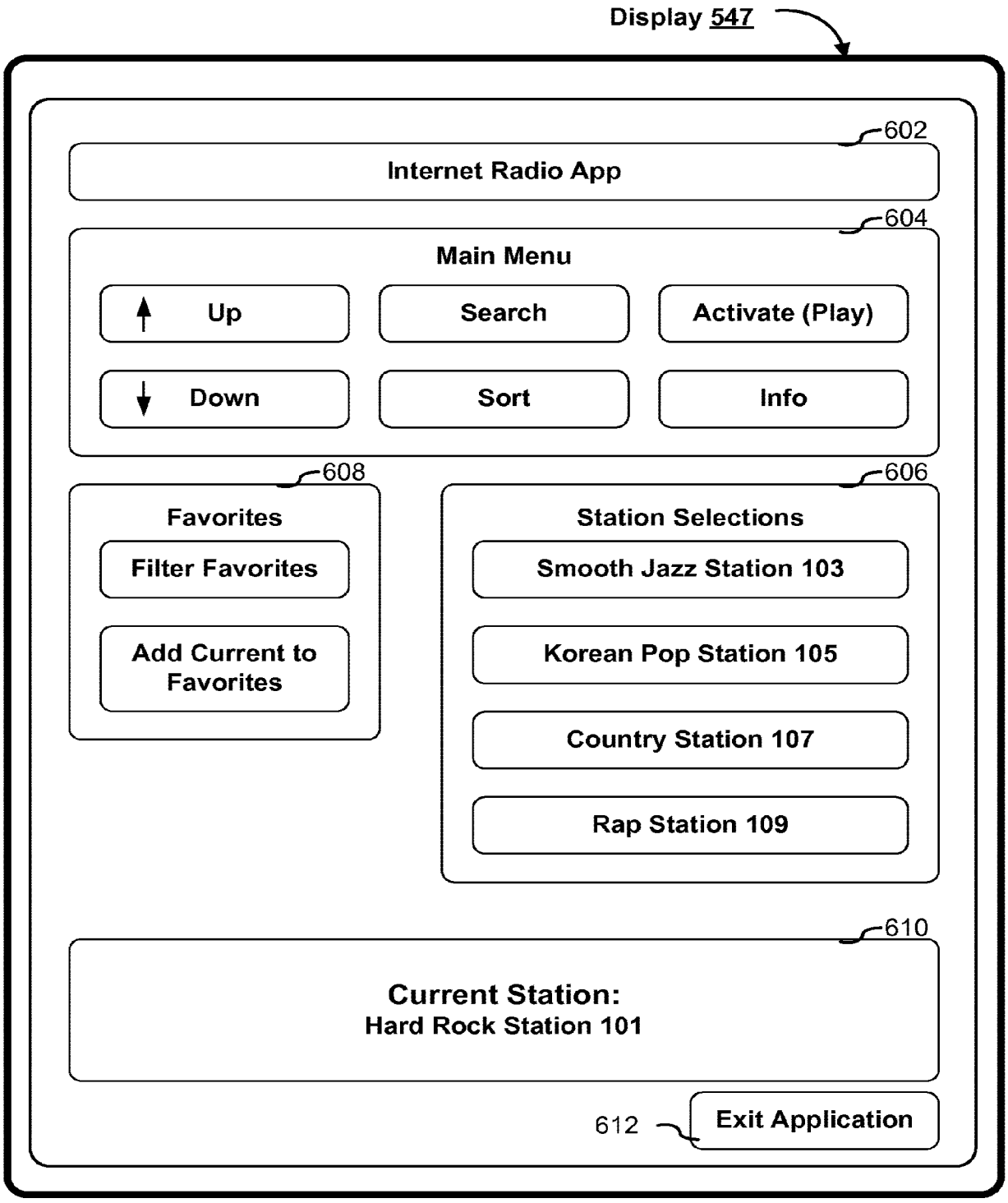

Display 547
602
Internet Radio App
604
Main Menu
Up
Search
Activate (Play)
Down
Sort
Info
608
Favorites
Filter Favorites
Add Current to Favorites
606
Station Selections
Smooth Jazz Station 103
Korean Pop Station 105
Country Station 107
Rap Station 109
610
Current Station:
Hard Rock Station 101
612
Exit Application

