

(21) Application No: 1004456.8
(22) Date of Filing: 17.03.2010

(51) INT CL:
H04L 29/08 (2006.01) G06F 17/30 (2006.01)
H04M 1/247 (2006.01) H04W 8/24 (2009.01)

(71) Applicant(s):
Samsung Electronics Co; Ltd.
416 Maetan-3 dong, Yeongtong-gu, Suwon-City,
Gyeonggi-do 443-742, Republic of Korea

(56) Documents Cited:
US 7502993 B1 US 6628644 B1
US 20070101004 A1

(72) Inventor(s):
Simon Hibbard
Erel Magnus

(58) Field of Search:
INT CL G06F, H04L, H04W
Other: Online: EPODOC, TEXTEN, INSPEC, INTERNET,
NPL, WPI, XPESP

(74) Agent and/or Address for Service:
EIP
Fairfax House, 15 Fulwood Place, LONDON,
WC1V 6HU, United Kingdom

(54) Title of the Invention: Method and apparatus for accessing services of a device
Abstract Title: Accessing services of a device

(57) A device (e.g. telephone, television, set-top box) is configured to run a browser application and one or more service applications (e.g. telephony, messaging, social networking, connectivity, video), and wherein the browser application is configured to display a user interface. The browser application is configured to receive a first response to a first command request, and in response to receiving the first response, to display a first user interface on the display which includes first content (e.g. updates, hyperlinks). The device further comprises a server application which is configured to recognise the first command request, the first command request being provided to the server application using a first transfer protocol (e.g. HTTP, HTTPS), and in response to the first command request, to provide instructions to a service application to execute a first set of one or more commands in order to control a function of the device. The service application is configured to, in response to receipt of the instructions from the server application, control a function of the device and generate first content for receipt by the server application. The server application is configured to, in response to receipt of the first content from said service application, generate the first response to the first command request, the first response comprising the first content.

