

(43) International Publication Date
7 September 2012 (07.09.2012)

(51) International Patent Classification:

G06F 17/30 (2006.01) *G06Q 30/00* (2012.01)
G06F 15/16 (2006.01)

(21) International Application Number:

PCT/AU2012/000197

(22) International Filing Date:

28 February 2012 (28.02.2012)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

2011900744 2 March 2011 (02.03.2011) AU

(71) Applicant (for all designated States except US): **THE UNIVERSITY OF SYDNEY** [AU/AU]; Parramatta Road, Sydney, New South Wales 2006 (AU).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **VON TAKACH DUKAI, Stephen Alexander** [AU/AU]; 23 Daly Avenue, Wahroonga, NSW 2076 (AU).(74) Agent: **SPRUSON & FERGUSON**; GPO BOX 3898, Sydney, New South Wales 2001 (AU).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

(54) Title: A DIGITAL SIGNAGE SYSTEM

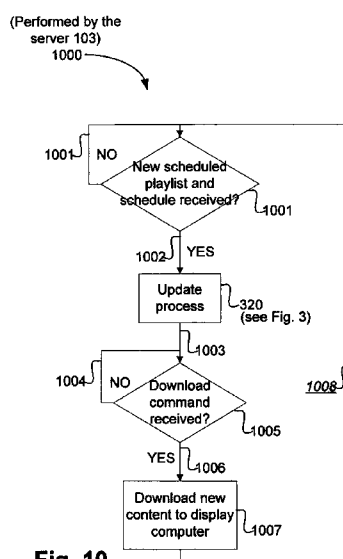


Fig. 10

(57) Abstract: A system (100) for displaying content, the system comprising: one or more user computers (105) for creating the content and uploading the content to a server (103); the server for communicating (316) an update command to a plurality of display computers (110) over a network (109); and the plurality of display computers (110), having respective display devices and running respective web browser software programs having associated cache memories; wherein the web browsers are configured to receive the update command and in response thereto to communicate with the server, download (1007) the content into the respective cache memories, and display the content on the respective display devices.

- 1 -

A DIGITAL SIGNAGE SYSTEM

Technical Field of the Invention

The present invention relates generally to display of visual and/or audio-visual content over wide geographic areas.

Background

5 “Digital Signage” refers to the delivery of media content, typically visual and audio-visual, to targeted display devices (clients) which may be distributed over a wide geographic area. Such content may be audio, images, video, or audio-visual, and all of these are referred to simply as “content” unless explicitly stated otherwise. Current arrangements require users to undergo extensive training in the use of complex software,
10 which involves manual management of users, and typically requires custom software and associated licenses on each client.

Summary

It is an object of the present invention to substantially overcome, or at least ameliorate, one or more disadvantages of existing arrangements.

15 Disclosed are arrangements, referred to as Remote Server Activated Browser Operation (RSABO) arrangements, which seek to address the above problems by concentrating specialized functionality in one or more RSABO servers which communicate over a communication network with user computers and display computers. The RSABO arrangements use standard web browsers 233, 115 in the user computers and the display
20 computers respectively. A “browser” in this specification is any piece of software that can display a web page. The RSABO server performs the scheduling and other RSABO functions using a specialised software application 114.

- 2 -

The browser running on the display computer is presented with lists to present, the lists being provided together with various conditions and timeout parameters.

If for any reason that list is changed, the server 103 publishes a new list for download by the relevant set of display computers. The display computers download the published new list from the server 103 after receiving an update command that is sent to the display computers from the server.

In one arrangement, the update command from the server is sent to a display computer such as 110 after the browser 115 in the display computer 110 polls the server 103, and once polled, the server 103 determines that a download is needed and thus sends the update command to the display computer 110. Alternately, the update command from the server is sent to the display computer when the server 103 determines, without being polled, that the update command should be sent.

The display computer 110 receives the update command at the same time that the browser 115 in the display computer 110 is presenting a current playlist. Upon receiving the update command the browser 115 in the display computer 110 downloads the update as a background process before switching to the newly downloaded playlist for display.

If a communication problem between the display computer 110 and the server 103 occurs during the download, thus corrupting the newly downloaded playlist, the display computer 110 detects this occurrence and does not use the corrupted newly downloaded data, but rather continues to use previously loaded playlists.

If the newly downloaded playlist(s) are uncorrupted then the browser will update its internal state to use these lists.

The browser 115 in the display computer 110 only plays what it is directed to by the server 103, and effectively makes no autonomous decisions in this regard.

- 3 -

According to a first aspect of the present invention, there is provided a system for displaying content comprising one or more playlists, the system comprising one or more servers, a plurality of user computers, and a plurality of display computers communicating over a telecommunications network, wherein:

5 the user computers and the display computers run respective web browser software applications having associated cache memories;

 the user computers are configured for creating or modifying the content and downloading the content to the servers;

 the servers are configured for storing the content in one or more databases and
10 sending an update command to those display computers having corresponding display schedules which are affected by the content; and

 the display computers are configured for uploading the content into their respective cache memories in response to receiving the respective update commands, and displaying the content.

15 According to another aspect of the present invention, there is provided an apparatus for implementing any one of the aforementioned methods.

 According to another aspect of the present invention, there is provided a computer readable non-transitory tangible medium having recorded thereon a computer program for implementing any one of the methods described above.