Figure 6B

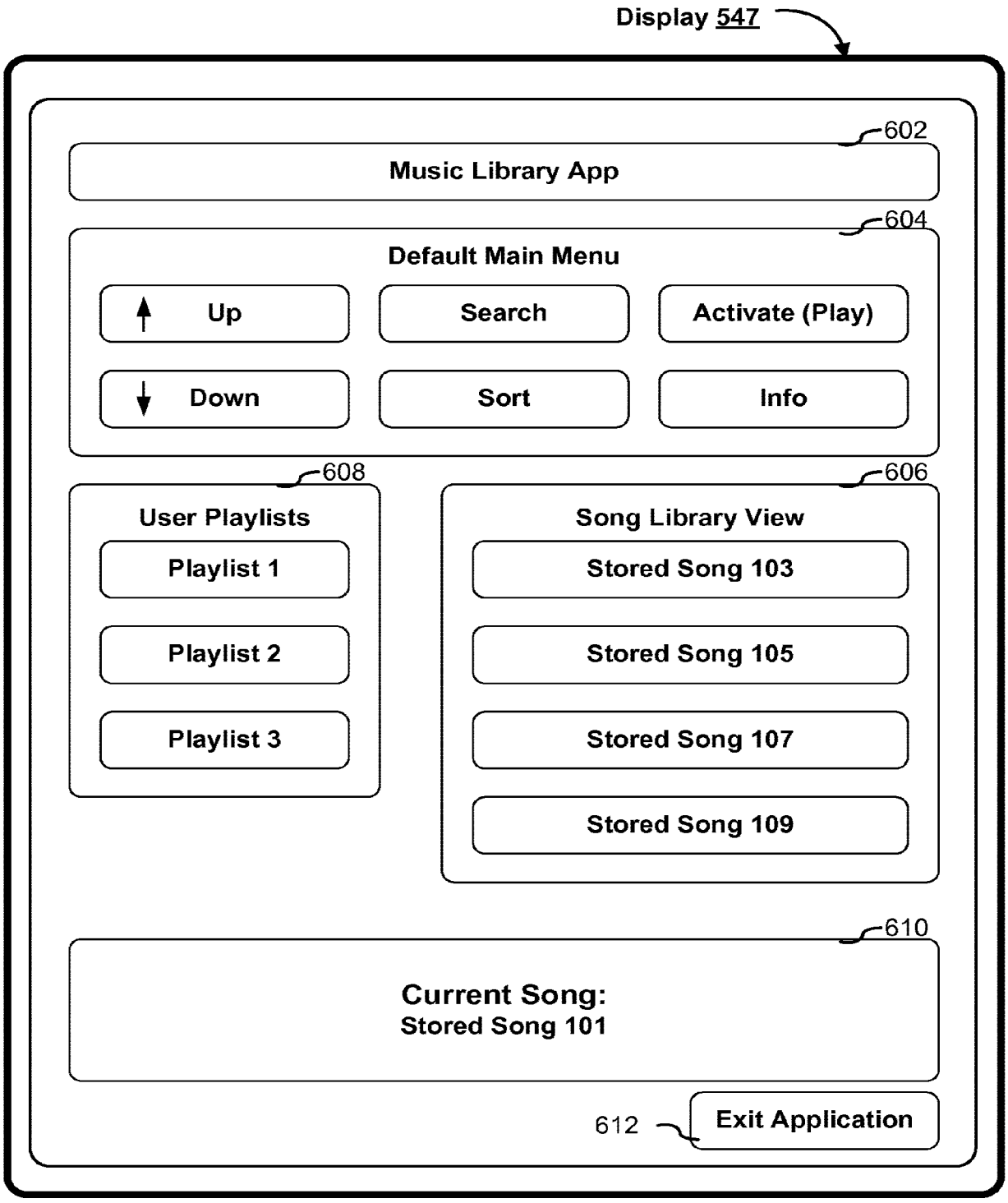

Display 547
602
Music Library App
604
Default Main Menu
Up
Search
Activate (Play)
Down
Sort
Info
608
User Playlists
Playlist 1
Playlist 2
Playlist 3
606
Song Library View
Stored Song 103
Stored Song 105
Stored Song 107
Stored Song 109
610
Current Song:
Stored Song 101
612
Exit Application

Figure 6C

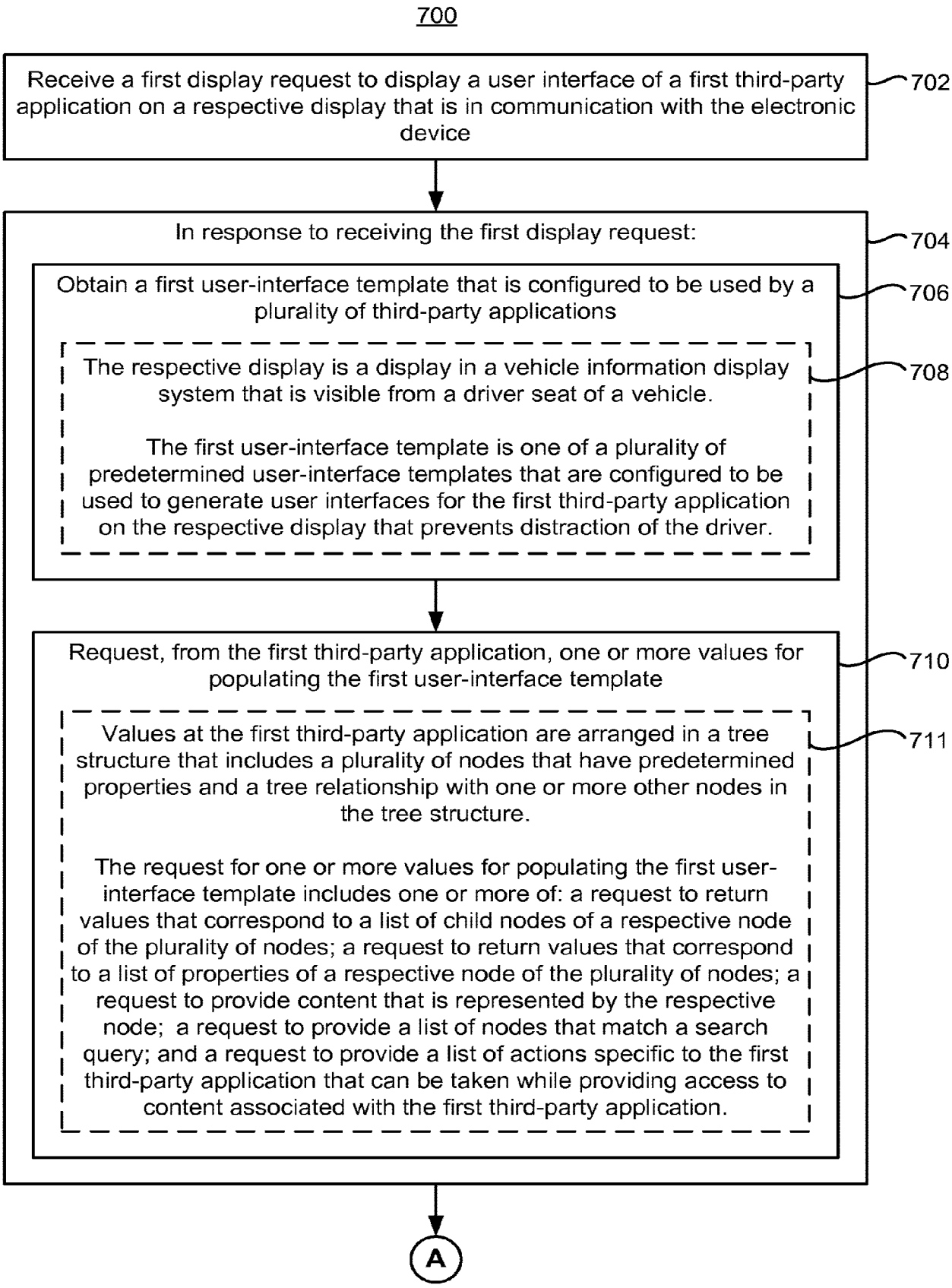

700
Receive a first display request to display a user interface of a first third-party application on a respective display that is in communication with the electronic device
702
In response to receiving the first display request:
704
Obtain a first user-interface template that is configured to be used by a plurality of third-party applications
706
The respective display is a display in a vehicle information display system that is visible from a driver seat of a vehicle.
The first user-interface template is one of a plurality of predetermined user-interface templates that are configured to be used to generate user interfaces for the first third-party application on the respective display that prevents distraction of the driver.
708
Request, from the first third-party application, one or more values for populating the first user-interface template
710
Values at the first third-party application are arranged in a tree structure that includes a plurality of nodes that have predetermined properties and a tree relationship with one or more other nodes in the tree structure.
The request for one or more values for populating the first user-interface template includes one or more of: a request to return values that correspond to a list of child nodes of a respective node of the plurality of nodes; a request to return values that correspond to a list of properties of a respective node of the plurality of nodes; a request to provide content that is represented by the respective node; a request to provide a list of nodes that match a search query; and a request to provide a list of actions specific to the first third-party application that can be taken while providing access to content associated with the first third-party application.
711
A