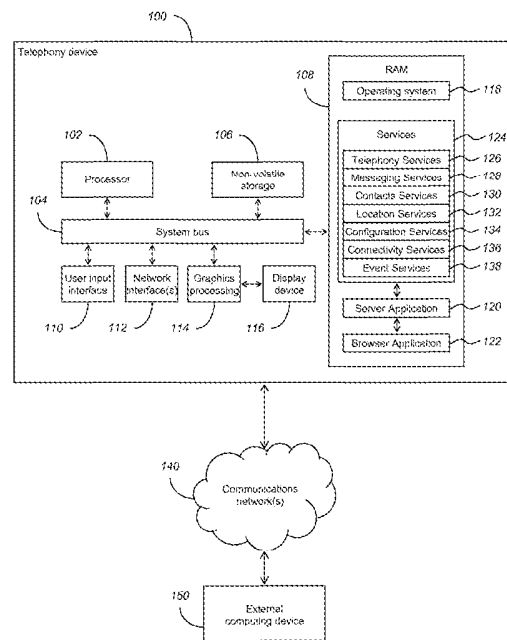


Figure 1

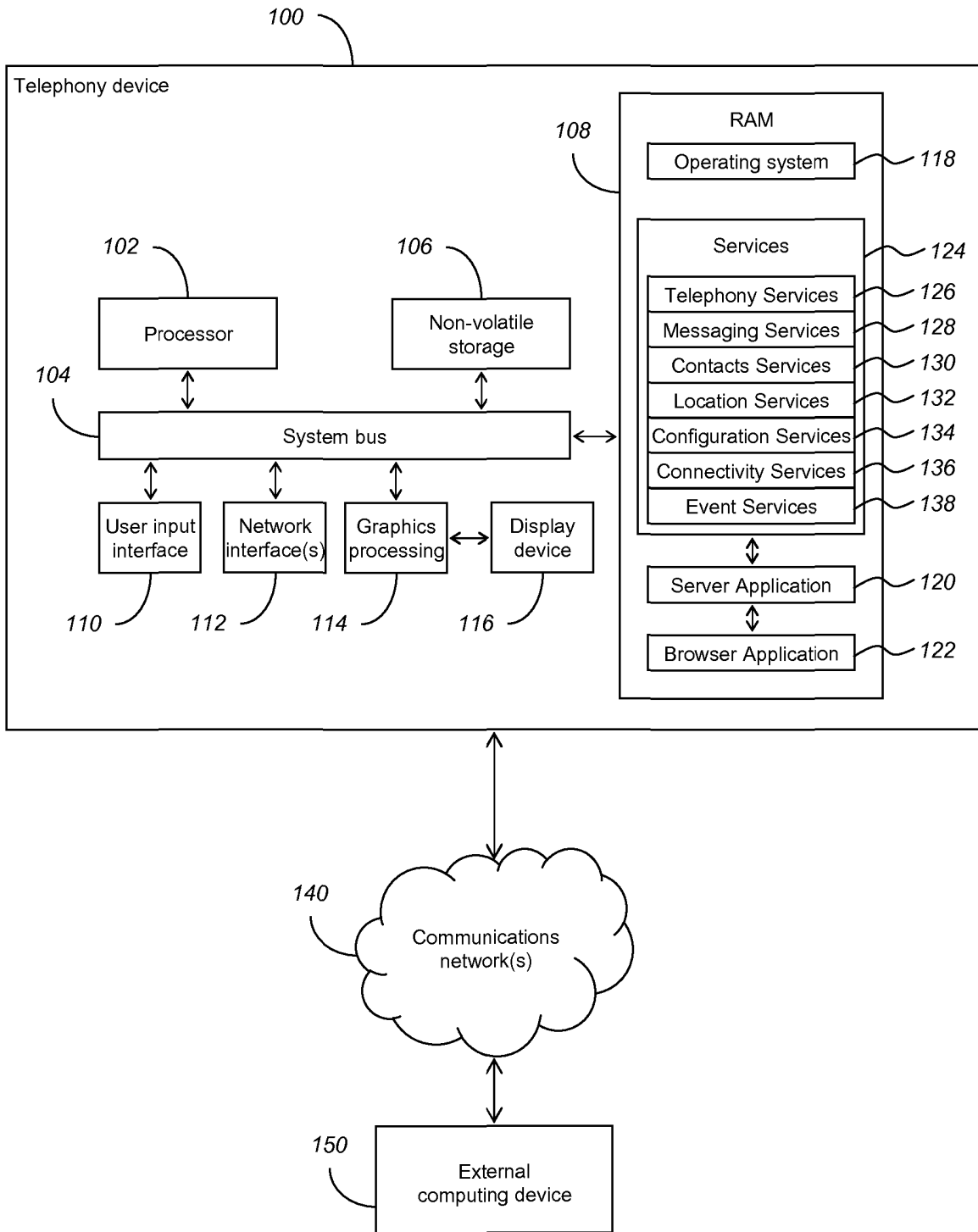


Figure 1

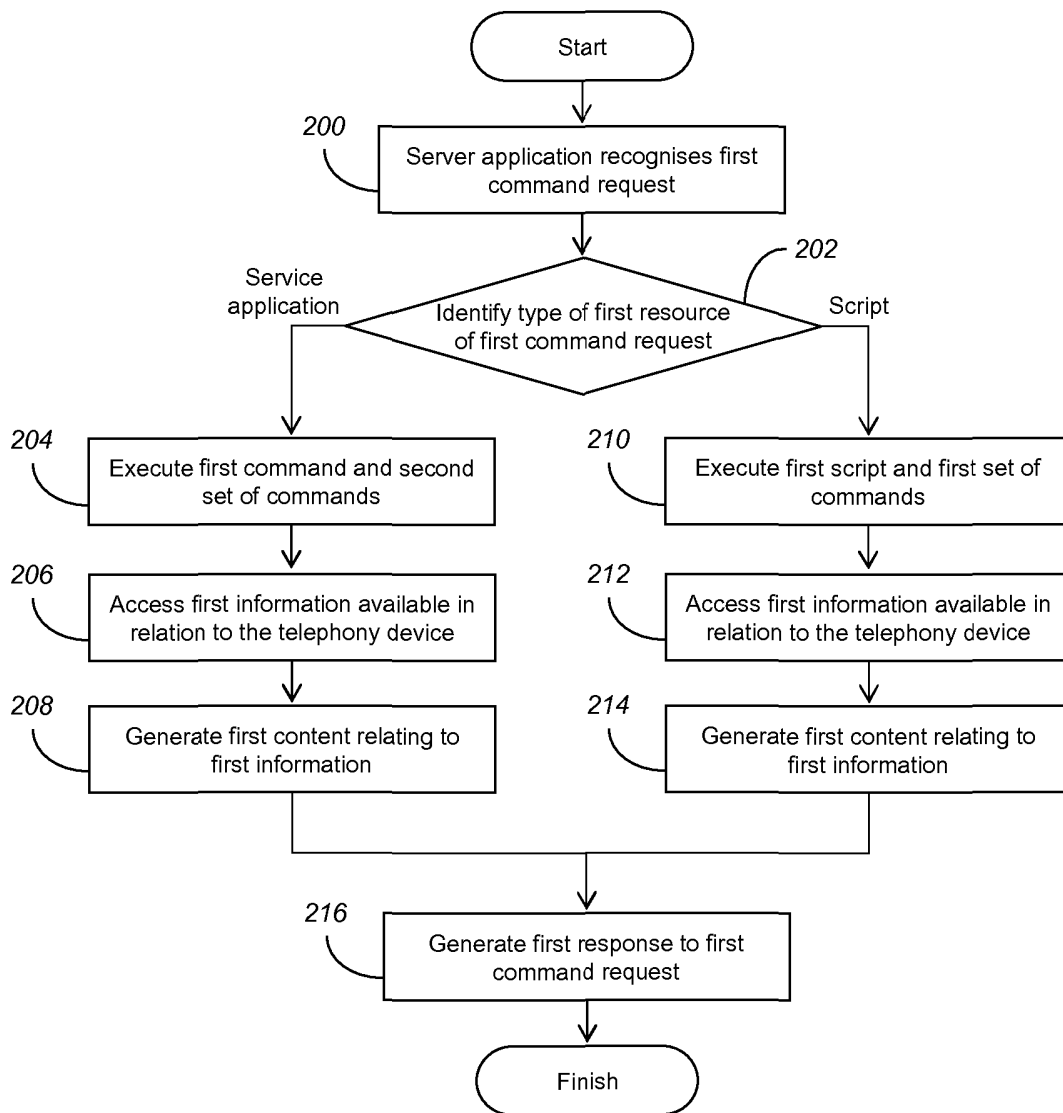


Figure 2a

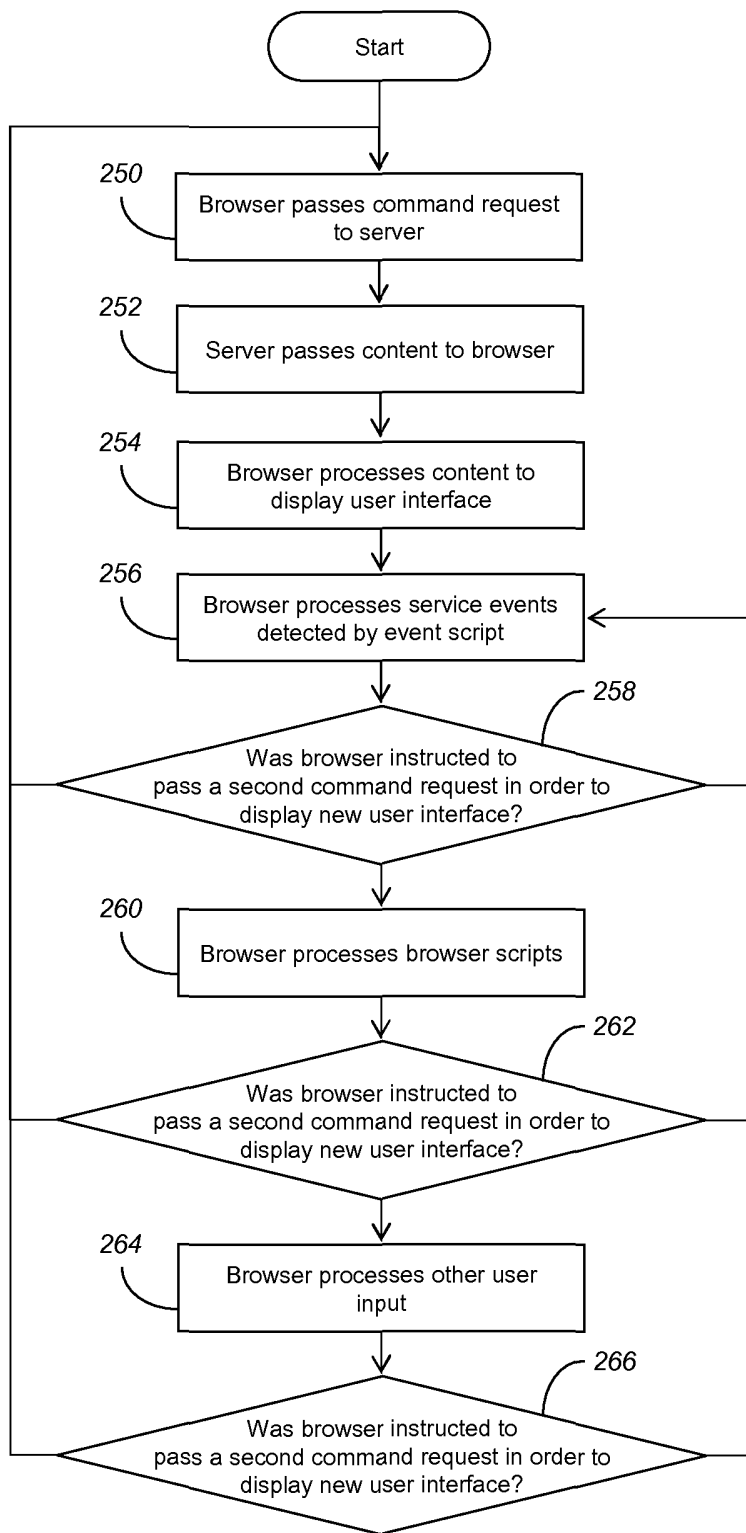


Figure 2b

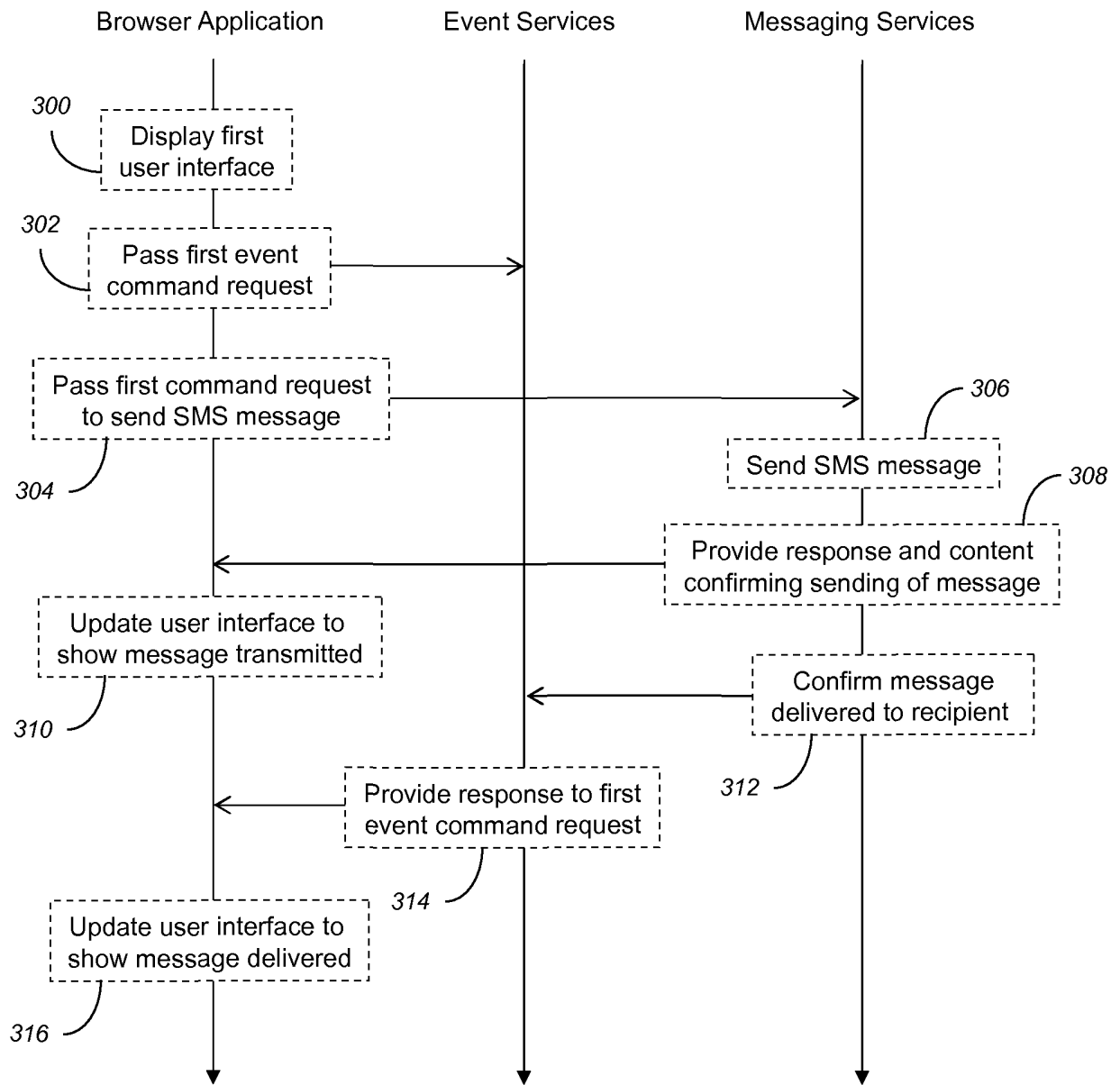


Figure 3

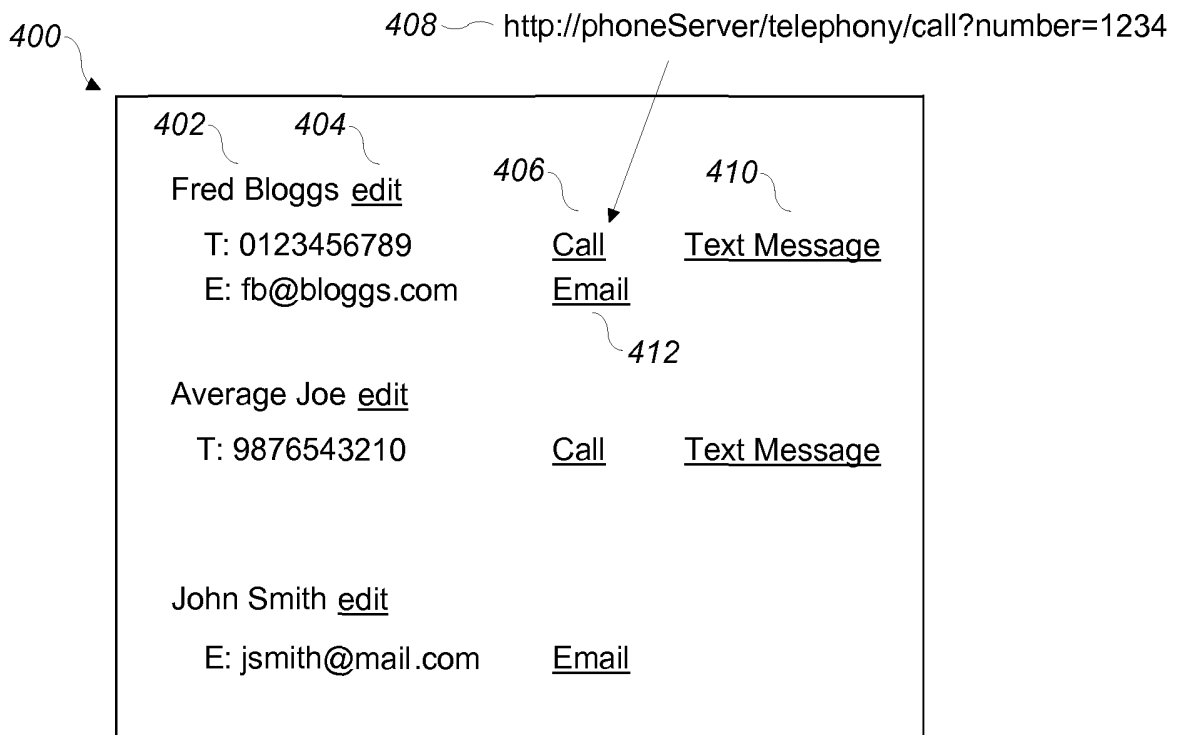


Figure 4

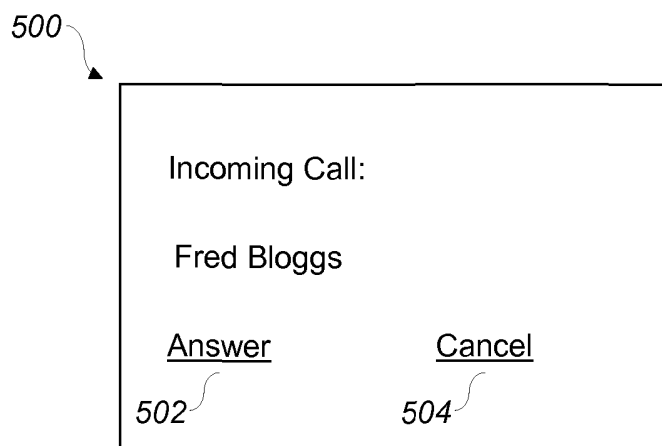


Figure 5

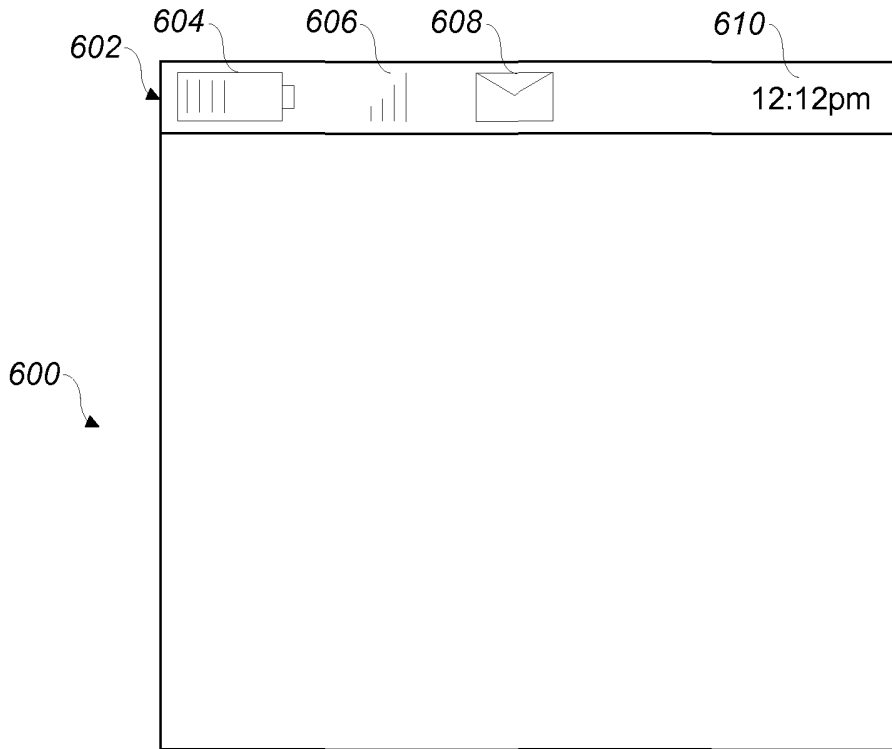


Figure 6

Method and apparatus for accessing services of a device

Field of the Invention

The present invention relates to a method and apparatus for accessing
5 one or more services of a device in order to control the functionality of the
device.