20 Other aspects of the invention are also disclosed.

Brief Description of the Drawings

At least one embodiment of the present invention will now be described with reference to the drawings and appendices, in which:

- 4 -

Fig. 1 shows a functional block diagram of a system upon which the disclosed RSABO arrangements can be implemented;

Figs. 2A and 2B show a representation of the system of **Fig. 1** in which a user computer such as 233 is implemented using a general purpose computer;

5 **Fig. 3** shows two process fragments, one fragment 300 that is performed by a user computer 105, and one fragment 320 that is performed by the server in **Fig. 1**;

Figs. 4A – 4C depict how the display computers operate in the RSABO arrangements;

Fig. 5 shows a GUI for inputting new content;

10 **Fig. 6** shows a GUI in which content is dragged and dropped into a playlist;

Fig 7 shows a GUI for previewing an item of content;

Fig. 8 shows a GUI for scheduling a playlist;

Fig. 9 shows a scheduled playlist;

Fig. 10 shows a process performed by the server;

15 **Fig. 11** shows a process fragment performed by a user computer and the server to create and store new content for possible subsequent display by the RSABO system; and

Fig. 12 shows two sets of process fragments, one for creating and/or modifying a playlist for uploading to the server, and one for reviewing and arranging, by a user computer, for publication of the reviewed playlist, and for publishing the playlist and
20 sending update commands as appropriate.

Detailed Description including Best Mode

Where reference is made in any one or more of the accompanying drawings to steps and/or features, which have the same reference numerals, those steps and/or features

- 5 -

have for the purposes of this description the same function(s) or operation(s), unless the contrary intention appears.

It is to be noted that the discussions contained in the "Background" section and that above, relating to prior art arrangements, relate to discussions of devices which may form public knowledge through their use. Such discussions should not be interpreted as a representation by the inventor or the patent applicant that such devices in any way form part of the common general knowledge in the art.

The disclosed RSABO arrangements display, on a plurality of diverse display computers having display screens (also referred to as display devices), user specified audio and/ or audio-visual content (also referred to as media for display, or simply as presentation(s)) including text, images, graphics and video as desired. The display computers can be geographically distributed, and communication between a RSABO server and the client machines (ie the display computers) takes place via a data communications network such as the "Internet".

The client machines (ie the display computers) run web browsers with associated cache memories, and in one RSABO arrangement the aforementioned server or a number of such servers enable the system to operate in a client-server configuration. This is described in more detail hereinafter with reference to Fig. 1. Client machines typically run standard web browser computer software applications, with no additional software being required. This is described in more detail hereinafter with reference to Figs. 2A and 2B.

The servers run specialized software associated with communicating "update" commands to the population of client machines in order to trigger those client machines to download updated information for their displays. This is described in more detail hereinafter with reference to Fig. 3. Users of the RSABO arrangements, using user

- 6 -

computers, construct display presentations from media items selected by the users, construct schedules for presentation of the content, and upload this material to the server(s).

Processing at the client end (ie at the display computers such as 110) is preferably
5 kept to a minimum.

When a client machine 110 first powers up, the machine automatically logs on to the RSABO server and downloads a pre-compiled presentation potentially comprising multiple default, or specifically defined, presentations for that particular display. Each content item in the presentation is associated with a control parameter representing a
10 specific time duration or call-back time period which needs to be monitored by the display machine. Video presentations, which are generally supported natively by web browsers, will play to the end of the video presentation by default if a duration parameter is not provided by the user. Alternately, call-back signalling can be used to control the flow of the video presentation if, for example, there is an error in the presentation or the video has
15 ended.

Having downloaded the above-noted information, (as a pre-requisite the client machine firstly downloads or loads from its offline cache whatever software is necessary for the browser to support "player" functionality), the client machine displays the relevant presentation until an update is received and then the old presentation is replaced. This
20 procedure continues whether or not the client machine is connected to the server. This provides a significant benefit since continuous connectivity to the server is not required. This is described in more detail hereinafter with reference to Fig. 4.

The disclosed RSABO arrangements are largely platform independent, and largely independent of operating systems specifics. These advantages accrue at least to some

- 7 -

extent because the client machines use standard web-browsers to perform the RSABO functions.

The RSABO arrangements can be used in a campus environment. This type of application can use a large number of screens, many of which are located outside particular campus buildings. Each of these screens display information associated with a relevant building, thus presenting information that is of interest to people using the particular building. The RSABO arrangements can also be used for more generalised advertising and other purposes in public areas.

The universe of users who are authorised to use the RSABO system create and display content on one or more desired display devices (referred to as authorised display devices) on display computers for which the user group has display permission. The users are partitioned into user groups, and within any particular user group, all users can create and display content on authorised display devices.

Any user can add a display machine to the system merely by being an authorised member of a user group. The RSABO arrangements can treat multiple individual display panes presented on a single physical display such as 214 in Fig. 2A as "virtual displays", each being individually addressable by associated URLs or similar addressing methods. These virtual displays can be used when required at the users' discretion, for applications such as temporary events or generally allowing a display computer to present multiple independent panes on the one physical display.

The RSABO arrangements enable data "hiding" and provision of privileges for different user groups and users. In order for a display computer to become accessible outside of his or her user group, intervention of an administrator level user is required. Administrators can share displays among groups of users, allowing disparate groups to

- 8 -

provide content independently of each other, while preventing one group of users from modifying a playlist being constructed by another group. A playlist is a list of presentations each of which may include one or more audio, image, video, or audio-visual segments. Playlists created by the disparate groups can be merged, and then played one after another
5 in a final presentation. A particular user group can 'take control' of a display exclusively for a period of time and potential disputes can be resolved through communication between the groups.

The RSABO arrangements enable authentication and similar operations to piggy back off those operations that are already operative in the user group systems upon which
10 the RSABO arrangements are implemented. Thus, for example, if a particular group of users runs Lightweight Directory Access Protocol (LDAP) and wishes to register for user status on an RSABO system, then its own LDAP arrangements will be used as the basis for enabling its own users to access the RSABO system. Another group of users using OpenID (an open decentralized standard for authenticating users which can be used for access
15 control) can also be authenticated on the same system using their OpenID authenticator. This approach is known as federated identity or federated authentication and allows users to maintain a single sign-on within their organisational group when the authentication system is trusted.