Figure 7A

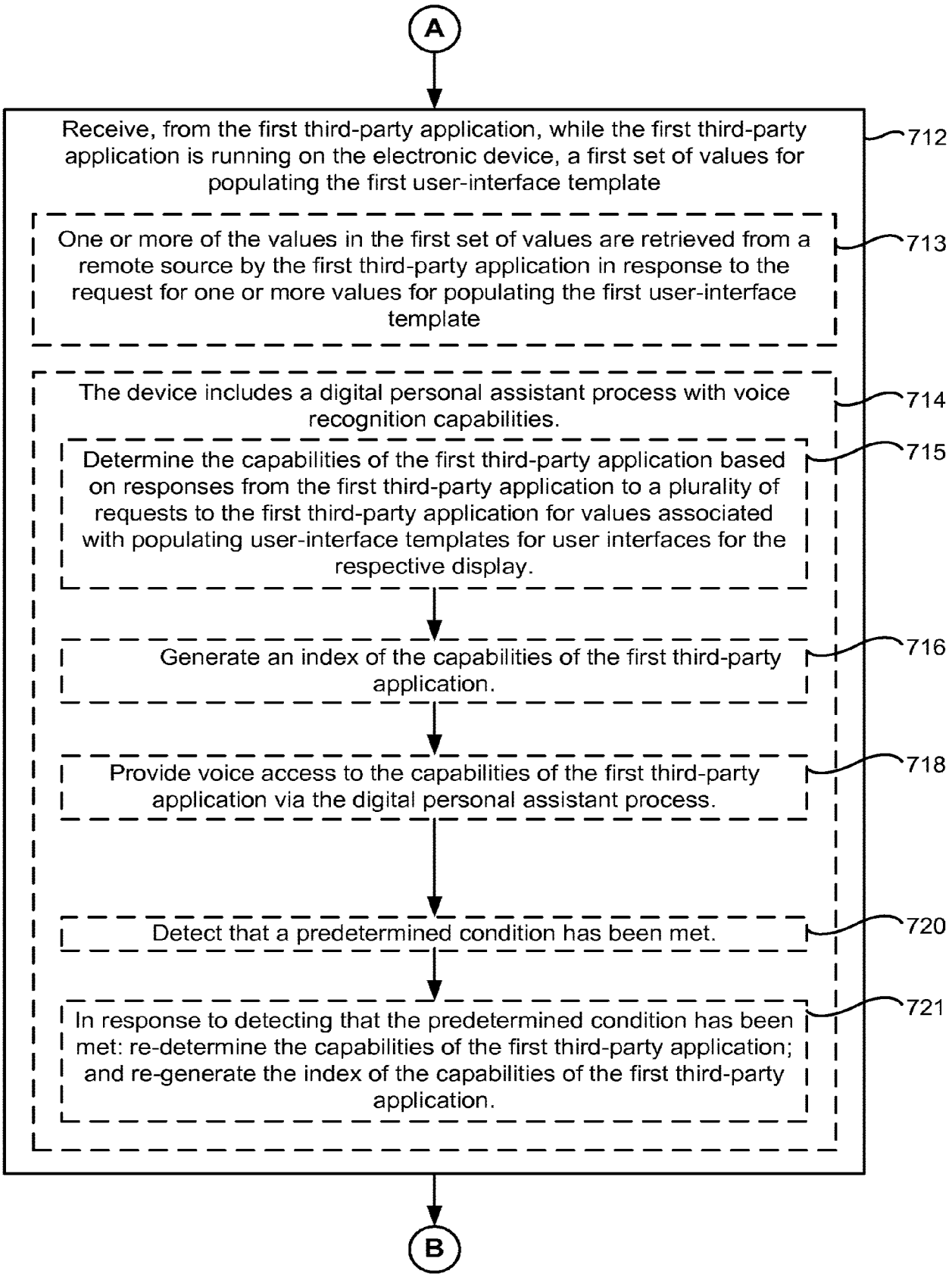

A
Receive, from the first third-party application, while the first third-party application is running on the electronic device, a first set of values for populating the first user-interface template
712
One or more of the values in the first set of values are retrieved from a remote source by the first third-party application in response to the request for one or more values for populating the first user-interface template
713
The device includes a digital personal assistant process with voice recognition capabilities.
714
Determine the capabilities of the first third-party application based on responses from the first third-party application to a plurality of requests to the first third-party application for values associated with populating user-interface templates for user interfaces for the respective display.
715
Generate an index of the capabilities of the first third-party application.
716
Provide voice access to the capabilities of the first third-party application via the digital personal assistant process.
718
Detect that a predetermined condition has been met.
720
In response to detecting that the predetermined condition has been met: re-determine the capabilities of the first third-party application; and re-generate the index of the capabilities of the first third-party application.
721
B