Background of the Invention

Devices are known which comprise a number of components whose
10 functionality can be controlled by a user using one or more input devices of the
device and a user interface displayed on a display of the device. Typically
specialised programs may be configured to provide user interfaces on the
display related to certain aspects of the device only, and may use service
applications configured to control the functionality of certain aspects of the
15 device. Each of these service applications may be configured to use system
calls of an operating system running on the device in order to interact with the
components of the device and so control its functionality. For example the
device may be a telephony device that comprises a specialised program for
providing a user interface relating to a telephone call which may use a telephony
20 services application to control a telephone call, and/or may comprise a
specialised program for providing a user interface relating to a list of contacts
and a contacts services application for controlling a list of contacts stored in the
telephony device, etc. Other exemplary devices include a television, set-top
box, media player, e-book reader, or digital camera.

25 These devices may each comprise an internet browser that may be
configured to provide access to internet websites. The internet browser may be
configured to use system calls of the operating system in order to interact with
the components of the device, for example it may use a network interface of the
telephony device to communicate with internet websites via a communications
30 network and may then display content retrieved from the internet website on the
display of the device.

The specialised programs that provide user interfaces for controlling functions of such a device may typically be configured to use specific methods for accessing the service applications of the device that may not be transferable for use in other devices. This causes a low degree of separation between the specialised programs and the service applications which makes it difficult to separate the design of the user interface for a device from the design of the programs for controlling functions of the device. Additionally further specialised means for accessing the service applications must be developed if the device is to be controlled remotely, for example using a computer or another device. This means that it is difficult and costly to develop means for remotely controlling a device.

The present invention aims to provide improvements to these shortcomings in the prior art by providing a device allowing access to its services via command requests transferred via a standard protocol that are handled by a server application.

Summary of the Invention

In accordance with one aspect of the present invention, there is provided a device comprising a display and a processor, wherein the device is configured to run a browser application and one or more service applications using the processor, and wherein the browser application is configured to display a user interface on the display,

wherein the browser application is configured to receive a first response to a first command request, and in response to receiving the first response, to display a first user interface on the display which includes first content;

wherein the device further comprises a server application which is configured to recognise said first command request, said first command request being provided to the server application using a first transfer protocol, and in response to the first command request, to provide instructions to a service application to execute a first set of one or more commands in order to control a function of the device,

wherein said service application is configured to, in response to receipt of said instructions from the server application, control a function of the device and generate first content for receipt by the server application,

5 wherein the server application is configured to, in response to receipt of the first content from said service application, generate said first response to the first command request, the first response comprising the first content.

By the device being configured to run a browser application and a server application as above it is possible to both control the device and provide a user interface for the device by making command requests between the browser application and the server application via the first transfer protocol, where this first protocol may be a well known transfer protocol. Thus the control of the device and provision of its user interface may not be via a specialised technique but via a standard protocol that hence allows the user interface to be easily modified and/or transferred to other devices. The control of the device and provision of its user interface can be via the server application being passed a command request locally (i.e. from the browser application) or from an external entity such as an external computing device. If the server application is passed the command request from an external computing device a first response may still be passed to the browser application so that a first user interface may be displayed on the display of the device.

10
15
20

In a particularly advantageous embodiment the first transfer protocol comprises HyperText Transfer Protocol (HTTP) or HTTP Secure (HTTPS).

Commands that may be used for controlling the device may thus be accessible by making command requests via a well known transfer protocol such as HTTP or HTTPS.

25

In a particular embodiment the device comprises a plurality of components connected to a system bus, wherein the device further comprises an operating system and wherein in order to control a function of the device said service application is configured to use the operating system to change the state of at least one of the first plurality of components.

30

Control of the components of a device, which may include a network interface, Global Positioning System (GPS) receiver and/or a video broadcast receiver may thus be achieved using command requests passed to the server application.

5 In a particular embodiment the service application provides event services and the service application is configured to delay providing said first content until a service event occurs in relation to a service application of the one or more service applications.

10 The service applications may thus 'push' data via the first transfer protocol to the entity that passed the first command request to the server application when a service event occurs, rather than responding directly to the request. Service events may include events triggered by components of which the device is comprised, and may for example include an incoming telephone call received by the device.

15 In accordance with a further aspect of the present invention, there is provided a system comprising a first device and a second device, the first device comprising a network interface for conducting communications via a communications network and being connected to the second device via the communications network, wherein the second device comprises a display, one
20 or more user input means and a processor, and the first device comprises a processor configured to run one or more service applications,

 wherein said second device is configured to, in response to receiving first user input from the one or more user input means, transmit a first command request to the first device using a first transfer protocol via the communications
25 network,

 wherein the first device further comprises a server application which is configured to recognise said first command request and in response to the first command request, to provide instructions to a service application to execute a first set of one or more commands in order to control at least one of a telephony
30 function, a messaging function, a connectivity function and an event function of the first device,

wherein said service application is configured to, in response to receipt of said instructions from the server application, control at least one of a telephony function, a messaging function, a connectivity function and an event function of the first device and generate first content for receipt by the server application,

5 wherein the server application is configured to, in response to receipt of the first content from said service application, generate a first response to the first command request, the first response comprising the first content, and transmit the first response to the second device,

10 wherein said second device is configured to, in response to receiving the first response, display a first user interface on the display of the second device which includes the first content.

The control of the telephony functions of the first device may thus be via the server application being passed a command request from the second device via a standard protocol. A user interface for controlling the first device may be provided externally to the first device such as on the display of the second device using content provided by the service application of the first device.

15 In accordance with a further aspect of the present invention, there is provided a method for transmitting a command request to a first device from a second device and receiving a response comprising content, wherein the first device comprises a network interface for conducting communications via a communications network and is connected to the second device via the communications network, wherein the second device comprises a display, one or more user input means and a processor, and the first device comprises a processor configured to run a server application and one or more service applications, comprising:

25 receiving at the second device first user input from the one or more user input means, and in response transmitting a first command request to the first device using a first transfer protocol via the communications network;

30 recognising at the server application the first command request, and in response the server application providing instructions to a service application to

execute a first set of one or more commands in order to control at least one of a telephony function, a messaging function, a connectivity function and an event function of the device;

5 in response to receipt of said instructions from the server application said service application controlling at least one of a telephony function, a messaging function, a connectivity function and an event function of the first device and generating first content for receipt by the server application;

10 in response to receipt of the first content from said service application the server application generating a first response to the first command request, the first response comprising the first content, and transmitting the first response to the second device; and

in response to receiving the first response comprising first content the second device displaying a first user interface on the display of the second device which includes the first content.

15 In accordance with a further aspect of the invention there is provided a telephony device comprising a display and a processor, wherein the telephony device is configured to run a server application and one or more service applications using the processor, and comprises a network interface for conducting communications via a communications network,

20 wherein the server application is configured to recognise a first command request provided to the server application using a first transfer protocol via the communications network, and in response to recognising the first command request, to provide instructions to a service application to execute a first set of one or more commands in order to control a function of the telephony device,

25 wherein said service application is configured to, in response to receipt of said instructions from the server application, control a function of the telephony device and generate first content for receipt by the server application,

30 wherein the server application is configured to, in response to receipt of the first content from said service application, generate a first response to the first command request, the first response comprising the first content and

transmit the first response to the first command request using the first transfer protocol via the communications network.

The control of the functions of the telephony device may thus be via the server application being passed a command request via a standard protocol either locally or remotely. First content may be provided to the entity that
5 passed the server application the command request in the first response so that the entity may display a user interface.

Further features and advantages of the invention will become apparent from the following description of preferred embodiments of the invention, given
10 by way of example only, which is made with reference to the accompanying drawings.

Brief Description of the Drawings

Figure 1 schematically illustrates the components and communication
15 links of a telephony device in accordance with an embodiment of the invention.

Figure 2a illustrates steps performed by a server application of the telephony device of Figure 1 in accordance with an embodiment of the invention.

Figure 2b illustrates steps performed by a browser application of the
20 telephony device of Figure 1 in accordance with an embodiment of the invention.

Figure 3 illustrates steps performed by a browser application, event services and messaging services of the telephony device of Figure 1 in accordance with an embodiment of the invention.

Figure 4 shows an exemplary screen relating to a list of contacts
25 displayed by the browser application of the telephony device of Figure 1.

Figure 5 shows an exemplary screen relating to an incoming telephone call displayed by the browser application of the telephony device of Figure 1.

Figure 6 shows part of an exemplary screen that includes a status bar
30 displayed by the browser application of the telephony device of Figure 1.

Detailed Description of the Invention

Figure 1 schematically illustrates the components and communication links of a telephony device 100, which is an exemplary device used to illustrate the features of the present invention. The telephony device 100 includes a processor 102 that is able to transmit control messages to, receive status information from, and transmit data to and from components within the telephony device 100 that are connected to a system bus 104, where these components may include a non-volatile storage device 106, random access memory 108, user input interface 110, network interface 112 and graphics processing component 114. The processor 102, which in this embodiment is a microprocessor, processes instructions stored in the random access memory (RAM) 108 that have been loaded from the non-volatile storage device 106 which could be for example a flash memory or a hard disk drive. These instructions are in the form of computer software in the form of one or more programs that implement an operating system 118, a server application 120, a browser application 122, one or more service applications 124, and other programs identified below. The RAM 108 is also used by programs running on the processor 102 as a means of storing and accessing data in the form of electronic signals where the data is used during the execution of the programs.

The operating system 118 is computer software in the form of a program or set of programs whose instructions are loaded from non-volatile storage 106 by the processor 102 and executed when the telephony device 100 is turned on. The operating system 118 may start further programs automatically and/or may allow a user to start further programs, for example by the user using a user interface provided by the telephony device 100. The operating system 118 enables the sharing of the processing power provided by the processor 102 between the programs (e.g. 118, 120, 122, 124) running on the processor 102.

The operating system 118 provides a programmatic interface for programs running on the processor 102 allowing them to request functionality from the operating system 118. This programmatic interface may take the form of procedures, i.e. system calls, which a program running on the processor 102

may use in order to invoke the operating system 118 and request it to provide desired functionality. In response to receiving a request for functionality the operating system 118 may transmit control messages to, receive status information from, transmit data to and/or receive data from components (e.g. 5 106, 108, 110, 112, 114, 116) connected to the system bus 104 in order to provide the requested functionality, and may also return data to the requesting program as a result.

The telephony device 100 includes a graphics processing component 114 that is able to render graphics in accordance with commands made by programs 10 running on the processor 102 and output these to a display device 116 which may reside within the telephony device 100. In alternative embodiments of the invention the display device 116 may be an external component connected to the telephony device 100 via a composite video, component video, Video Graphics Array, Digital Visual Interface, or High-Definition Multimedia Interface 15 connection.

Programs running on the processor 102 can process user input obtained from a user input interface 110 that receives user input from a user input device or devices (not shown). The user input devices may include a touch screen interface that may be incorporated within the display device 116 (i.e. such that 20 this is a touchscreen display device 116). Alternatively or in addition user input devices may include a keypad, keyboard, mouse and/or remote control which may be incorporated within the telephony device 100 or may be connected to it.

The telephony device 100 also includes a network interface 112 (or a plurality of such interfaces) that allows programs running on the processor 102 25 to transmit and receive data to and from a number of other devices and systems via a communications network 140 (or a plurality of such networks).

The network interface 112 (or the plurality of such interfaces) may include a modem and/or an Ethernet card or interface for use with a corresponding communications network (or networks) 140 such as the Internet 30 and/or a private data communications network.