In one RSABO arrangement a user, using a user computer, compiles a playlist
20 from a library containing media content associated with his group, and inserts the playlist into a schedule for the display (s) of interest, using a user-friendly graphical user interface described in more detail hereinafter with reference to Figs. 5-9. Playlists (also referred to as presentations) can be scheduled for display on one or more displays and displays can be

- 9 -

scheduled to display one or more playlists. If a display is shared then a display will play all the playlists scheduled for it by the various user groups.

Fig. 1 shows a functional block diagram of a system upon which the disclosed RSABO arrangements can be implemented. The users of the RSABO system 100 each have associated user computers such as user computer 105. The users create, on their respective user computers, the content they wish to display and the schedule according to which it is to be displayed, and upload the content and the schedule to a server 103 as described hereinafter, from the user's perspective, with reference to a process fragment 300 in **Fig. 3**. As the system 100 grows (ie scales up), additional servers such as 103 can be added to cater for increased load. The server 103 stores this material on a content storage database 101. The server 103 traverses, as described hereinafter with reference to a process fragment 320 in **Fig. 3**, a playlist, and sends update commands to a specified set of display computers such as 110 over a network 109. The set of display computers download new content and display the new content as described hereinafter with reference to **Fig. 4**.

Fig. 1 shows a number of user computers 105, ..., 107 that are connected by respective connections 106, ..., 108 to a network 109 (which is depicted as the Internet). However, the skilled reader will appreciate that the network 109 may comprise other networks including, but not limited to Local Area Networks (LANs), Wide Area Networks (WANs), or Wireless Local Area Networks (such as WiFi™). A number of display computers 110, ..., 112 are connected by respective connections 111, ..., 113 to the network 109. The server (or multiple servers) 103 is connected to the network 109 by a connection 104, and to the content database 101 by a connection 102.

Figs. 2A and 2B depict a general-purpose computer system 200, upon which the various RSABO arrangements described can be practiced. The user computers such as 105,

- 10 -

the display computers such as 110 and the server(s) such as 103 typically, but not necessarily, all have similar structural and functional features to those referred to in **Figs. 2A and 2B**.

The server runs, in addition to other software, RSABO software (114) associated
5 with the functionality described hereinafter with reference to the process fragment 320 in **Fig. 3** and the maintenance of schedules of media to be displayed (not shown). The display computers such as 110 run, in addition to other software, software 115 comprising standard web browsers supporting the functionality described hereinafter with reference to **Fig. 4**. The user computers such as 105 run, in addition to other software, RSABO software 233
10 comprising standard web browsers associated with the functionality described hereinafter with reference to a process fragment 300 in **Fig. 3**. The following description relating to **Figs. 2A and 2B** is directed specifically to the user computers such as the user computer 105. However the description applies equally, with minor variations associated primarily with different software running on the respective machines, to the structural and functional
15 features of the display computers such as the display computer 110 and the server(s) such as the server 103.

As seen in **Fig. 2A**, the computer system 200 includes: a display computer module 105; input devices such as a keyboard 202, a mouse pointer device 203, a scanner 226, a camera 227, and a microphone 280; and output devices including a
20 printer 215, a display device 214 and loudspeakers 217. An external Modulator-Demodulator (Modem) transceiver device 216 may be used by the computer module 105 for communicating to and from the server 103 over the communications network 109 via a connection 221. The communications network 109 may be a wide-area network (WAN), such as the Internet, a cellular telecommunications network, or a private WAN. Where the

- 11 -

connection 221 is a telephone line, the modem 216 may be a traditional "dial-up" modem. Alternatively, where the connection 221 is a high capacity (e.g., cable) connection, the modem 216 may be a broadband modem. A wireless modem may also be used for wireless connection to the communications network 109.

5 The computer module 105 typically includes at least one processor unit 205, and a memory unit 206. For example, the memory unit 206 may have semiconductor random access memory (RAM) and semiconductor read only memory (ROM). The computer module 105 also includes an number of input/output (I/O) interfaces including: an audio-video interface 207 that couples to the video display 214, loudspeakers 217 and
10 microphone 280; an I/O interface 213 that couples to the keyboard 202, mouse 203, scanner 226, camera 227 and optionally a joystick or other human interface device (not illustrated); and an interface 208 for the external modem 216 and printer 215. In some implementations, the modem 216 may be incorporated within the computer module 105, for example within the interface 208. The computer module 105 also has a local network
15 interface 211, which permits coupling of the computer system 200 via a connection 223 to a local-area communications network 222, known as a Local Area Network (LAN). As illustrated in Fig. 2A, the local communications network 222 may also couple to the wide network 109 via a connection 224, which would typically include a so-called "firewall" device or device of similar functionality. The local network interface 211 may comprise
20 an EthernetTM circuit card, a BluetoothTM wireless arrangement or an IEEE 802.11 wireless arrangement; however, numerous other types of interfaces may be practiced for the interface 211.

The I/O interfaces 208 and 213 may afford either or both of serial and parallel connectivity, the former typically being implemented according to the Universal Serial Bus

- 12 -

(USB) standards and having corresponding USB connectors (not illustrated). Storage devices 209 are provided and typically include a hard disk drive (HDD) 210. Other storage devices such as a floppy disk drive and a magnetic tape drive (not illustrated) may also be used. An optical disk drive 212 is typically provided to act as a non-volatile source of data. Portable memory devices, such optical disks (e.g., CD-ROM, DVD, Blu-ray DiscTM), USB-RAM, portable, external hard drives, and floppy disks, for example, may be used as appropriate sources of data to the system 200.

The components 205 to 213 of the computer module 105 typically communicate via an interconnected bus 204 and in a manner that results in a conventional mode of operation of the computer system 200 known to those in the relevant art. For example, the processor 205 is coupled to the system bus 204 using a connection 218. Likewise, the memory 206 and optical disk drive 212 are coupled to the system bus 204 by connections 219. Examples of computers on which the described arrangements can be practised include IBM-PC's and compatibles, Sun Sparcstations, Apple MacTM or a like computer systems.