Figure 7B

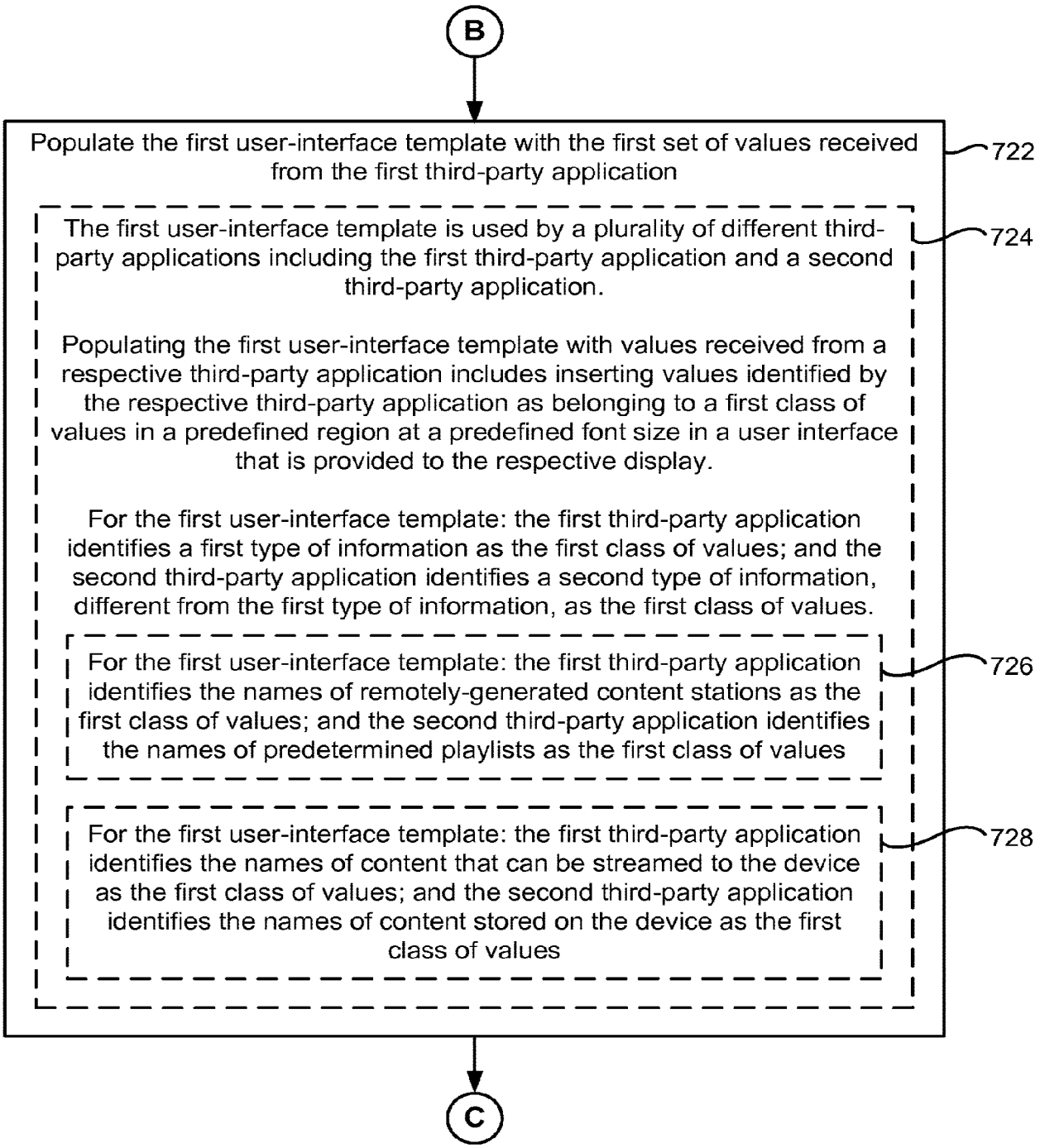

B
Populate the first user-interface template with the first set of values received from the first third-party application
722
The first user-interface template is used by a plurality of different third-party applications including the first third-party application and a second third-party application.
724
Populating the first user-interface template with values received from a respective third-party application includes inserting values identified by the respective third-party application as belonging to a first class of values in a predefined region at a predefined font size in a user interface that is provided to the respective display.
For the first user-interface template: the first third-party application identifies a first type of information as the first class of values; and the second third-party application identifies a second type of information, different from the first type of information, as the first class of values.
For the first user-interface template: the first third-party application identifies the names of remotely-generated content stations as the first class of values; and the second third-party application identifies the names of predetermined playlists as the first class of values
726
For the first user-interface template: the first third-party application identifies the names of content that can be streamed to the device as the first class of values; and the second third-party application identifies the names of content stored on the device as the first class of values
728
C

Figure 7C

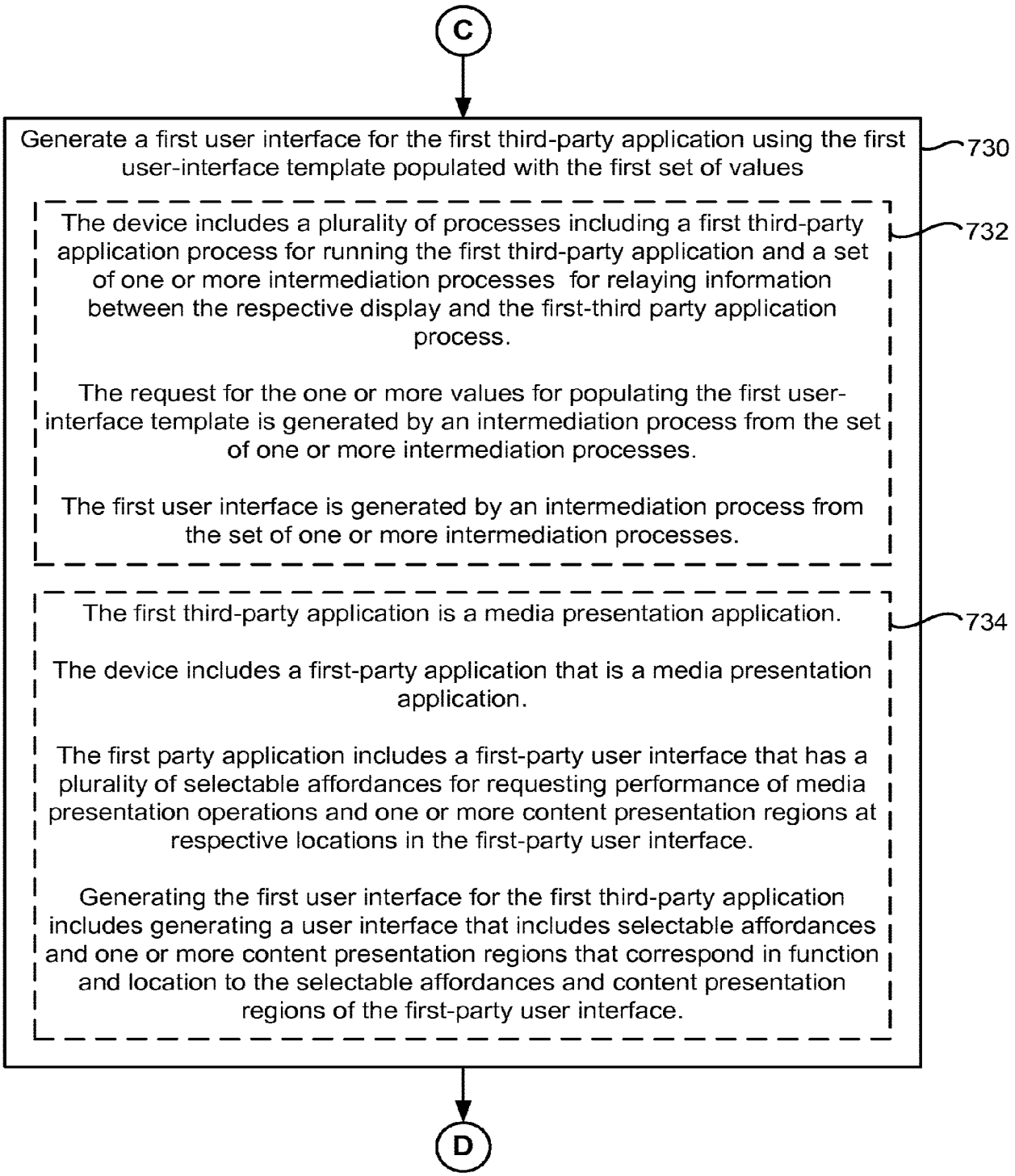

C
Generate a first user interface for the first third-party application using the first user-interface template populated with the first set of values
730
The device includes a plurality of processes including a first third-party application process for running the first third-party application and a set of one or more intermediation processes for relaying information between the respective display and the first-third party application process.
The request for the one or more values for populating the first user-interface template is generated by an intermediation process from the set of one or more intermediation processes.
The first user interface is generated by an intermediation process from the set of one or more intermediation processes.
732
The first third-party application is a media presentation application.
The device includes a first-party application that is a media presentation application.
The first party application includes a first-party user interface that has a plurality of selectable affordances for requesting performance of media presentation operations and one or more content presentation regions at respective locations in the first-party user interface.
Generating the first user interface for the first third-party application includes generating a user interface that includes selectable affordances and one or more content presentation regions that correspond in function and location to the selectable affordances and content presentation regions of the first-party user interface.
734
D