The network interface 112 (or the plurality of such interfaces) may include a radio access network interface (or a plurality of such interfaces) for use with a corresponding communications network 140 (or a plurality of such networks) that may be a radio access network. Exemplary radio access networks to which the network interface 112 (or plurality of such interfaces) may be able to connect include Global System for Mobile Communications (GSM) networks, Universal Mobile Telecommunications System (UMTS) networks, Long Term Evolution (LTE) networks, fixed wireless access (such as IEEE 802.16 WiMax), and wireless networking (such as IEEE 802.11 WiFi). These radio access networks 140 may also provide access to the Internet.

The operating system 118 may include a networking program that allows communication between programs running on the processor 112 of the telephony device 100 using networking protocols such as the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP), as well as communication between programs running on the processor 102 and external devices to which data can be transmitted to and received from via the communications network 140 (or plurality of such networks). External devices to which data can be transmitted to and received from via the communications network 140 may include an external computing device 150, which may for example be a personal computer or a computer server with a network interface configured to allow the external communications device to connect to the communications network 150.

The telephony device 100 includes one or more service applications 124 that provide in one or more computer programs programmatic functions and procedures for operating the telephony device 100. These programmatic functions and procedures may use the programmatic interface made available by the operating system 118 to operate the components of the telephony device 100 that are connected to the system bus 104. These programmatic functions and procedures thus form a first plurality of commands that may be used to control the functions of the telephony device 100, and in doing so may change the state of one or more of the components of the telephony device 100.

The first plurality of commands of the service applications 124 may be made available for use by other programs running on the processor 102 by way of the following exemplary methods. For example a first command of the first plurality of commands may be called by a first program running on the processor by the first program instructing the operating system 118 to start the appropriate service application 124 with a particular set of parameters specifying the first command. Alternatively, the first program may communicate with an instance of the appropriate service application 124 running on the processor 102 using a method for inter-process communication made available by the operating system 118, such as a pipe or shared memory, in order to instruct the instance to execute the first command. In either case, the result of an executed command may be returned to the requesting program from a service application 124 by using an appropriate method of inter-process communication. The first plurality of commands of the service applications 124 may alternatively or in addition be made available for use by other programs running on the processor 102 via a common interface, such as the Common Gateway Interface (CGI) or the Fast Common Gateway Interface (FastCGI).

The service applications 124 may include telephony services 126, messaging services 128, contacts services 130, media services (not shown), social networks services (not shown), configuration services 134, connectivity services 136 and event services 138 and data sharing services (not shown), each of which may form one of the programs which the one or more service applications 124 may include.

The telephony services 126 of the telephony device 100 provide commands for controlling the creation and receipt of telephone calls using the telephony device 100 via the network interface 112 (or a plurality of such interfaces) and corresponding communications network 140 (or a plurality of such networks). The telephony services 126 may change the state of the network interface 112 of the telephony device 100 by, for example, using the network interface to create or receive telephone calls.

The messaging services 128 of the telephony device 100 provide commands for controlling the sending and receiving of messages via the network interface 112 (or a plurality of such interfaces) and corresponding communications network 140 (or a plurality of such networks) to other devices. Such messages may include Short Message Service (SMS) messages, Multimedia Messaging Service (MMS) messages and e-mails. The messaging services 128 may also provide commands for editing, storing and retrieving previously sent, received or drafted messages. The messaging services 128 may change the state of the network interface 112 of the telephony device 100 by, for example, using the network interface to send or receive a text message.

The contacts services 130 of the telephony device 100 provide commands for editing, storing and retrieving a list (or a plurality of lists) of contacts and for editing, storing and retrieving entries in that list (or lists), where each entry may include the name and contact details (such as telephone number, e-mail address, etc.) of a contact.

The media services of the telephony device 100 provide commands for storing and retrieving multimedia files to which access is available on the telephony device 100 (e.g. stored in non-volatile storage 106) or to which access is available via the communications network 140. These multimedia files may include sound recordings, music files, image files, and/or video files. The media services may also allow the editing, recording and/or playback of these multimedia files.

The social networks services of the telephony device 100 provide commands for accessing one or more social networks such as MySpaceTM, FacebookTM, TwitterTM, etc. via the communications network 140. Accessing social networks may include authenticating (i.e. logging in) the user of the telephony device 100 with some or all of the social networks using user-entered or pre-stored (e.g. in non-volatile storage) authentication information, and editing, sending and receiving messages, multimedia files, contact information and other information using those social networks. The social networks services may change the state of the network interface 112 of the telephony device 100

by, for example, using the network interface to send or receive data to or from a social network in order to conduct the above activities.

The location services 132 of the telephony device 100 provide commands for controlling the receipt and use of location information using a location receiver device (not shown) of which the telephony device 100 may be comprised and which may be connected to the system bus 104 of the device. The location receiver device may for example be a Global Positioning System (GPS) receiver.

The configuration services 134 of the telephony device 100 provide commands for editing, storing and retrieving settings for the telephony device 100, which may include settings for components connected to the system bus 104, and/or the operating system 118, server application 120, browser application 122, service applications 124, and/or other programs identified below.

The connectivity services 136 of the telephony device 100 provide commands for connecting the network interface 112 (or plurality of such interfaces) of the telephony device 100 to an available communications network 140 (or plurality of such networks) and for configuring any such connections to communications networks 140 (or plurality of such networks). The connectivity services 136 may change the state of the network interface 112 of the telephony device 100 by, for example, connecting or disconnecting the network interface from a communications network 140.

The event services 138 of the telephony device 100 provide commands allowing other service applications 124 to notify the occurrence of service events which may occur in relation to the one or more service applications 124. Service events may include events triggered by one of the components connected to the system bus 104 or by the operating system 118 that are processed by one or more of the service applications 124. Examples of service events may include an incoming telephone call detected by the network interface 112 which is processed by the telephony services 126, or the activation by the operating system 118 of an alarm at a time set by the user which is processed by

the configuration services 134, or receiving a new device location from the location receiver device (not shown) that is processed by the location services 132.

When a service application 124 receives a new service event it may
5 notify the occurrence of the service event to the event services 138 by passing information relating to the new service event to the event services 138. Information may be passed between the service applications 122 in this way by using a method for inter-process communication, and alternatively or in addition information may be passed between service applications 122 using a common
10 interface, such as using pipes, sockets, shared memory, the Common Gateway Interface (CGI) or the Fast Common Gateway Interface (FastCGI). The event services 138 may then provide the information to other entities that have, via a command of the event services 138, requested to receive information in relation to service events, as is explained in greater detail below.

15 The data sharing services of the telephony device 100 provide commands that allow the sharing of data between the telephony device 100 and one or more other devices via the communications network 140. The data sharing services may for example allow the passing of messages between a program or a script running in the browser application 122 of the telephony
20 device 100 and programs or scripts running in browsers of other devices. These messages may be used to communicate between programs or scripts implementing a game involving multiple players that is being played by the user of the telephony device 100 and users of the other devices, such as for example a card game. Alternatively the messages may be used to share files between the
25 telephony device 100 and other devices. The data sharing services may for example use the event services 138 to notify the receipt of a message from the user of the telephony device 100 or from another device to programs or scripts running in the browser application 122 on the telephony device 100 and/or programs or scripts running in browsers of other devices, so that for example
30 new actions made by a player in a game involving multiple players may be

received by the programs and scripts running the game on both the telephony device and the other devices.

When a command in the first plurality of commands is used (i.e. executed) by another program running on the processor 102 the state of the telephony device 100 may be changed from one state to another. For example
5 the state of the telephony device 100 may be changed by executing one of the commands of the telephony services 126 for making or receiving a telephone call, i.e. the state of the telephony device 100 may be changed from 'not in a call' to 'in a call'. Other examples of commands in the first plurality of
10 commands that may change the state of the telephony device 100 include sending a message using the messaging services 128, editing the details of a contact using the contacts services 130, changing the configuration of the device using the configuration services 134, etc.

In order for a user of the telephony device 100 to use the functionality of
15 the telephony device 100 embodied in the first plurality of commands of the one or more service applications 124, a user interface is provided that is displayed on the display device 116 by the graphics processing device 114. The user may use the user input device or devices to interact with this user interface and so control the functions of the telephony device 100. In preferred embodiments of
20 the invention the display of and functionality provided by this user interface is managed by both the server application 120 and browser application 122 that may run on the processor 102 of the telephony device 100.

The server application 120 may be a web server e.g. a HyperText Transfer Protocol (HTTP) server that may use the service applications 124 to
25 generate content using information available in relation to the telephony device 100 in order to provide a user interface. The service applications 124 may therefore comprise a set of web services that are accessible via the server application 120. The generated content may in preferred embodiments of the invention be provided to the browser application 122 which will process the
30 content and display it on the display device 116 by appropriately instructing the graphics processing component 114 in order to display a user interface.

Figure 2a illustrates the steps that may be performed by the server application 120 in order to execute one or more commands from the first plurality of commands of the service applications 124 and to provide content that may include information relating to a user interface.

5 Initially the server application 120 may recognise a first command request (step 200). The server application 120 may be configured to recognise command requests provided to it using a first transfer protocol, for example using HTTP. The first command request may be received by the server application 120 from the browser application 122 running on the telephony device 100, or may be received from an external computing device 150.

The server application 120 may receive the first command request from the browser application 122 by the browser application 122 passing the first command request to the server application 120 using a method for inter-process communication provided by the operating system 118 of the telephony device. For example, the first command request may be passed from the browser application 122 to the server application 120 over TCP or UDP by the networking program of the operating system 118.

The server application 120 may alternatively receive the first command request from an external computing device 150 by the external computing device 150 transmitting the first command request over a communications network 140 to which the telephony device 100 is connected. For example, the first command request may be provided to the server application 120 over TCP or UDP via a communications network 140. The first command request may then be received by the network interface 112 of the telephony device 100 and passed to the server application 120.

Once the server application 120 has recognised a first command request as in step 200, the server application 120 may identify the type of a first resource that the server application 120 may instruct to execute a first set of one or more commands from the first plurality of commands of the service applications 124, in order to control the functions of the telephony device 100 (step 202).

The first command request may be in the form of an HTTP GET request that includes a Uniform Resource Locator (URL) or Uniform Resource Identifier (URI) that identifies the first resource that may be instructed by the server application 120 to execute the first set of one or more commands. The first resource may specify a first service application 124 and a first command, for example by using the URL “/phonebook/Contacts”, where the first service application is the contacts services and the first command is one named “Contacts”. Alternatively the first resource may specify a script that may be instructed by the server application 120 to execute the first set of one or more commands, for example by using the URL “/phonebookApp.php”.

Each of the commands in the first set of one or more commands is a command of the first plurality of commands of the service applications 124, and is executed in response to instructions made by the server application 120 using one of the service applications 124 in either steps 204 and 206 or step 214, depending on the type of the first resource, as is explained in greater detail below.

The first resource identified by the first command request may identify a first service application 124 and a first command from the first plurality of commands which the server application 120 may use to instruct the execution of the first set of one or more commands. The first resource may identify both first the service application 124 and the first command by using a URL that includes the name and path of the first service application 124 and the first command in a HTTP POST request made to the server application 120. For example the first resource may identify the first service application 124 and the first command by using the URL “/messaging/sms” where “/messaging” represents the name of the messaging service 128, and “sms” represents the name of the first command. Data may be passed to the first command in the POST request, for example the phone number and message text to use in sending an SMS message may be specified using the POST data “number=012345&msgTxt=Hi” in the message body of the POST request.

If the server application 120 identifies the type of the first resource as a first service application 124 and a first command in step 202 as above, the server application 120 may proceed by instructing the first service application 124 to execute the first command, by the server application 120 using one of the methods by which the first plurality of commands are made available for execution by other programs (step 204). The server application 120 may for example take advantage of a common interface between it and the first service application 124, such as pipes, sockets, shared memory, CGI or FastCGI, in order to instruct the first service application 124 to execute the first command.

10 The first command may, when executed by the first service application 124, execute a second set of commands belonging to the first plurality of commands of the service applications 124. The first set of one or more commands executed in response to the instructions of the server application 120 may thus include the first command and the second set of commands, if the type of the first resource is identified as a first service application 124 and a first command in step 202. The first set of one or more commands may control the functions of the telephony device 100, as described above for the first plurality of commands of the service applications 124.