The RSABO arrangements may be implemented using the computer system 200 wherein the process fragment 300 of Fig. 3, to be described, may be implemented as one or more RSABO software application programs 233 executable within the computer system 200. In particular, the steps of the method of Fig. 3 are effected by instructions 231 (see Fig. 2B) in the software 233 that are carried out within the computer system 200. The software instructions 231 may be formed as one or more code modules, each for performing one or more particular tasks. The software may also be divided into two separate parts, in which a first part and the corresponding code modules performs the RSABO methods and a second part and the corresponding code modules manage a user interface between the first part and the user.

- 13 -

The software may be stored in a computer readable medium, including the storage devices described below, for example. The software is loaded into the computer system 200 from the computer readable medium, and then executed by the computer system 200. A computer readable medium having such software or computer program recorded on the
5 computer readable medium is a computer program product. The use of the computer program product in the computer system 200 preferably effects an advantageous apparatus for effecting the RSABO arrangements.

The RSABO software 233 is typically stored in the HDD 210 or the memory 206. The software is loaded into the computer system 200 from a computer readable medium,
10 and executed by the computer system 200. Thus, for example, the software 233 may be stored on an optically readable disk storage medium (e.g., CD-ROM) 225 that is read by the optical disk drive 212. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer program product in the computer system 200 preferably effects an apparatus for effecting
15 the RSABO arrangements.

In some instances, the RSABO application programs 233 may be supplied to the user encoded on one or more CD-ROMs 225 and read via the corresponding drive 212, or alternatively may be read by the user from the networks 109 or 222. Still further, the software can also be loaded into the computer system 200 from other computer readable
20 media. Computer readable storage media refers to any non-transitory tangible storage medium that provides recorded instructions and/or data to the computer system 200 for execution and/or processing. Examples of such storage media include floppy disks, magnetic tape, CD-ROM, DVD, Blu-ray Disc, a hard disk drive, a ROM or integrated circuit, USB memory, a magneto-optical disk, or a computer readable card such as a

- 14 -

PCMCIA card and the like, whether or not such devices are internal or external of the computer module 105. Examples of transitory or non-tangible computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data to the computer module 105 include radio or infra-red
5 transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

The second part of the RSABO application programs 233 and the corresponding code modules mentioned above may be executed to implement one or more graphical user
10 interfaces (GUIs) to be rendered or otherwise represented upon the display 214 (see Figs. 5-9 for further detail). Through manipulation of typically the keyboard 202 and the mouse 203, a user of the computer system 200 and the application may manipulate the interface in a functionally adaptable manner to provide controlling commands and/or input to the applications associated with the GUI(s). Other forms of functionally adaptable user
15 interfaces may also be implemented, such as an audio interface utilizing speech prompts output via the loudspeakers 217 and user voice commands input via the microphone 280.

Fig. 2B is a detailed schematic block diagram of the processor 205 and a “memory” 234. The memory 234 represents a logical aggregation of all the memory modules (including the HDD 209 and semiconductor memory 206) that can be accessed by
20 the computer module 105 in Fig. 2A.

When the computer module 105 is initially powered up, a power-on self-test (POST) program 250 executes. The POST program 250 is typically stored in a ROM 249 of the semiconductor memory 206 of Fig. 2A. A hardware device such as the ROM 249 storing software is sometimes referred to as firmware. The POST program 250 examines

- 15 -

hardware within the computer module 105 to ensure proper functioning and typically checks the processor 205, the memory 234 (209, 206), and a basic input-output systems software (BIOS) module 251, also typically stored in the ROM 249, for correct operation. Once the POST program 250 has run successfully, the BIOS 251 activates the hard disk drive 210 of Fig. 2A. Activation of the hard disk drive 210 causes a bootstrap loader program 252 that is resident on the hard disk drive 210 to execute via the processor 205. This loads an operating system 253 into the RAM memory 206, upon which the operating system 253 commences operation. The operating system 253 is a system level application, executable by the processor 205, to fulfil various high level functions, including processor management, memory management, device management, storage management, software application interface, and generic user interface.

The operating system 253 manages the memory 234 (209, 206) to ensure that each process or application running on the computer module 105 has sufficient memory in which to execute without colliding with memory allocated to another process. Furthermore, the different types of memory available in the system 200 of Fig. 2A must be used in such a manner as to enable each process to run effectively. Accordingly, the aggregated memory 234 is not intended to illustrate how particular segments of memory are allocated (unless otherwise stated), but rather to provide a general view of the memory accessible by the computer system 200 and how such is used.

As shown in Fig. 2B, the processor 205 includes a number of functional modules including a control unit 239, an arithmetic logic unit (ALU) 240, and a local or internal memory 248, sometimes called a cache memory. The cache memory 248 typically include a number of storage registers 244 - 246 in a register section. One or more internal busses 241 functionally interconnect these functional modules. The processor 205

- 16 -

typically also has one or more interfaces 242 for communicating with external devices via the system bus 204, using a connection 218. The memory 234 is coupled to the bus 204 using a connection 219.

The RSABO application program 233 includes a sequence of instructions 231 that
5 may include conditional branch and loop instructions. The program 233 may also include data 232 which is used in execution of the program 233. The instructions 231 and the data 232 are stored in memory locations 228, 229, 230 and 235, 236, 237, respectively. Depending upon the relative size of the instructions 231 and the memory locations 228-230, a particular instruction may be stored in a single memory location as depicted by the
10 instruction shown in the memory location 230. Alternately, an instruction may be segmented into a number of parts each of which is stored in a separate memory location, as depicted by the instruction segments shown in the memory locations 228 and 229.