Figure 7D

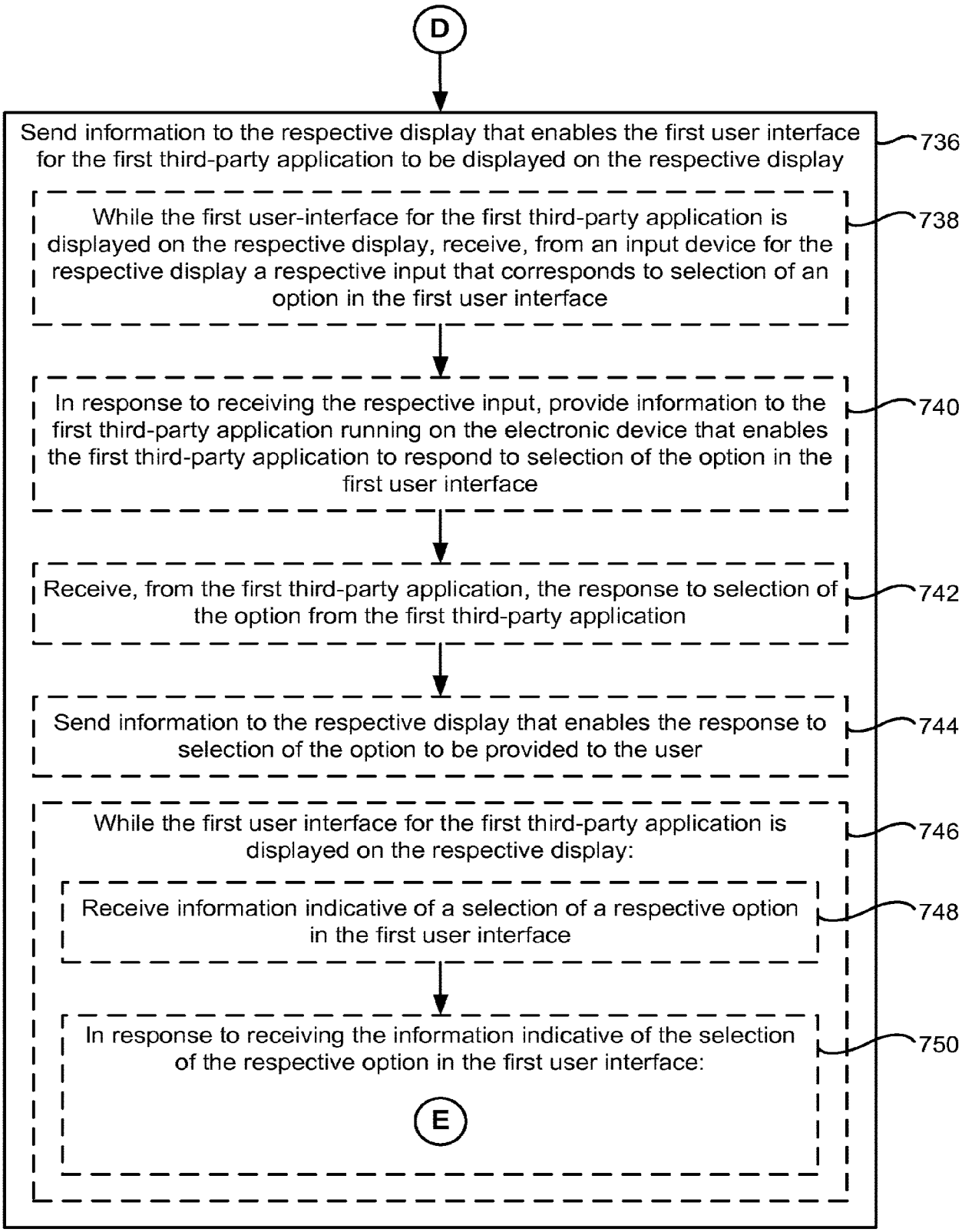

D
Send information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display
736
While the first user-interface for the first third-party application is displayed on the respective display, receive, from an input device for the respective display a respective input that corresponds to selection of an option in the first user interface
738
In response to receiving the respective input, provide information to the first third-party application running on the electronic device that enables the first third-party application to respond to selection of the option in the first user interface
740
Receive, from the first third-party application, the response to selection of the option from the first third-party application
742
Send information to the respective display that enables the response to selection of the option to be provided to the user
744
While the first user interface for the first third-party application is displayed on the respective display:
746
Receive information indicative of a selection of a respective option in the first user interface
748
In response to receiving the information indicative of the selection of the respective option in the first user interface:
750
E