20 In response to executing the first set of one or more commands in step 204 the first service application 124 may access first information available in relation to the telephony device 100 (step 206). The first information may include a set of one or more portions of information returned by the execution of the first set of one or more commands in step 204. Each command in the first set of one or more commands executed in step 204 may return one of the portions of information in the set of one or more portions of information, where each portion of information may indicate the success or failure of the execution of the command. Additionally each portion of information may return data regarding the status of the telephony device 100, and/or data stored on the telephony device 100 (e.g. in RAM 108 or in non-volatile storage 106) and/or data received via the communications network 140. Examples of the first information may include one or more entries from a list of contacts maintained

by the contacts services 130, one or more messages received by the telephony device 100 such as SMS messages or emails received by the messaging services 128, and/or one or more multimedia files provided by the media services.

5 If the second set of commands executed in step 206 included at least one command, portions of information resulting from the execution of the second set of commands may be processed during the execution of the first command and used to form the first information along with the portion of information resulting from the execution of the first command. Otherwise if the second set commands executed in step 206 was empty the first information may be gathered and
10 processed using the portion of information resulting from the execution of the first command only.

The first service application 124 may then use the service application 124 that executed the first command in the first set of one or more commands in step 204 in order to access the first information which is returned by the
15 execution of the first command.

The first service application 124 then generates first content relating to the first information (step 208). The first information accessed as a result of the execution of the first set of one or more commands may be processed by the service application 124 that executed the first command and may then be used to
20 form the first content that is generated by that service application 124, for example by the service application 124 representing the first information as programmatic information in a structured data format such as JavaScript Object Notation (JSON) or eXtensible Mark-up Language (XML). This programmatic information may include the first information, or it may include processed
25 information resulting from the processing of the first information. The programmatic information may be processed later by a program (such as the browser application 122, or a script interpreted by the server application 120, or a suitable program running on the external computing device 150) that receives the first content in order to update a first user interface.

30 The first content may be returned to the server application 120 from the service application 124 by using an appropriate method of inter-process

communication, for example the service application 124 may take advantage of a common interface between it and the server program 120 such as pipes, sockets, shared memory, CGI or FastCGI.

5 The server application 120 may alternatively identify the type of the first resource of the first command request as a script in step 202. A first script may be identified as the first resource of a command request by for example using the URL “/phonebookApp.php” in an HTTP GET request made to the server application 120. Data may be passed to the server application 120 for use by the first script by using parameters in the URL of the first resource, for example by
10 using the URL “/phonebookApp.php?sort=byName”. The first script may be in the form of one or more computer programs that may be instructed by the server application 120 to execute the first set of one or more commands.

The server application 120 may use an interpreter program to load and interpret the first script in order to instruct the script to execute the first set of
15 one or more commands (step 210). Scripts that the server application 120 may be configured to use in this way may include PHP: Hypertext Preprocessor (PHP) scripts and/or JavaServer Pages (JSP) scripts and/or Ruby on Rails scripts, which may each be executed by an appropriate interpreter program. The server application 120 may request that an interpreter program interpret the first
20 script by instructing the operating system 118 to start the interpreter program with a particular set of parameters specifying the first script. Alternatively the server application 120 may communicate with an instance of the interpreter program running on the processor 102 using a method for inter-process communication made available by the operating system 118, for example the
25 server application 120 may take advantage of a common interface between it and the appropriate interpreter program such as pipes, sockets, shared memory, CGI or FastCGI, in order to instruct the instance of the interpreter program to interpret the first script.

30 During interpretation of the first script by an interpreter program one or more commands in the first plurality of commands may be executed in order to control the functions of the telephony device 100. The execution of one or more

commands in the first plurality of commands may be requested by using code in the first script that passes one or more command requests to the server application 120. For example the script may be configured to pass a command request in the form of an HTTP GET request to the server application 120 by using a method for inter-process communication provided by the operating system 118 of the telephony device such as via TCP or UDP and the networking program of the operating system 118. Alternatively the script may be configured to pass command requests to the server application 120 via a common interface between it and the server application 120, such as pipes, sockets, shared memory, CGI or FastCGI.

Each command request made by the first script to the server application 120 may identify a resource that is a service application 124 and a command. The server application 120 may instruct the service application 124 execute the command by the server application 120 using one of the methods by which the first plurality of commands are made available for execution by other programs. The server application 120 may for example take advantage of a common interface between it and the appropriate service application 124, such as pipes, sockets, shared memory, CGI or FastCGI, in order to instruct the service application 124 to execute the command. The first set of one or more commands may thus include each of the commands that are executed by the server application 120 as a result of command requests made to it in the first script in this way.

A command request made by the first script may identify a resource that is a second script which may be used by the server application 120 to execute a further set of commands (i.e. by the second script making command requests identifying service applications 124 that may be used to execute commands and/or further scripts). Thus the first set of one or more commands may also include sets of commands executed by service applications 124 specified in command requests made by further scripts that are interpreted as a result of command requests made by the first script.

In response to executing the first set of one or more commands in step 210 one or more service applications 124 may access first information available in relation to the telephony device 100 (step 212). The first information may include a set of one or more portions of information returned by the execution of the first set of one or more commands in step 210. Each command in the first set of one or more commands executed in step 210 may return one of the portions of information in the set of one or more portions of information, and each portion of information may therefore indicate the success or failure of the execution of the command. Additionally each portion of information may return data regarding the status of the telephony device 100, and/or data stored on the telephony device 100 (e.g. in RAM 108 or in non-volatile storage 106) and/or data received via the communications network 140. Examples of the first information may include one or more entries from a list of contacts maintained by the contacts services 130, one or more messages received by the device such as SMS messages or emails received by the messaging services 128, and/or one or more multimedia files provided by the media services.

For each command in the first set of one or more commands, the service application 124 that executed that command may access the portion of information returned by the execution of that command. Thus the first information accessed by the server application 120 is formed of the portions of information returned to the service application(s) 124 in response to the execution of the first set of one or more commands.

One or more service applications 124 may then generate first content relating to the first information (step 214). The first information accessed in step 212 as a result of the execution of the first set of one or more commands will first be passed to the server application 120. The first information may be passed to the server application 120 from each service application 124 by using an appropriate method of inter-process communication, for example the service application 124 may take advantage of a common interface between it and the server program 120 such as pipes, sockets, shared memory, CGI or FastCGI.

The first information may be formed of one or more portions of information obtained as a result of the execution of each command in the first set of one or more commands. Each of these portions of information may be passed separately to the server application 120 in a structured data format such as JSON or XML, and the server application 120 may then pass each portion of information to the script that made the command request which resulted in the accessing of that portion of information, i.e. either the first script specified in the first command request or further scripts interpreted in response to command requests made by the first script, as appropriate. Alternatively, each portion of information may be passed separately to the server application 120 from each service application 124 in the message body of an HTTP POST request, where the portion of information may be encoded using URL encoding (i.e. 'percent encoding'), as comma separated values (CSV) or in a packed binary format.

These portions of information may be returned to the appropriate script from the service application 124 by using an appropriate method of inter-process communication, for example the service application 124 may take advantage of a common interface between it and the interpreter program (i.e. that is interpreting the script) such as pipes, sockets, shared memory, CGI or FastCGI.

Each script that receives a portion of information may then process that portion of information and use it to generate a portion of content. For example the portion of information which is represented in a structured data format might be processed, and/or formatted and inserted by the script that is processing it into an HTML document or used to determine the appearance of that document.

Each of the portions of content that are created by any further scripts interpreted in response to command requests made by the first script may be passed to the first script by first passing each of those portions of content to the server application 120 and the server application 120 then passing those portions of content to the first script. The passing of these portions of content between the further scripts, server application 120 and first script may be via an appropriate method of inter-process communication, for example via a common interface between the server application and the interpreter programs (i.e. that

are interpreting the scripts) such as pipes, sockets, shared memory, CGI or FastCGI.

In order to create the first content the portions of content that are created by the first script are used in conjunction with any portions of content created by any further scripts interpreted in response to command requests made by the first script. The first content may be returned to the server application 120 from the first script by using an appropriate method of inter-process communication, for example the interpreter program running the script may take advantage of a common interface between it and the server program 120 such as pipes, sockets, shared memory, CGI or FastCGI.

First content generated by a first resource that is a script in this way may include appearance information, programmatic information, and command information that may be used in order to provide a first user interface. The first information used in the process of generating this content may be used in generating each of these types of information.

The appearance information contained in the content may include display data relating to the appearance of a first user interface, such as display data that relates to the look and feel of the first user interface including the look and feel and layout of content and objects within the first user interface. The display data may also relate to the first information available in relation to the device accessed by the server application in step 212. For example the first information may be the name of a contact that is making an incoming call, and the display data may include the first information so that the name of the contact may be displayed in the first user interface. In another example the first information may include the time of day and the display data may set the look and feel of the first user interface by including within the display data colours, icons (e.g. a sun or moon icon), etc. to be used in the first user interface in order to indicate the time of day.

The appearance information may be defined in a mark-up language such as Hyper Text Mark-up Language (HTML). This appearance information may be processed later by a program (such as the browser application 122 or a

suitable program running on the external computing device 150) that receives the first content in order to display a first user interface.

The programmatic information contained in the content may be in the form of one or more scripts configured to manage the first user interface, and may be in the form of a scripting language such as JavaScript or Visual Basic Script (VBScript) that is embedded within HTML content or instead may be in a separate JavaScript file. These scripts may be interpreted later by a program (such as the browser application 122 or a suitable program running on the external computing device 150) that receives the first content in order to update and/or manage a first user interface. The scripts may be used to manage the display of a user interface in response to user input, or in response to information received from the server application 120.

Additionally the programmatic information contained in the content may include the first information accessed in step 212, or it may include processed information resulting from the processing of the first information. For example the programmatic information may be embedded in HTML content and/or represented in JavaScript Object Notation (JSON) or eXtensible Mark-up Language (XML), as comma-separated values (CSV), or encoded using URL encoding (i.e. 'percent encoding'), or in another structured data format.

The programmatic information may be processed later by a program (such as the browser application 122 or a suitable program running on the external computing device 150) that receives the first content in order to update a first user interface.

The command information contained in the content may include a set of command requests that may be used in response to interactions with the user interface made by a user, or in response to service events, in order to retrieve further content or update the user interface, as in explained in further detail below. The command information may be in the form of hyperlinks within HTML content, and/or may be in the form of instructions within the scripts of the programmatic information to use one or more command requests to retrieve content. This command information may be processed later by a program (such

as the browser application 122 or a suitable program running on the external computing device 150) that receives the first content in order to allow the program to make further command requests in order to display a second user interface and/or in order to update a first user interface.

5 Finally, the server application 120 generates a first response to the first command request, the first response including the first content generated in either step 208 or step 214 (step 216). The first response to the first command request may be made using the same transfer protocol by which the first command request was provided to the server application 120, for example the
10 first response could be an HTTP response to the HTTP request that includes the first command request, where the first command request was made by either the browser application 122 or an external computing device 150 as described above.

 The first response to the first command request is either passed (in the
15 case of the browser application 122) or transmitted (in the case of the external computing device 150) to the originator of the first command request in order for the originator of the request to process the first content in the first response.

 If the server application 120 received the first command request from the browser application 122 as defined in step 200 above, the first response is
20 passed back to the browser application 122 in order for the browser application 122 to process the first content in the first response in order to display on the display device 116 a first user interface which is indicative of the first information which the first content may include. For example, the first response may be passed back to the browser application 122 from the server application
25 120 over TCP or UDP by the networking program of the operating system 118.

 If the server application 120 received the first command request from an external computing device 150, the first response is transmitted by the server application 120 back to the external computing device 150 via a communications network 140 to which the telephony device 100 is connected
30 (i.e. by the server application 120 using the network interface 112). For

example, the first response may be transmitted back to the external computing device 150 over TCP or UDP via a communications network 140.