In general, the processor 205 is given a set of instructions which are executed therein. The processor 205 waits for a subsequent input, to which the processor 205 reacts
15 to by executing another set of instructions. Each input may be provided from one or more of a number of sources, including data generated by one or more of the input devices 202, 203, data received from an external source across one of the networks 109, 202, data retrieved from one of the storage devices 206, 209 or data retrieved from a storage medium 225 inserted into the corresponding reader 212, all
20 depicted in Fig. 2A. The execution of a set of the instructions may in some cases result in output of data. Execution may also involve storing data or variables to the memory 234.

The disclosed RSABO arrangements use input variables 254, which are stored in the memory 234 in corresponding memory locations 255, 256, 257. The RSABO arrangements produce output variables 261, which are stored in the memory 234 in

- 17 -

corresponding memory locations 262, 263, 264. Intermediate variables 258 may be stored in memory locations 259, 260, 266 and 267.

Referring to the processor 205 of **Fig. 2B**, the registers 244, 245, 246, the arithmetic logic unit (ALU) 240, and the control unit 239 work together to perform
5 sequences of micro-operations needed to perform “fetch, decode, and execute” cycles for every instruction in the instruction set making up the program 233. Each fetch, decode, and execute cycle comprises:

(a) a fetch operation, which fetches or reads an instruction 231 from a memory location 228, 229, 230;

10 (b) a decode operation in which the control unit 239 determines which instruction has been fetched; and

(c) an execute operation in which the control unit 239 and/or the ALU 240 execute the instruction.

Thereafter, a further fetch, decode, and execute cycle for the next instruction may
15 be executed. Similarly, a store cycle may be performed by which the control unit 239 stores or writes a value to a memory location 232.

Each step or sub-process in the process fragment 300 of **Fig. 3** is associated with one or more segments of the program 233 and is performed by the register section 244, 245, 247, the ALU 240, and the control unit 239 in the processor 205 working together to
20 perform the fetch, decode, and execute cycles for every instruction in the instruction set for the noted segments of the program 233.

The RSABO arrangements may alternatively be implemented in dedicated hardware such as one or more gate arrays and/or integrated circuits performing the RSABO functions or sub functions. Such dedicated hardware may also include graphic

- 18 -

processors, digital signal processors, or one or more microprocessors and associated memories. If gate arrays are used, the process flow chart for the process fragment 300 in Fig. 3 is converted to Hardware Description Language (HDL) form. This HDL description is converted to a device level netlist which is used by a Place and Route (P&R) tool to produce a file which is downloaded to the gate array to program it with the design specified in the HDL description.

Fig. 3 shows two process fragments, a fragment 300 that is performed by the user computers such as the user computer 105, and a fragment 320 that is performed by the server 103.

10 The process fragment 300, which is depicted from the user's perspective, is performed by a user computer such as the user computer 105, and commences with a decision step 301 in which the processor 205 determines, in response to a manual command via the keyboard 202 or the mouse 203, or in response to a data input over the network 109, whether new content is to be processed into a new playlist. Fig. 5 depicts a GUI by which the user can direct, by inputting data into windows 501, 502 and then clicking on a button 503, the processor 205 that new content is to be received. Alternately, file based uploads can be performed by dragging and dropping the desired content onto a browser window displaying on the user computer display.

If new content is not to be created and/or processed, then the process 300 follows a NO arrow 302 back to the step 301. If however new content is to be created, then the process 300 follows a YES arrow 303 from the step 301 to a step 304. In the step 304 the processor 205 receives the content data according, for example, to the inputs provided as depicted in Fig. 5, and then the process 300 follows an arrow 305 to a step 306.

- 19 -

In the step 306, the processor 205 creates a playlist using new content as illustrated by the GUI screen shots in Fig. 6 in which content such as 601 is dragged and dropped into a playlist 602, or modifies an existing playlist. The process 300 then follows an arrow 307 to a step 308. In the step 308 the processor 205 displays the playlist created thus far for the user to review. This is depicted in the GUI in Fig. 7 in which the content of the playlist 602 is displayed at 701, with a run time 702 which may be adjusted by entering a different run time at 703. A play button (not shown) is used for previewing a playlist and opens a replica of the player on the display computer in a new display pane in the display. The process 300 then follows an arrow 309 to an optional step 310.

10 In the step 310 the processor 205 optionally schedules the new playlist (if an existing playlist is being modified by the step 306 then the playlist will already have an associated schedule) as illustrated by a GUI 800 in Fig. 8. In Fig. 8 the aforementioned playlist 602 is scheduled to start and end as depicted at 801 and 802 respectively. Other features such as "repeat" may also be used as depicted in Fig 9 at 901 and 902. The playlist
15 is then inserted into a calendar as depicted at 803. The process 300 then follows an arrow 318 to a step 319.

In the step 319 the processor 205 uploads the scheduled playlist and the associated scheduling information to the server 103. The process 300 then follows an arrow 311 back to the step 301.

20 The disclosed RSABO arrangements provide for a prioritisation mechanism whereby potentially competing (i.e. colliding) presentations on a particular display device can be scheduled. In one RSABO arrangement, colliding presentations are queued and displayed in queued order.

- 20 -

For example consider the example in which user A^1 wishes to display playlist P^1 in timeslot X^1 and user A^2 wishes to display playlist P^2 in timeslot X^1 . If user A^1 submitted their scheduled playlist to the server 103 before user A^2 , then, presuming that all timeslots are available for the sake of explanation, in one arrangement the playlist P^1 will be
5 displayed on the noted display device in the timeslot X^1 and the playlist P^2 will be displayed on the noted display device in the timeslot immediately following the timeslot X^1 .

In another arrangement, if two users each schedule a playlist that is 30 seconds long for presentation at the same time over the duration of an hour, then each user will
10 receive 30min of play time where each user's playlist is alternatively played, starting with the user who first submitted their schedule to the server.