Figure 7E

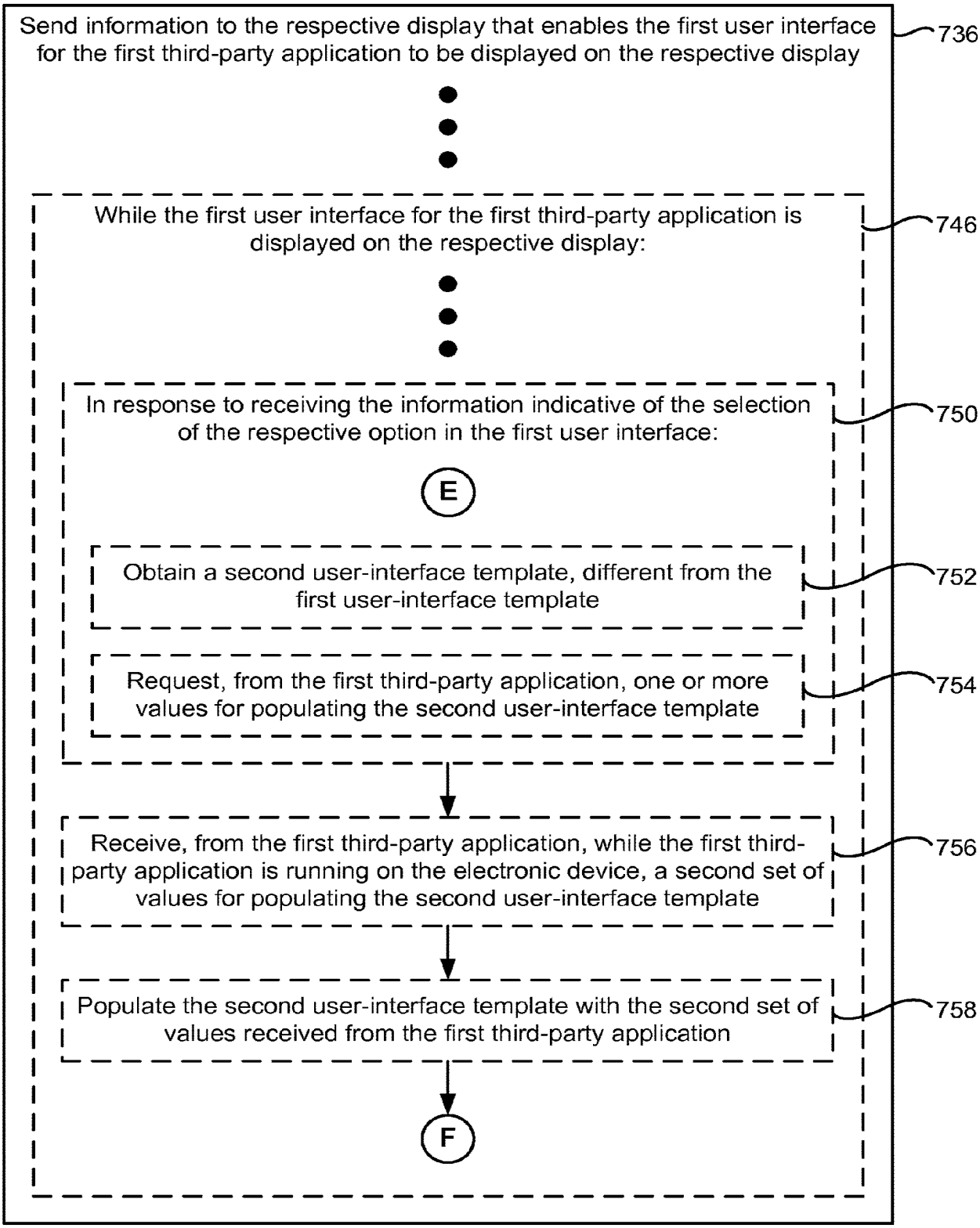

Send information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display
736
While the first user interface for the first third-party application is displayed on the respective display:
746
In response to receiving the information indicative of the selection of the respective option in the first user interface:
750
E
Obtain a second user-interface template, different from the first user-interface template
752
Request, from the first third-party application, one or more values for populating the second user-interface template
754
Receive, from the first third-party application, while the first third-party application is running on the electronic device, a second set of values for populating the second user-interface template
756
Populate the second user-interface template with the second set of values received from the first third-party application
758
F

Figure 7F

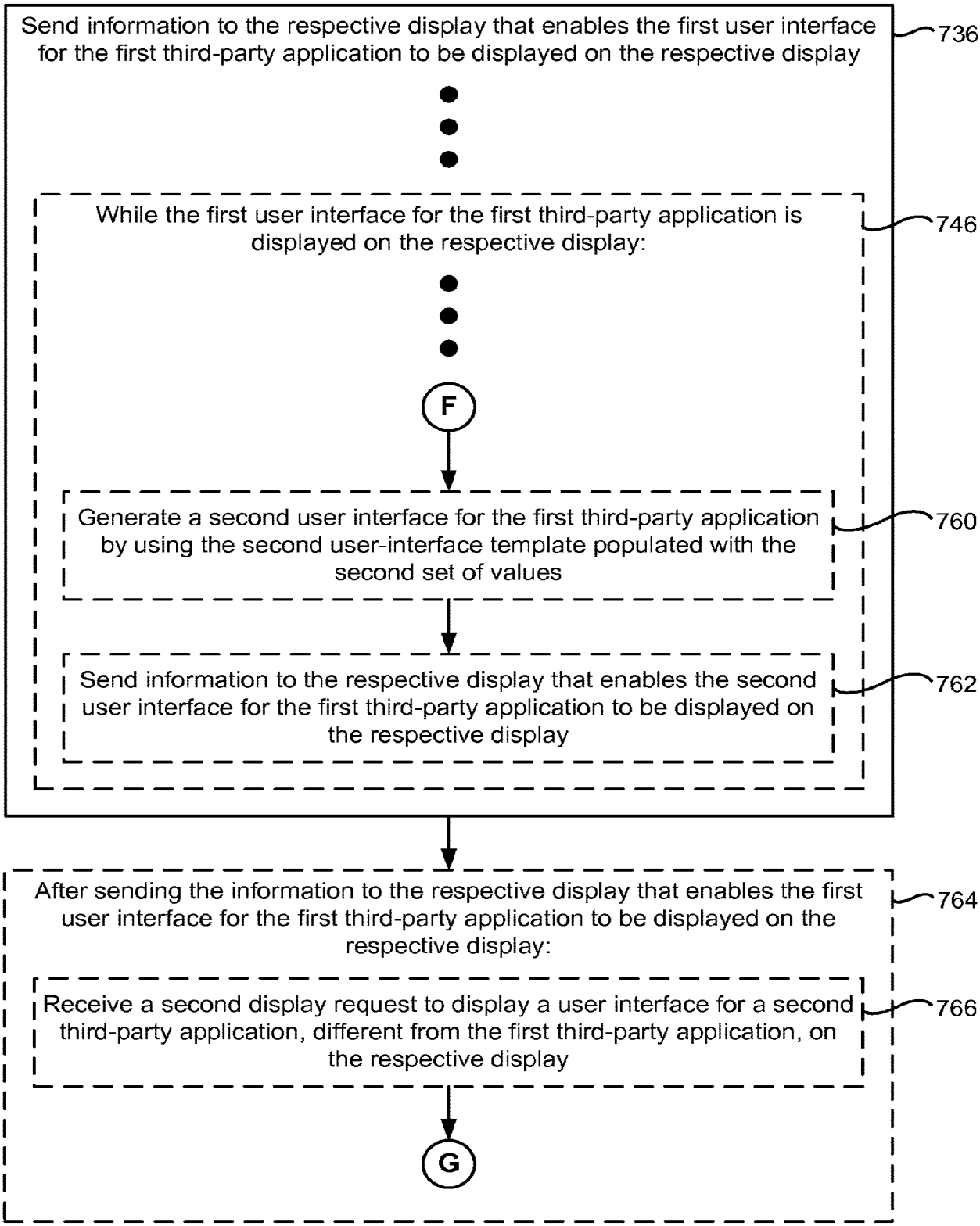

Send information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display
736
While the first user interface for the first third-party application is displayed on the respective display:
746
F
Generate a second user interface for the first third-party application by using the second user-interface template populated with the second set of values
760
Send information to the respective display that enables the second user interface for the first third-party application to be displayed on the respective display
762
After sending the information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display:
764
Receive a second display request to display a user interface for a second third-party application, different from the first third-party application, on the respective display
766
G