The server application 120 may also be arranged to provide static content retrieved from files that are specified in command requests made to the server application 120 by either the browser application 122 or the external computing device 150. Static content may not change over time e.g. from one command request to another. Such content might include for example HTML documents, Cascading Style Sheets (CSS) documents JavaScript programs, and/or multimedia files that may be requested from the server application 120 in a command request provided to it using HTTP (e.g. using a URL that includes the name and path of the required document e.g. “/help/index.html” in a HTTP GET request). The server application 120 may provide this static content in order to provide multimedia to enhance a user interface, or to provide static HTML, CSS or JavaScript documents for use in displaying and/or managing a user interface, where the user interface may be displayed in either the browser application 122 or by the external computing device 150.

The browser application 122 may be an internet browser application that is able to request and retrieve content using a transfer protocol, for example using HTTP. Once content has been retrieved by the browser application 122 it processes the required content and uses the graphics processing device 114 to display the processed content on the display device 116 in order to display a user interface.

The browser application 122 may be configured to display user interfaces that are described in a mark-up language, such as for example HTML. The links, buttons and other objects displayed in a particular user interface may be used by a user of the telephony device 100 to control the functionality of the telephony device 100 using the one or more input devices as is explained in more detail below.

Figure 2b illustrates the steps that may be performed by the browser application 122 in order to display a user interface on the display device 116 of the telephony device 100.

Initially the browser application 122 passes a first command request to the server application 120 using a first transfer protocol, for example using an HTTP GET request (step 250). As described from the point of view of the server application 120 in step 200 above, the browser application 122 may provide command requests to the server application 120 by using a method for inter-process communication provided by the operating system 118 of the telephony device. For example, the first command request may be passed from the browser application 122 to the server application 120 over TCP or UDP by the networking program of the operating system 118.

10 The first command request passed by the browser application 122 to the server application 120 in step 250 may include a URL or URI identifying a resource that is a script. For example, when the browser application 122 is first loaded after the telephony device 100 is turned on and the operating system 118 has been started, the browser application 122 may be configured to send a command request to the server application 120 that includes the URL
15 “http://phoneServer/homeScreen.php” in order to allow the browser application 122 to display an initial user interface.

In order to pass a first command request that includes a URL such as “http://phoneServer/homeScreen.php” from the browser application 122 to server application 120, the browser application 122 may recognise that the command request should be made via the HTTP protocol (i.e. due to “http://” in the above URL), may then use the networking program of the operating system 118 to determine that the server application 120 should be passed the command request (e.g. as the networking program may include or use a nameserver
20 function to resolve “phoneServer” into the Internet Protocol (IP) address of the server application 120), and may thus pass the command request to the server application 120.

The server application 120 may then use the script identified in the command request to execute a first set of one or more commands, access first
30 information available in relation to the telephony device 100, and generate first content relating to the first information, in accordance with steps 210 to 214

shown in Figure 2a. The first content is then passed from the server application 120 to the browser application 122 in a first response to the first command request using the first transfer protocol, in accordance with step 214 shown in Figure 2a (step 252).

5 Once the browser application 122 receives the first content from the server application 120, it may process the first content in order to display on the display device 116 a first user interface which is indicative of the first information accessed by the server application in step 212, as is described below (step 254).

10 The first content received by the browser application 122 may include appearance information (which may be encoded in HTML) which the browser application 122 may process in order to display a first user interface consisting of text, hyperlinks, forms, buttons etc.

15 The first content may include programmatic information that includes one or more browser scripts (for example JavaScript) which the browser application 122 may interpret in order to manage the display of the first user interface, for example by re-arranging or updating elements of the first user interface in response to user input.

20 The first content may also include command information which includes a set of command requests. The browser application 122 may be configured to transmit to the server application 120 one or more of the command requests in the set of command requests of the command information in response to the user interacting with the first user interface. For example one of these command requests may be passed to the server application 120 in response to the user
25 selecting a hyperlink or a button, or by submitting a form using the first user interface.

30 The browser scripts in the programmatic information of the first content may also be configured to pass one or more of the command requests in the set of command requests of the command information to the server application 120 in response to events detected by the browser scripts, such as user input via the

user input means or programmatic conditions detected by the one or more browser scripts.

The appearance information may contain instructions to the browser application 122 to make further command requests to the server application 120 in order to retrieve further content such as further appearance information, further programmatic information, and/or further command information which the browser application 122 may also process in order to display the first user interface. Further content may be received in the form of one or more files or documents such as HTML documents, JavaScript documents, and/or multimedia files which may include the information, and may for example be used to display a first user interface that includes a plurality of HTML documents (e.g. using different documents in different frames), and/or to display images in the first user interface, etc.

The browser application 122 may maintain a cache (e.g. in RAM 108 or in non-volatile storage 106) of some of the content and/or information it has previously received from the server application 120 such that it may retrieve some of this content and/or information from the cache rather than requesting it from the server application 122 during later requests for the same content and/or information.

Once the first user interface has been displayed according to step 254, the browser application 122 may execute a service event script contained in the programmatic information in order to detect service events occurring in relation to the service applications 124 of the telephony device 100 (step 256). The service event script may be in the form of JavaScript embedded in the first content received in step 252, or may be a separate JavaScript file received by the browser application 122 in response to a request for further content made by the browser application 122 in step 252. The service event script may be interpreted by the browser application 122 in order to control the update of the first user interface provided by the browser application 122 in response to service events.

The service event script may be configured to detect service events by initially passing a first event command request to the server application 120.

This first event command request passed to the server application 120 may include a URL or URI identifying a resource that is a service application 124 and a first event command. The service application 124 identified in the first event command request may be the event services 138, and the first event
5 command may be a receive new events command of the event services 138. The server application 120 will thus instruct the event services 138 to execute the first event command according to steps 200 to 208 and 216 as shown in Figure 2a. The execution of the receive new events command of the event services 138 will thus return to the server application 120 first new event content, which may
10 include programmatic information represented in a structured data format relating to new service events that have recently occurred. The new event content will then be passed to the service event script being interpreted by the browser application 122 in a first new event response. If no new service events have occurred the receive new events command of the events service 138 may
15 not return any content to the server application 120 until a new service event has occurred, such that the server application 120 may not provide the first new events response until a new service event has occurred.

The service event script may then use the programmatic information contained in the first event response to update the first user interface, for
20 example the programmatic information may indicate that the network interface 112 of the telephony device 100 has connected to a communications network 140 such as a wireless network, which may be displayed in the first user interface by showing an icon or image in a particular area of the first user interface. The service event script may then pass a second event command
25 request to the server application 120 in order to detect further service events.

In order for the first event script to pass command requests to the server application 129 to update the first user interface in this way, the first event script and server application 120 may operate according to the Asynchronous
30 JavaScript and XML (AJAX) framework in order to update the display and behaviour of the first user interface in the background, i.e. without the browser

application 122 needing to entirely re-load the first user interface from the server application 120.

The programmatic information contained in the first event response may indicate that the browser application 122 should pass a second command request to the server application 120 in order to access information and display a second user interface in response to a service event, where the second user interface may replace the first user interface (step 258). In this case the programmatic information may include a URL or URI which the browser application 122 may use to make the second command request. This allows the server application 120 to notify the browser application 122 of an event that requires a new interface to be displayed, for example if whilst in the middle of the user using a first user interface displayed by the browser application 122 the telephony device 100 receives an incoming telephone call, a service event may be triggered in order to instruct the browser application 122 to load a second user interface from the server application 120 which consists of information relating to the incoming call and hyperlinks and/or buttons associated with command requests that may allow the user to accept or reject the incoming call.

If the first event response indicates that the browser application 122 should pass a second command request to the server application 120 in order to display a second user interface in step 258, the browser will return to step 250 so that it may do so, otherwise it will continue to step 260.

In alternative embodiments of the invention the browser application 122 may execute one or more service event scripts contained in the programmatic information in order to detect service events occurring in relation to the service applications 124 of the telephony device 100. Each service event script may detect service events according to the steps described above. Each event command request passed to the server application 120 by a service event script may indicate that the service event script wishes to receive information relating to the service events of a specific service application 124, for example a service event script may indicate that it wishes to receive information relating to the

service events of the messaging services 128 so that it may display a notification if a new SMS message is received.

The browser application 122 may then process updates to the first user interface made by the one or more browser scripts (step 260). As described
5 above, the one or more browser scripts may be configured to pass one or more command requests from the command information to the server application 120 in response to events detected by the browser scripts, such as user input or other programmatic conditions.

Command requests passed to the server application 120 by a first
10 browser script may be used to instruct the server application 120 to access information in relation to the telephony device 100 and in response update the first user interface. A command request used to update the first user interface in this way may include a URL or URI identifying a resource that is a service application 124 and a first update command. The server application 120 will
15 thus instruct the specified service application 124 to execute the first update command according to steps 200 to 208 and 216 as shown in Figure 2a. In accordance with those steps the execution of the first update command will return to the server application 120 first update content, which may include programmatic information represented in a structured data format. The first
20 update content will then be passed to the first browser script being interpreted by the browser application 122 in a first update response. The first browser script may then use the programmatic information contained in the first update response to update the first user interface, for example the programmatic information may include a list of contacts and their telephone numbers which
25 the first browser script may format and display using an HTML table in the first user interface.

In order for browser scripts to pass command requests to the server application 120 so as to update the first user interface in this way, the browser scripts and the server application 120 may example operate according to the
30 Asynchronous JavaScript and XML (AJAX) framework in order to update the display and behaviour of the first user interface in the background, i.e. without

the browser application 122 needing to entirely re-load the first user interface from the server application 120.

Command requests passed to the server application 120 by a browser script may alternatively be used to instruct the server application 120 to access
5 information in relation to the telephony device 100 and in response provide a second user interface for display by the browser application 122, where the second user interface may replace the first user interface. This may be used for example, to provide a second user interface if certain programmatic conditions tested by the browser script are met.

10 If in step 260 the browser application 122 is instructed by a browser script to replace the first user interface currently displayed as a result of steps 250 to 254 with a second user interface that includes content to be retrieved from the server application 120 using a second command request, the browser application will return to step 250 in order to pass the second command request
15 to the server application 120 and prepare to display this second user interface (step 262), otherwise it will continue to step 264.

The browser may then process any other user input received via the one or more means for user input, where the user input has not been processed by the one or more browser scripts in step 260, such as for example the user selecting an
20 item in the user interface associated with a command request (i.e. where the command request forms part of the set of command requests of the command information) (step 264). Examples of items the user may select that may be associated with command requests include hyperlinks, buttons, forms, etc. If the user has selected an item associated with a command request, the command
25 request may be used by the browser application 122 to instruct the server application 120 to access information in relation to the telephony device 100 and in response provide a second user interface for display by the browser application 122, where the second user interface may replace the first user interface. This may be used to provide a second user interface allowing the user
30 to enter an SMS message to a particular contact that was selected from a list of contacts presented in a first user interface.

If in step 264 the browser application 122 processes other user input selecting an item in the user interface that indicates that the browser application 122 should pass a second command request to the server application 120 in order to display a second user interface, the browser application will return to
5 step 250 in order to pass the second command request to the server application 120 and prepare to display this second user interface (step 266), otherwise it will return to step 256 in order to continue processing service events etc.

The display of user interfaces on the telephony device 100 using the server application 120 and browser application 122 as described above has
10 several advantages described below. Firstly, it is possible to control the telephony device 100 by making command requests via the first transfer protocol which may be a well known transfer protocol such as HTTP. This allows requests for the execution of commands in the first plurality of commands to be made using a well known transfer protocol available for use in
15 existing browsers and by scripts that may run in those browsers.

The granularity of control over the functions of the telephony device 100 offered by the first plurality of commands could be very fine, allowing detailed control of the telephony device 100 using command requests passed to the server application 120 via a first transfer protocol (such as HTTP). The first
20 plurality of commands may thus provide an Application Programming Interface (API) for controlling the functions of the telephony device 100 via HTTP. The provision of information and/or results available as a result of the execution of the first plurality of commands may be returned to the entity passing the command requests in a structured data format via responses using first transfer
25 protocol (such as HTTP responses).