In another arrangement, if user A^1 schedules a 45min playlist to play at a particular time for an hour, and a second user A^2 schedules a 10min playlist for the same hour, then the user A^1 will have their playlist play for 45min, then user A^2 will have theirs
15 play for 10min, and then user A^1 will have theirs play the first 5min again before the schedules expire.

In another arrangement, if an "exclusive" tag is assigned to a particular presentation, then provided that no other presentations are allocated an "exclusive" tag, the exclusive presentation is granted priority for the particular time slot selected for display on
20 the display device in question.

One or more "emergency" tags are also available, these being able to override all other types of tags for presentation priority however this tag is typically only available to administrators. If user A^1 schedules an emergency playlist, then their play list will play in preference to a display from any other user who schedules at that time. If someone else

- 21 -

schedules an emergency playlist after the first emergency playlist is scheduled, then they will be notified that their presentation will not be displayed until the first schedule has expired).

Fig. 10 shows an update process 1000 performed by the server 103. In one arrangement, the update process 1000 is performed once a minute, each process cycle performing batch processing of any changes made by any user in the RSABO arrangement. The update process generates cache files that represent the update. If there are changes, a command is sent (see 433 in Fig. 4) to the affected browsers in the client display machines, directing the client machines to download (see 423 in Fig. 4B) the cached file. Polling browsers (see 438 in Fig. 4B) in the client display machines will alternately download (see 423 in Fig. 4B) the cached file if a date stamp has changed since their last download, or if once polled the server indicates that a download is needed.

The process 1000 commences with a decision step 1001 in which the server processor (not shown) determines if a new scheduled playlist and associated schedule material has been received from one of the user computers 105, ..., 107. If a new scheduled playlist and associated schedule material has not been received from one of the user computers 105, ..., 107, the process follows a NO arrow 1001 back to the step 1001.

If a new scheduled playlist and associated schedule material has been received from one of the user computers 105, ..., 107, then the process 1000 follows a YES arrow 1002 to an update process 320, which is described in more detail below with reference to Fig. 3.

The process 1000 then follows an arrow 1003 to a decision step 1005 in which the server processor determines if a download command has been received from one of the display computers 110, ..., 112. If this is not the case, then the process 1000 follows a NO

- 22 -

arrow back to the step 1005. If a download command has been received from one of the display computers 110, ..., 112 then the process 1000 follows a YES arrow 1006 to a step 1007. In the step 1007 the server processor downloads the new scheduled playlist and associated schedule material to the display computers 110, ..., 112. The process 1000 then
5 follows an arrow 1008 back to the step 1001.

Returning to Fig. 3, turning to the process fragment 320, this process commences with a step 312 in which the processor (not shown) of the server 103 traverses a playlist that has been uploaded at a step 319 of process fragment 300 by one of the user computers 105, ..., 107. The process 320 then follows an arrow 313 to a decision step 314. In the step
10 314 the processor (not shown) of the server 103 determines if an update command is required. An update command is required when, for example, the set of the display machines 110 to which the uploaded playlist is directed have not been sent an update command for a specified time period. If the step 314 determines that an update command is required, then the process 320 follows a YES arrow 315 from the step 314 to a step 316. In
15 the step 316 the processor (not shown) of the server 103 sends an update command, which is received in a step 433 in Fig. 4B by the designated set of display computers. The update command sent by the step 316 is only sent to a designated set of the display machines.

Polling machines such as the display computer 110 check date stamps in their individual update files which are handled by the web server (which may be based upon IIS
20 or Apache standards) to help reduce load on the server CPU and the database101. These files also allow for memory caching to enable fast request handling by the server 103 for polling browsers such as 115.

Other update methods can be used. Thus for example, communication between the server(s) 103 and the display computers such as 110 can utilise the Web-socket

- 23 -

standard, if this is supported by the RSABO browsers, noting that presently not all browsers support Web-sockets.

In one RSABO arrangement, an "If-Modified-Since" request header is used on polling browsers, in which event the web servers return either a "Not Modified" response,
5 or alternately, send the update file to the polling machine.

The process 320 then follows an arrow 317 back to the step 312. Returning to the decision step 314, if an update command is not required, then the process 320 follows a NO arrow 321 back to the step 312.

Figs. 4A – 4C depict a process 400 showing how the display computers such as
10 the display computer 110 operate in the RSABO arrangements. Browsers power up and load the last list and version of the player software they had previously uploaded. The display computers start presenting that content until they can download new content if any is available. This makes use of "offline caching" and "load to cache" operations, to enable the browser to be launched according to the step 405 in **Fig. 4A** while the display computer
15 is not connected to the server. As soon as the display computer connects to the server, the display computer can download player and playlist updates. As depicted in **Fig. 4A**, the display computer 110 displays the "offline content" (ie the previously loaded content) before checking for the availability of updates as depicted by the step 421 in **Fig. 4B**. Not all browsers support offline caching, however such browsers can, after initially
20 downloading the player software and an initial playlist, be disconnected from the server and continue to operate.

Turning to **Fig. 4A**, the process 400 commences with a start step 401 after which the process 400 follows an arrow 402 to a power up step 403. The typical power up process IS described hereinafter with reference to **Figs. 2A and 2B** in more detail. The process 400

- 24 -

then follows an arrow 404 to a step 405 in which the processor (not shown) of the display computer launches a web browser application. The process 400 then follows an arrow 406 to a step 407 which loads a previously retrieved player specification (ie a previously retrieved player application and the set of playlists for presentation, or a pre-loaded default
5 player application and set of playlists for presentation) if the display computer is off line and communications are not presently available to the server 103. The web browser then performs two different processes 409 and 412 as follows.

The process 400 follows an arrow 408 from the step 407 to the step 409 which updates the set of playlists as described hereinafter with reference to **Fig. 4B**. The process
10 400 then follows an arrow 410 in a looping manner from the step 409 back to the step 409.