Figure 7G

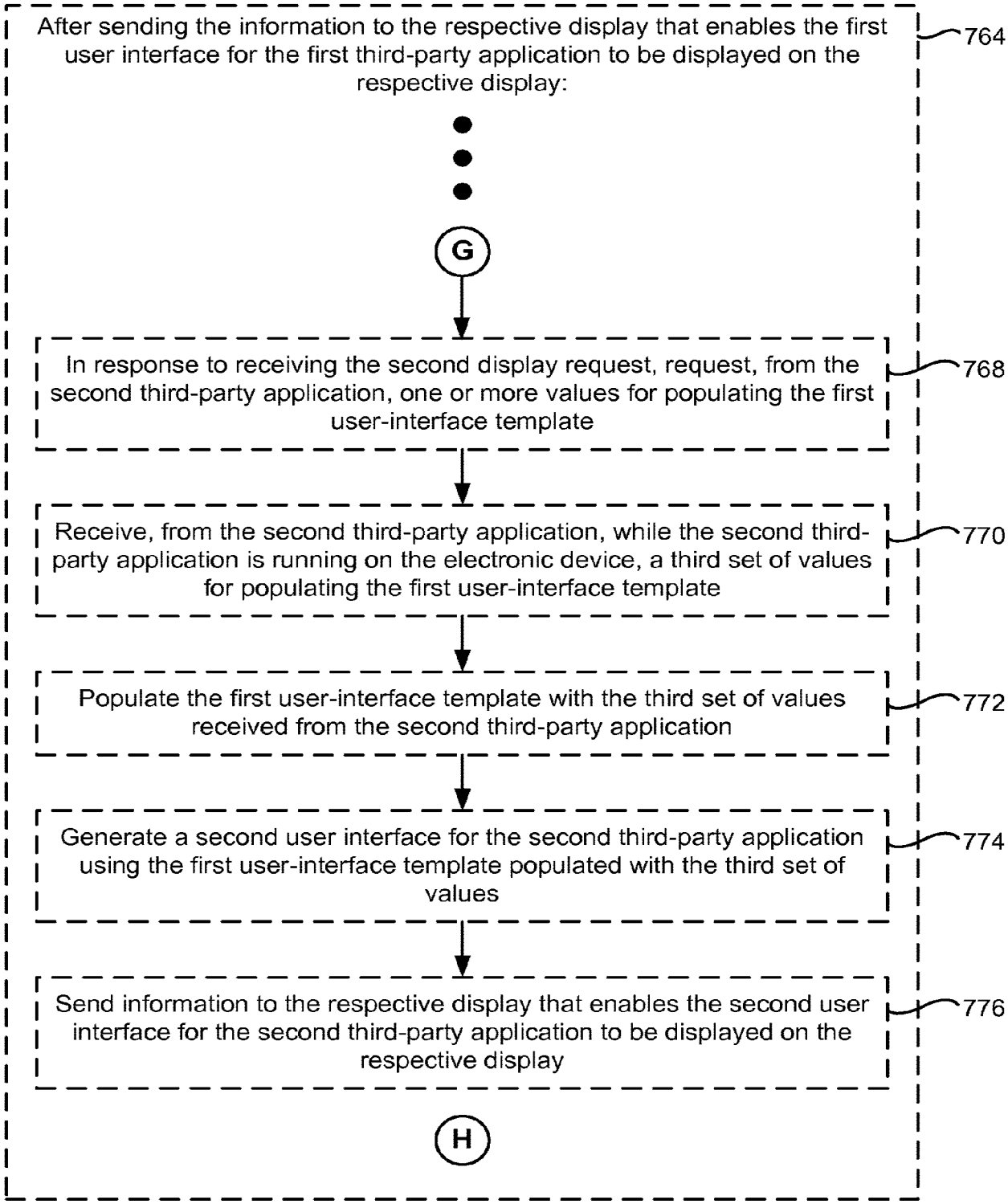

After sending the information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display:
764
G
In response to receiving the second display request, request, from the second third-party application, one or more values for populating the first user-interface template
768
Receive, from the second third-party application, while the second third-party application is running on the electronic device, a third set of values for populating the first user-interface template
770
Populate the first user-interface template with the third set of values received from the second third-party application
772
Generate a second user interface for the second third-party application using the first user-interface template populated with the third set of values
774
Send information to the respective display that enables the second user interface for the second third-party application to be displayed on the respective display
776
H