The present invention may allow the creation of complex user interfaces that may incorporate scripts interpreted by the browser application 122 that may be used to enhance user interaction with the user interfaces. These scripts may be used to update portions of a user interface in response to user input with new
30 information or with required information, and/or may be used to update portions

of these user interfaces in response to service events occurring in relation to the one or more service applications 124.

Another advantage of the present invention is that it allows the separation of the programmatic commands necessary to control the functions of the telephony device 100 (i.e. provided by the first set of one or more commands of the service applications 124) from those necessary to generate the user interface (i.e. the scripts executed by the server application 120) and those necessary to display the user interface (i.e. by the browser application 122). This allows the design of each of these components to be undertaken separately and for new user interfaces for the telephony device 100 to be designed easily using well known tools, transfer protocols, etc.

The user interface can be highly customised in accordance with information available in relation to the device that is accessed in response to execution of one or more of the first set of commands, for example the appearance of the user interface displayed in the browser application may be changed (using the scripts interpreted by the server application 120 and/or those interpreted by the browser application 122) according to the time of day, or the location of the telephony device 100 (which may be determined by the location services 132 using the location receiver device (not shown) within the telephony device 100), etc.

It is not necessary for the browser application 122 to use any command requests to the server application 120 that specify resources that are scripts in order to control the functionality of the telephony device 100 or display a user interface. Therefore, in alternative embodiments of the invention the browser application 122 need not be configured to display a user interface that includes content formatted using HTML provided by the server application 120, but may instead display a user interface whose appearance is determined using one or more scripts and/or procedures of the browser application 122. For example the browser application 122 need not be an internet browser configured to display content formatted using HTML, but may use command requests that specify resources that are service applications 124 and commands only, and that are

passed to the server application 120 to execute sets of commands and access information in relation to the telephony device 100 as content in a structured data format such as XML or JSON. The browser application 122 may process and insert this information into a custom user interface that it is configured to be displayed without retrieving any appearance information from the server application 120, i.e. all decisions made in relation to the appearance of the user interface may be made by the browser application 122.

As described in step 200 above, the server application 120 may also be configured to recognise command requests that it has received from an external computing device 150. The external computing device 150 may be configured to run a second browser application that operates in a similar way to the browser application 122 of the telephony device 100, i.e. according to the steps of Figure 2b, in order to display a user interface on a display device of which the external computing device 150 may include or to which it may be connected. Where in Figure 2b the browser application 122 of the telephony device 100 communicates with the server application 120 e.g. by passing command requests to the server application 120 or by receiving from the server application 120 a response to a command request, the second browser application of the external computing device 150 must instead communicate with the server application 120 by using the communications network 140, e.g. by sending HTTP requests over TCP. The telephony device 100 may be accessible to the external computing device 150 via the communications network 140 by it using, for example, a fixed Internet Protocol (IP) address on the communications network 140 known to the external computing device 150, or alternatively by the telephony device 100 registering a current IP address by which it may be accessed with a known server accessible to the external computing device 150 via for example the internet. Thus the functionality of the telephony device 100 may be completely controlled using an external computing device 150 that may display substantially the same user interface as that displayed by the browser application 122, i.e. as both user interfaces are provided in content generated by the server application 120.

In alternative embodiments of the invention a user interface program running on the external computing device may display a user interface whose appearance is determined using one or more scripts and/or procedures of the user interface program. For example the user interface program need not be an internet browser configured to display content formatted using HTML, but may use command requests that specify resources that are service applications 124 and commands only, and that are transmitted to the server application 120 via the communications network 140 to execute one or more commands and access information in relation to the telephony device 100 as content in a structured data format such as XML or JSON. The user interface program may process and insert this information into a custom user interface that it is configured to display on a display device of which the external computing device 150 may include or to which it may be connected, without the user interface program retrieving any appearance information from the server application 120, i.e. all decisions made in relation to the appearance of the user interface are made by the user interface program. Thus a user may control his telephony device 100 from an external computing device 100 using a custom user interface presented by a user interface program running on the external computing program.

In alternative embodiments of the invention the external computing device 150 may include a web control service that may be used to control the functionality of the telephony device 100 and generate content for use in displaying a user interface using a personal computer.

The web control service may use command requests transmitted to the server application 120 via the communications network 140 to execute one or more of the first set of one or more service commands and access information in relation to the telephony device 100 as content in a structured data format such as XML or JSON. The web control service may process this information and insert it into web content which it may provide to a personal computer so that the web content may be displayed on a display device to which the personal computer is connected, without the web control service retrieving any appearance information from the server application 120. Thus all decisions

made in relation to the appearance of the user interface may be made by the web control service. Thus a user may control his telephony device 100 from a personal computer by using a custom user interface presented on the personal computer, where the custom user interface includes content generated by the web control service running on the external computing device 150. In this way a custom user interface by which the telephony device 100 may be controlled may be incorporated into, for example, an internet webpage being viewed by a user of the telephony device 100 on a personal computer. For example, a custom user interface for controlling a user's telephony device 100 could be included in internet webpages that are presented to the user when viewing his Facebook™ page on a personal computer, where the external computing device 150 (which may be one or more computer servers that also provide the content of the Facebook™ page) provides the custom user interface. The custom user interface could for example allow the user to make a mobile phone call using the telephony device 100, or to send an SMS message using the telephony device 100 to another user.

In alternative embodiments of the invention, the server application 120 may in step 216 of Figure 2a pass the first response to the browser application 122 in response to a first command request received in step 200 from, for example, an external computing device 150. The first response may include the first content generated by the server application in either step 208 or step 214. A second response that includes the first content may also be provided to the computing device 150. This allows the browser application 122 of the telephony device 100 to display a first user interface on the display device 116 of the telephony device 100 in response to a user controlling the functionality of the telephony device 100 using the external computing device 150 according one of the alternative embodiments of the invention described above. The first user interface displayed by the browser information may be indicative of the first information accessed by the server application 120 in response to the first command request in either step 206 or 212.

In alternative embodiments of the invention, the server application 120 may in step 216 of Figure 2a pass a first event response to a service event script that is being interpreted by the browser application 122 in response to a first command request received in step 200 from, for example, an external computing device 150. The first event response may include the first content generated by the server application in either step 208 or step 214. Alternatively the first event response may include an indication to a service event script that the user interface displayed by the browser application 122 should be updated, for example by providing a command request to the service event script that the service event script may instruct the browser application 122 to use in order to update the user interface displayed by the browser application 122.

Figure 3 illustrates steps that may be carried out by the browser application 122, the event services 138 and the messaging services 128 in order to transmit a SMS message in response to input from a user into a first user interface, and in order to display an indication confirming delivery of the text message, according to an embodiment of the invention.

Initially the browser application 122 may receive a first response that includes first content from the server application 120 and display a first user interface using the first content in accordance with the steps in Figure 2b described above (step 300).

The browser application 122 may also execute a service event script contained in the programmatic information in order to detect service events occurring in relation to the service applications 124 of the telephony device 100. The service event script may be configured to detect service events by initially passing a first event command request to the server application 120 (step 302). The first event command request passed to the server application 120 by the event script may indicate that the service event script wishes to receive information relating to the service events of the messaging services 128 so that it may display a notification if a new SMS message is received, or if a transmitted SMS message is successfully delivered.

The server application 120 may then instruct the events services 138 to execute the first event command request using the event services 138 according to the steps of Figure 2a. No new service events of the messaging services 128 have occurred, so the receive new events command of the events service 138 may not return any content to the server application 120 until a new service event has occurred, such that the server application 120 may not provide the first new events response until a new service event has occurred. The first event command request thus remains pending whilst the events services 138 waits for a new event to occur. The first event command request may be configured to lapse i.e. timeout after a certain period of time, in which case a new first event command request is passed by the service events script in order to replace the first event command request that has lapsed.

The user may in the meantime use the first user interface to enter an SMS message. The user may select to send the SMS message using the first user interface by pressing a link or a button in the first user interface. This link or button may either cause the browser application 122 to pass a first command request to the server application 120, or may cause a browser script running in the browser application to pass a first command request to the server application 120 (step 304). This first command request may instruct the server application 120 to execute a first set of one or more commands of the first plurality of commands in order to send the entered SMS message. The first command request may include data relating to the SMS message to be sent, for example the first command request may comprise a URL with a parameter containing the text of the SMS message to be sent, another parameter with the telephone number to send the SMS message to, etc.

The server application 120 may then process the first command request according to the steps of Figure 2a. The first command request may indicate that the server application 120 should instruct the messaging services 128 to execute one or more commands necessary to send the SMS message, so the server application 120 may request that the messaging services 128 execute the set of one or more commands of the first plurality of commands in order to send

the entered SMS message. As a result of executing the set of one or more commands the messaging services may use the network interface 112 to transmit via the communications network 140 (which may be a GSM network) an SMS message comprising the text specified by the user to the specified telephone number (step 306).

The messaging services 128 may then access information in relation to the telephony device 100, such as information relating to the successful (or otherwise) transmission of the SMS message by the network interface 112, generate first content using this information which is returned to the server application 120. The server application 120 may then provide a first response containing the first content to the browser application (step 308).

The browser application 122 may then receive the first response and use the first content to update the first user interface in order to show that the SMS message had been transmitted (step 310). For example the first content may include programmatic information such as data in a structured data format that indicates that the SMS message has been transmitted, which may be used by a browser script running in the browser application 122 to update the first user interface.

In the meantime the SMS message may be delivered by the communications network 140 (which may be for example a GSM network) to the telephone with the telephone number specified by the user. This may result in a confirmation of SMS message delivery being transmitted to the telephony device 100 via the communications network 140, which may be received by the network interface 112 and processed by the messaging services 128. The messaging services 128 may then notify the event services 138 of the confirmation of the delivery of the SMS message (step 312), for example by using a method for inter-process communication.

The event services 138 may then identify that it is waiting for service events in relation to the messaging services 128. The event services 138 may thus generate second content relating to the confirmation of the delivery of the SMS message which is returned to the server application 120 so that it may

provide a second response that includes the second content to the service events script running in the browser application 122 (step 314). The second response may thus be a response to the first event command request.

5 The service events script running in the browser application 122 may then receive the second response and use the second content to update the first user interface in order to show that the SMS message had been delivered (step 316). For example the second content may include programmatic information such as data in a structured data format that indicates that the SMS message has been delivered, which may be used by the service events script to update the
10 second user interface by displaying a delivery notification message.

Figure 4 shows an exemplary screen 400 relating to a list of contacts resulting from the browser application 122 processing content in order to display a user interface indicative of information accessed in relation to the telephony device 100 by the server application 120. The screen 400 includes contact
15 information e.g. 400 that may have been retrieved from the non-volatile storage 106 of the telephony device 100, i.e. it is indicative of information accessed in relation to the telephony device 100. The screen 400 includes a number of hyperlinks e.g. 404, 406, 410, 412 that may each be associated with a command request, e.g. hyperlink 406 is associated with the command request 408 which is
20 the URL “http://phoneServer/telephony/call?number=1234”. If one of these hyperlinks is selected by the user, the browser application 122 may pass the command request to the server application 120 in order to update the user interface or display a new user interface.

Figure 5 shows an exemplary screen 500 relating to an incoming
25 telephone call resulting from the browser application 122 processing content in order to display a user interface indicative of information accessed in relation to the telephony device 100 by the server application 120. The screen may have been displayed by the browser application 122 as a result of a service event script (i.e. being interpreted by the browser application 122) receiving an event
30 response from server application 120. The event response may have indicated that the browser application 122 should pass a command request to the server

application 120 in order to access information and display the displayed user interface in response to a service event, or the event response may have included programmatic information that the service event script may have processed in order to update the displayed user interface with the displayed information relating to the incoming telephone call. The screen 500 includes a number of hyperlinks e.g. 502, 504 that may each be associated with a command request. If one of these hyperlinks is selected by the user the browser application 122 may pass the command request to the server application 120 in order to answer or cancel the incoming call (as appropriate) and update the user interface or display a new user interface.