The process 400 also follows an arrow 411 from the step 407 to the step 412 which presents the presently loaded playlists as described hereinafter with reference to **Fig. 4C**. The process 400 then follows an arrow 413 in a looping manner from the step 412 back to the step 412.

15 Turning to **Fig. 4B**, the process 409 commences with a test step 442 which determines whether the display computer has just powered up. If this is the case then the process 409 follows a YES arrow 420 to a decision step 421 which determines if an update is available. In this step the processor of the display computer signals the sever 103 to make the relevant enquiry. If an update is available, then the process 409 follows a YES
20 arrow 422 to a step 423 which downloads the update from the server 103 to the display computer. The process 409 then follows an arrow 424 to a step 425 which determines if the update is a priority update (eg whether the update has an "exclusive" tag). If this is the case then the process 409 follows a YES arrow to a step 427 which causes the display computer to immediately substitute the priority playlist for the current playlist and present the

- 25 -

priority playlist. Returning to the step 425, if the update is not a priority update, then the process 409 follows a NO arrow 429 to a step 430 which schedules the newly received update for display and presentation as soon as the current presentation has finished. The process 409 then follows an arrow 431 to the step 433.

5 Returning to the step 427, the process 409 then follows an arrow 428 to the step 433 which determines if a server command (such as the update command from the step 316 in Fig. 3), has been received. If this is the case, then the process 409 follows a YES arrow 434 to the step 423.

 If however the step 433 indicates that a server command has not been received,
10 then the process 409 follows a NO arrow 435 to a decision step 436 which determines if a polling timeout time stamp has expired. If this is the case then the process 409 follows a YES arrow 437 to a step 438 which causes the display computer to poll the server 103 to see if an update is forthcoming. The process 409 then follows an arrow 439 to the step 433.

 Returning to the step 436 if the polling timeout has not expired, then the process
15 409 follows an arrow 440 to the step 433. Returning to the step 442, if the step determines that the display computer has not just powered up, then the process 409 follows a NO arrow 432 to the step 433.

 Turning to Fig. 4C, the process 412 commences with a step 450 which presents the current item of the presently loaded presentation. The process 412 then follows an
20 arrow 451 to a decision step 452 which determines if the current item is finished. If this is the case then the process 412 follows an arrow 453 to a step 454 which sets the next item in the list as the current item, and the process 412 follows an arrow 455 back to the step 450 which displays the current item. Returning to the step 452, if the current item has not

- 26 -

yet finished being displayed, then the process 412 follows a NO arrow 456 back to the step 450.

Fig. 11 shows a process 1100, composed of two process fragments 1110 and 1111, according to which a user computer, such as 105 in **Fig. 1**, creates new content and uploads it for storage into the server (i.e. 103 in **Fig. 1**). **Fig. 11** is partitioned down the centre into two segments, a left-hand segment relating to processes performed by the user computer 105, and a right-hand segment relating to processes performed in the server 103.

Turning to the processes associated with the user computer 105, the process fragment 1110 commences with a step 1102 in which the processor in the user computer determines if new content is to be created. If this is the case, then the process is directed from the decision step 1102 according to a YES arrow 1103 to a step 1104. In the step 1104 the user computer, in conjunction with and under the control of a user, creates new content, this content typically comprising one or more playlists. The process fragment 1110 is then directed according to an arrow 1105 to a step 1106 in which the user computer uploads, as depicted by a dashed arrow 1108, the newly created content to the server 103.

Turning to the process fragment 1111, the server 103, in a step 1109, receives the aforementioned new content data, and stores this in the content database 101.

Returning to the process fragment 1110 associated with a user computer, after the user computer has uploaded the content to the server in the step 1106, the process fragment 1110 is directed from the step 1106 as depicted by an arrow 1107 back to the decision step 1102.

Returning to the decision step 1102, if there is no new content to be created, then the process fragment 1110 follows a NO arrow 1101 from the step 1102 back to the decision step 1102, in a looping fashion.

- 27 -

Fig. 12 shows two pairs of process fragment, namely 1240 and 1241, and 1242 and 1243. The process fragment pair 1240 and 1241 relates to creation or modification of a playlist by the user computer 105, and upload of the newly created or newly modified playlist to the server 103. The process fragment pair 1242 and 1243 relates to review of a
5 playlist by the user computer 105, and publication of the review playlist by the server 103 upon receipt of a publish command from the user computer 105.

Turning firstly to the process fragment pair 1240 and 1241, the process fragment 1240 commences with a decision step 1202 in which the processor of the user computer 105 determines whether a new playlist is to be created, or an existing playlist is to be
10 modified. If this is not the case, then the process fragment 1240 follows a NO arrow 1201 from the decision step 1202 back to the decision 1202 in a looping fashion. If, however, a new playlist is to be created or an existing playlist is to be modified, then the process fragment 1240 follows a YES arrow 1203 from the decision step 1202 to a step 1204. In the step 1204 the user computer 105, in conjunction with commands from the user, creates
15 a new playlist or modifies an existing playlist. The process fragment 1240 then follows an arrow 1205 from the step 1204 to a step 1206. In the step 1206 the user computer uploads, as depicted by a dashed arrow 1208, the newly created or modified playlist to the server 103.

Turning to the process fragment 1241, the server 103, in a step 1209, saves the
20 newly created or modified playlist as an unpublished playlist in the database 101.

Turning now to the process fragment pair 1242 and 1243, and on the fragment 1242 associated with the user computer 105 in the first instance, the process fragment 1242 commences with a decision step 1222 in which the processor in the user computer 105 determines whether a playlist is to be reviewed. If this is not the case, then the process

- 28 -

fragment 1242 follows a NO arrow 1221 from the decision step 1222 back to the decision step 1222 in a looping fashion. If, on the other hand, a playlist is to be reviewed, then the process fragment 1242 follows a YES arrow 1223 from the decision step 1222 to a step 1224.