Figure 7H

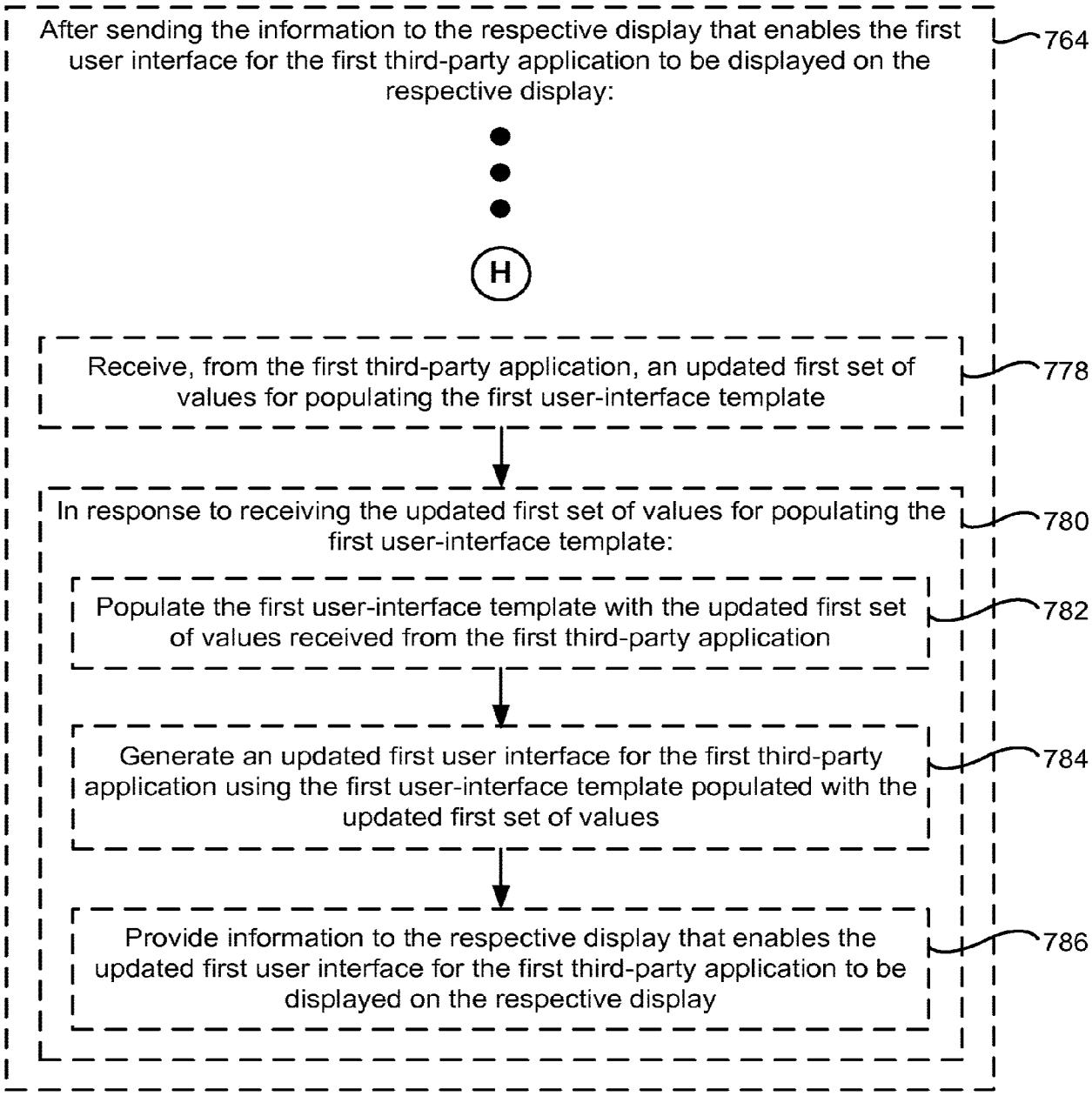

After sending the information to the respective display that enables the first user interface for the first third-party application to be displayed on the respective display:
764
H
Receive, from the first third-party application, an updated first set of values for populating the first user-interface template
778
In response to receiving the updated first set of values for populating the first user-interface template:
780
Populate the first user-interface template with the updated first set of values received from the first third-party application
782
Generate an updated first user interface for the first third-party application using the first user-interface template populated with the updated first set of values
784
Provide information to the respective display that enables the updated first user interface for the first third-party application to be displayed on the respective display
786

Figure 7I

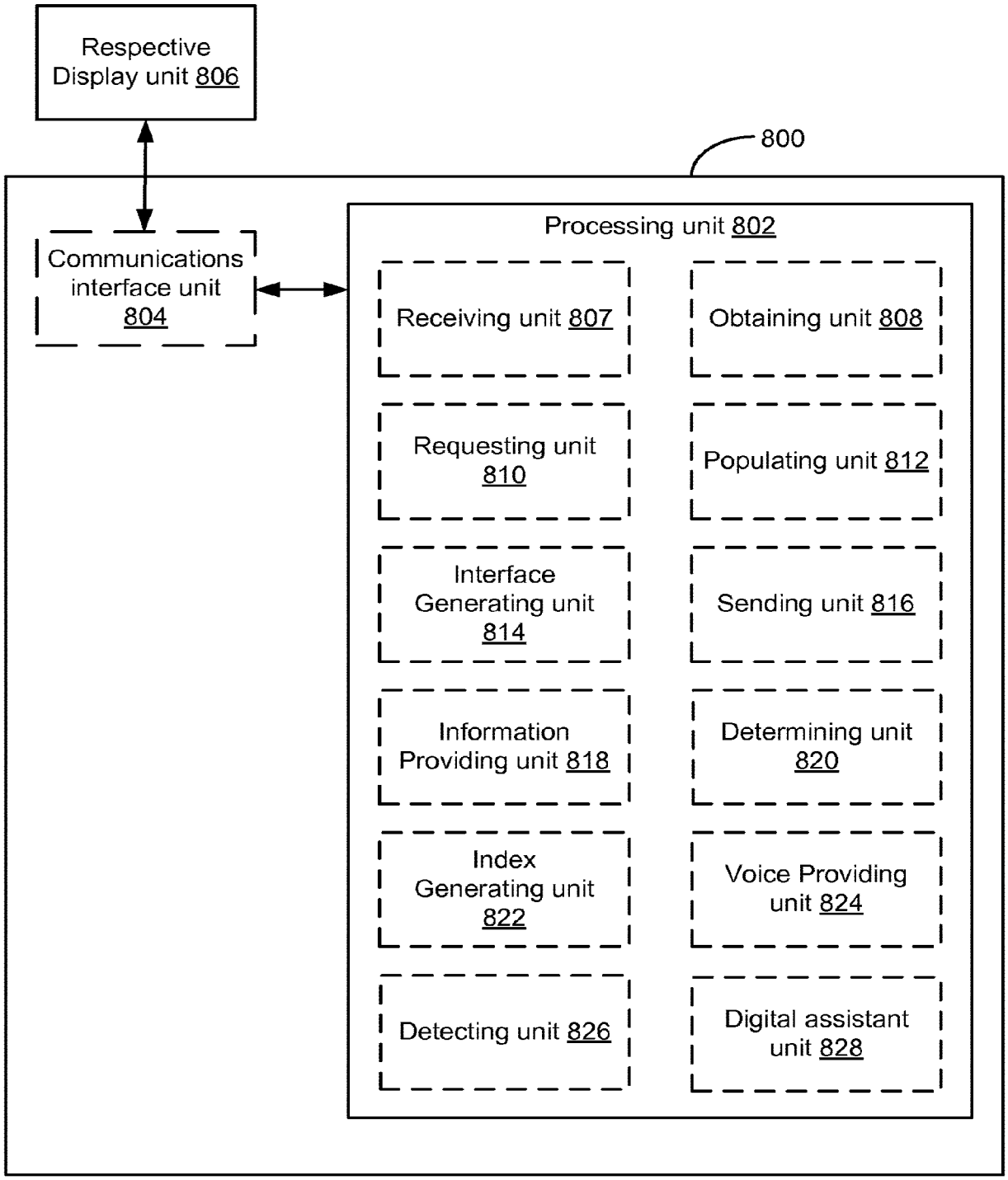

Respective Display unit 806
800
Processing unit 802
Communications interface unit 804
Receiving unit 807
Obtaining unit 808
Requesting unit 810
Populating unit 812
Interface Generating unit 814
Sending unit 816
Information Providing unit 818
Determining unit 820
Index Generating unit 822
Voice Providing unit 824
Detecting unit 826
Digital assistant unit 828

Figure 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/040416

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/44
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data, COMPENDEX, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2013/086597 A1 (CORNWALL KEVIN [GB] ET AL) 4 April 2013 (2013-04-04) paragraph [0001] - paragraph [0016]	1-38
X	US 2011/145863 A1 (ALSINA THOMAS M [US] ET AL) 16 June 2011 (2011-06-16) cited in the application paragraph [0007]; claim 1 paragraph [0030] paragraph [0165]	1-38
X	US 2010/077058 A1 (MESSER ALAN [US]) 25 March 2010 (2010-03-25) paragraph [0002] - paragraph [0014]	1-38

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
7 August 2014	18/08/2014

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

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2014/040416

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013086597 A1	04-04-2013	EP 2587370 A2	01-05-2013
		US 2013086597 A1	04-04-2013
US 2011145863 A1	16-06-2011	NONE	
US 2010077058 A1	25-03-2010	NONE	