Figure 6 shows part of an exemplary screen 600 that includes a status bar 602 resulting from the browser application 122 processing content in order to display a user interface indicative of information accessed in relation to the telephony device 100 by the server application 120. The screen may include a plurality of HTML frames that were passed by the server application 120 to the browser application 122 as first content within a first response to a first command request made by the browser application 122. One of the HTML frames in the plurality of HTML frames may be a status bar frame 602, which displays information indicative of information available in relation to the telephony device 100. The contents of the status bar frame may be updated by a service events script running in the browser application 122, in response to the service events script receiving new event content from the events service 138 in response to service events occurring in relation to the service applications 124. For example, such new event content might indicate that the operating system 118 has detected that the remaining battery power of the telephony device 100 has dropped, and may include content that may be used to update the battery level indicator 604. Alternatively such new event content might indicate that the connectivity services 136 have detected that the level of connectivity of the telephony device 100 to a communications network has changed, and may include content that may be used to update the connection level indicator 606. Alternatively such new event content might indicate a new SMS message, MMS

message or email has been received by the messaging services 128, and may include content that may be used to update the new message indicator 608. Alternatively such new event content might indicate that the operating system 118 has detected that the time has changed, and may include content that may be used to update the time indicator 610.

The above embodiments are to be understood as illustrative examples of the invention. Further embodiments of the invention are envisaged as follows.

Embodiments of the invention are envisaged wherein the server application 120 and/or browser application 120 may not provide all of the functionality described above and/or may be further modified so that they may be used with a telephony device 100 that includes a processor 102 of comparatively lower performance. For example, some or all of the browser scripts may not be interpreted by the browser application 122, and/or scripts interpreted by the server application 120 for executing one or more commands and generating content may be simplified in order to provide a more basic user interface and reduce the processing burden on the processor 102.

In alternative embodiments of the invention the browser application 122 of the telephony device 100 may also be configured to request and receive second content from external devices by using the network interface 112 of the telephony device 100 to conduct communications via a communications network 140 with external devices such as web servers using the same first transfer protocol used in the steps of Figures 2a and 2b, i.e. for example HTTP. The browser application 122 may therefore be used to provide access to internet content provided by these external devices as well as to provide a user interface for use in controlling the functionality of the telephony device 100.

It is to be understood that the first transfer protocol used to pass command requests to the server application 120 (from either the browser application 122 or an external computing device 150) and used by the server application 120 to provide responses to the command requests may be either a secured transfer protocol, such as HTTP Secure (HTTPS), or an unsecured transfer protocol, such as HTTP.

It is to be understood that the telephony device 100 described above is an exemplary embodiment of the invention and that it is envisaged that the invention can be applied to other devices that include a display and a processor, wherein such a device may be configured to run a server application and a browser application using the processor, and wherein the browser application may be configured to display a user interface on the display. The server application of such a device may proceed according to the steps shown for the server application 120 in Figure 2a and according to other steps and embodiments incorporating the server application 120 described above. The browser application of such a device may proceed according to the steps shown for the browser application 122 in Figure 2b and according to other steps and embodiments incorporating the browser application 122 described above. Such a device may also include substantially the same components as the telephony device 100 and those components may operate in substantially the same way as their equivalents in the telephony device 100 as described. The device may however include a different set of one or more service applications (i.e. to the service applications 124 of the telephony device 100) that a different first plurality of commands that allow control of the functions of that device.

For example, such a device could be a television device or a set-top box for connecting to a television device that includes a video broadcast receiver. Such a device may include service applications such as video source selection services for changing a received video source of the video broadcast receiver (and thus changing the state of the video broadcast receiver), for example allowing the selection of different television channels and/or video input sources. Such a device may also include video playback services for decoding and playback of different types of video, and guide services for accessing and displaying a television listings guide using the network interface 112.

Other examples of the device include a portable media player such as a portable music, picture and/or video player, or a portable electronic book reader, each of which may include service applications such as media services for storing, retrieving, editing, recording and/or playback of multimedia files and/or

ebook services for storing, retrieving, editing, recording and/or playback of electronic book files. In the above examples, the device may include configuration services, connectivity services and events services that provide substantially the same functionality as those of the telephony device 100.

5 Other examples of the device include a digital camera which may include service applications such as media services for storing, retrieving, editing and/or playback of multimedia files and/or camera services for controlling an image sensor, lens arrangement and/or other elements of the camera in order to take digital images. In the above examples, the device may
10 include configuration services, connectivity services and events services that provide substantially the same functionality as those of the telephony device 100.

 It is to be understood that any feature described in relation to any one embodiment may be used alone, or in combination with other features described,
15 and may also be used in combination with one or more features of any other of the embodiments, or any combination of any other of the embodiments. Furthermore, equivalents and modifications not described above may also be employed without departing from the scope of the invention, which is defined in the accompanying claims.

20

Claims

1. A device comprising a display and a processor, wherein the device is configured to run a browser application and one or more service applications using the processor, and wherein the browser application is
5 configured to display a user interface on the display,

wherein the browser application is configured to receive a first response to a first command request, and in response to receiving the first response, to display a first user interface on the display which includes first content;

10 wherein the device further comprises a server application which is configured to recognise said first command request, said first command request being provided to the server application using a first transfer protocol, and in response to the first command request, to provide instructions to a service application to execute a first set of one or more commands in order to control a
15 function of the device,

wherein said service application is configured to, in response to receipt of said instructions from the server application, control a function of the device and generate first content for receipt by the server application,

20 wherein the server application is configured to, in response to receipt of the first content from said service application, generate said first response to the first command request, the first response comprising the first content.

2. A device according to claim 1, wherein the first transfer protocol comprises HyperText Transfer Protocol (HTTP) or HTTP Secure (HTTPS).

25

3. A device according to claim 1 or 2, wherein the device comprises a plurality of components connected to a system bus, wherein the device further comprises an operating system and wherein in order to control a function of the device said service application is configured to, in response to receipt of said
30 instructions from the server application, control said function of the device

through use of the operating system to change the state of at least one of the first plurality of components via the system bus.

4. A device according to any of claims 1 to 3, wherein said service
5 application provides telephony services, and wherein said function includes
creating or receiving a telephone call via the telephony services using a network
interface.

5. A device according to any preceding claim, wherein said service
10 application provides messaging services, and wherein said function includes
sending or receiving a text message via the messaging services using a network
interface.

6. A device according to any preceding claim, wherein said service
15 application provides connectivity services, and wherein said function includes
connecting to or disconnecting to a communications network via the
connectivity services using a network interface.

7. A device according to any preceding claim, wherein said service
20 application provides social networking services, and wherein said function
includes sending or receiving data to a social network via the social networks
services using a network interface.

8. A device according to any preceding claim, wherein said service
25 application provides video source selection services, and wherein said function
includes changing a received video source via the video source selection
services using a video broadcast receiver.

9. A device according to any preceding claim, wherein said service
30 application provides event services and wherein said service application is

configured to delay providing said first content until a service event occurs in relation to a service application of the one or more service applications.

10. A device according to claim 9, wherein said service event is
5 receiving an incoming telephone call.

11. A device according to claim 9, wherein said service event is
detecting a new device location.

10 12. A device according to any preceding claim, wherein the first
command request identifies a first resource which identifies said service
application and a first command, and wherein the server application instructs the
service application to execute the first command in order to execute the first set
of one or more commands.

15 13. A device according to any of claims 1 to 11, wherein the first
command request identifies a first resource which includes a first script, and
wherein the server application instructs an interpreter program to interpret the
first script in order to execute the first set of one or more commands.

20 14. A device according to any preceding claim, wherein the device
comprises a network interface for conducting communications via a
communications network, and wherein the browser application is configured to
request and receive second content using the first transfer protocol via the
25 communications network.

15. A device according to any preceding claim, wherein the device
comprises one or more user input means and wherein the device is configured to
pass the first command request from the browser application to the server
30 application in response to user input received from the one or more user input
means.

16. A device according to any of claims 1 to 14, wherein the device comprises a network interface for conducting communications via a communications network, and wherein the server application is configured to receive the first command request from an external computing device via the communications network.

17. A system comprising a first device and a second device, the first device comprising a network interface for conducting communications via a communications network and being connected to the second device via the communications network, wherein the second device comprises a display, one or more user input means and a processor, and the first device comprises a processor configured to run one or more service applications,

wherein said second device is configured to, in response to receiving first user input from the one or more user input means, transmit a first command request to the first device using a first transfer protocol via the communications network,

wherein the first device further comprises a server application which is configured to recognise said first command request and in response to the first command request, to provide instructions to a service application to execute a first set of one or more commands in order to control at least one of a telephony function, a messaging function, a connectivity function and an event function of the first device,

wherein said service application is configured to, in response to receipt of said instructions from the server application, control at least one of a telephony function, a messaging function, a connectivity function and an event function of the first device and generate first content for receipt by the server application,

wherein the server application is configured to, in response to receipt of the first content from said service application, generate a first response to the

first command request, the first response comprising the first content, and transmit the first response to the second device,

wherein said second device is configured to, in response to receiving the first response, display a first user interface on the display of the second device
5 which includes the first content.

18. A method for transmitting a command request to a first device from a second device and receiving a response comprising content, wherein the first device comprises a network interface for conducting communications via a
10 communications network and is connected to the second device via the communications network, wherein the second device comprises a display, one or more user input means and a processor, and the first device comprises a processor configured to run a server application and one or more service applications, comprising:

15 receiving at the second device first user input from the one or more user input means, and in response transmitting a first command request to the first device using a first transfer protocol via the communications network;

recognising at the server application the first command request, and in response the server application providing instructions to a service application to
20 execute a first set of one or more commands in order to control at least one of a telephony function, a messaging function, a connectivity function and an event function of the device;

in response to receipt of said instructions from the server application said service application controlling at least one of a telephony function, a messaging
25 function, a connectivity function and an event function of the first device and generating first content for receipt by the server application;

in response to receipt of the first content from said service application the server application generating a first response to the first command request, the first response comprising the first content, and transmitting the first response
30 to the second device; and

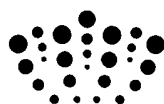
in response to receiving the first response comprising first content the second device displaying a first user interface on the display of the second device which includes the first content.

5 19. A telephony device comprising a display and a processor, wherein the telephony device is configured to run a server application and one or more service applications using the processor, and comprises a network interface for conducting communications via a communications network,

 wherein the server application is configured to recognise a first
10 command request provided to the server application using a first transfer protocol via the communications network, and in response to recognising the first command request, to provide instructions to a service application to execute a first set of one or more commands in order to control a function of the telephony device,

15 wherein said service application is configured to, in response to receipt of said instructions from the server application, control a function of the telephony device and generate first content for receipt by the server application,

 wherein the server application is configured to, in response to receipt of the first content from said service application, generate a first response to the
20 first command request, the first response comprising the first content and transmit the first response to the first command request using the first transfer protocol via the communications network.



Application No: GB1004456.8

Examiner: Kalim Yasseen

Claims searched: 1-19

Date of search: 15 July 2010

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
X	at least 1, 17, 18, 19	US7502993 B1 (CISCO TECHNOLOGIES) see whole document especially col. 2 lines 19-50, col. 7 line 21- col.8 line 42
X	at least 1	US6628644 B1 (CISCO) see whole document especially col. 6 lines 25-44
X	at least 1,	US2007/0101004 A1 (LOEN) see whole document especially abstract, claims 1-3

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC^X :

--

Worldwide search of patent documents classified in the following areas of the IPC

G06F; H04L; H04W

The following online and other databases have been used in the preparation of this search report

Online: EPODOC, TEXTEN, INSPEC, INTERNET, NPL, WPI, XPESP

International Classification:

Subclass	Subgroup	Valid From
H04L	0029/08	01/01/2006
G06F	0017/30	01/01/2006
H04M	0001/247	01/01/2006
H04W	0008/24	01/01/2009