5 In the step 1224 the user computer 105, in conjunction with the user, reviews the playlist by making changes and amendments as desired. Thereafter, the process fragment 1242 follows an arrow 1226 from the step 1224 to a decision step 1228. In the step 1228 the user computer 105 determines, based upon inputs, for example, from the user, whether the reviewed playlist is to be published. If this is not the case, then the process fragment
10 1242 follows a NO arrow 1225 from the decision step 1228 back to the decision step 1222. If, on the other hand, the reviewed playlist is to be published, then the process fragment 1242 follows an arrow 1229, depicted as a YES arrow, from the decision step 1228 to a step 1230.

 In the step 1230 the user computer 105 sends a "PUBLISH" command to the
15 server 103. The process fragment 1242 then follows an arrow 1227 from the step 1230 back to the decision step 1222.

 Turning to the process fragment 1243 which is associated with the server 103, the process fragment 1243 commences with a decision step 1232 in which the server 103 processor determines whether a PUBLISH command has been received. If this is not the
20 case, then the process fragment 1243 follows a NO arrow 1231 from the decision step 1232 back to the decision step 1232. If, on the other hand, the server has received a PUBLISH command, then the process fragment 1243 follows a YES arrow 1233 from the decision step 1232 to a step 1234. In the step 1234 the server publishes the playlist, in other words the server makes the reviewed playlist available for download by display computers such

- 29 -

as 115 which belong to the appropriate user group. The process fragment 1243 then follows an arrow 1236 from the step 1234 to a decision step 1237. In the step 1237 the server 103 determines if any schedules of existing playlists have been affected by the newly published playlist. If this is not the case, then the process fragment 1243 follows a
5 NO arrow 1235 from the decision step 1237 to the decision step 1232. If, on the other hand, any schedules have been affected, then the process fragment 1243 follows a YES arrow 1238 from the decision step 1237 to a step 1239.

In the step 1239 the server 103 determines which display computers such as 115 should receive an update command. Display computers which should receive the update
10 command are those display computers which are currently displaying, or are due to display, the published playlist. The server 103 then sends relevant update commands to those display computers such as 110 which require this command. The process fragment 1243 then follows an arrow 1240 from the step 1239 back to the decision step 1232.

Returning to Fig. 11, it is noted that the process fragments 1110, 1111 reflect the
15 fact that the server 103 can be used as a content library in order to store content that users can draw upon in creating and/or modifying playlists. This content can also be stored elsewhere, however storage on the server is a particularly advantageous arrangement in that appropriate controls and access can be incorporated into the RSABO system as desired.

20 Turning to Fig. 12, it is noted that the creation and/or modification of a playlist is a free-standing process which can be performed whenever it is desired. This is a separate process, as shown in Fig. 12, from the review process fragment 1242 and the publication process fragment 1243.

Industrial Applicability

- 30 -

The arrangements described are applicable to the computer and data processing industries and particularly those segments concerned with the display of visual and audio-visual content.

The foregoing describes only some embodiments of the present invention, and
5 modifications and/or changes can be made thereto without departing from the scope and spirit of the invention, the embodiments being illustrative and not restrictive.

- 31 -

CLAIMS:

1. A system for displaying content, the system comprising:
one or more user computers for creating the content and uploading the content to a
5 server;
the server for communicating an update command to a plurality of display
computers over a network; and
the plurality of display computers, having respective display devices and running
respective web browser software programs having associated cache memories; wherein:
10 the web browsers are configured to receive the update command and in response
thereto to communicate with the server, download the content into the respective cache
memories, and display the content on the respective display devices.
2. A system according to claim 1, wherein:
15 the downloaded content comprises associated information defining a duration
parameter for which the content is to be displayed by the display computers; and
the web browsers are configured to display the content for the duration defined by
the associated information and to display default content after the duration has expired.
- 20 3. A system according to claim 2, wherein:
the web browsers are further configured to interrupt the display of the content if a
further update command is received, and in response to the further update command, to
communicate with the server, download further content into the respective cache
memories, and display the further content on the respective display devices.

- 32 -

4. A system according to claim 1, wherein the web browsers are further configured, upon power up of the respective display computers, to communicate with the server, download a default display content from the server; and display the default content until
5 scheduled otherwise or an update command is received.

5. A system according to claim 1, wherein the web browsers are further configured, upon power up of the respective display computers, to communicate with the server, retrieve information defining a player software program, and download the defined player
10 software program for displaying the content.

6. A server for use in a system for displaying content, the server comprising:
a memory for storing a program; and
a processor for executing the program, said program directing the processor to
15 execute a method for displaying visual content, the method comprising the steps of:
communicating an update command to a plurality of display computers over a
network, said update command indicating that new content is available for display; and
uploading the new content to the plurality of display computers upon receipt from
said display computers of communications in response to the update command.

20

7. A method according to which a system displays content, the method comprising the steps of:
creating by one or more user computers the content and uploading the content to a
server;

- 33 -

communicating by the server an update command to a plurality of display computers over a network, the display computers having respective display devices and running respective web browser software programs having associated cache memories;

receiving by the web browsers the update command and in response thereto:

5 communicating with the server;

downloading the content into the respective cache memories; and

displaying the content on the respective display devices.

8. A method according to claim 7, wherein:

10 the downloaded content comprises associated information defining a duration parameter for which the content is to be displayed by the display computers; and

the displaying step performed by the web browsers comprises displaying the content for the duration defined by the associated information; and

displaying default content after the duration has expired.

15

9. A method according to claim 7, comprising the further steps of:

interrupting the display of the content if a further update command is received;

communicating, in response to the further update command, with the server;

downloading further content into the respective cache memories; and

20 displaying the further content on the respective display devices.

10. A method performed by a server in a system for displaying content, the method comprising the steps of:

receiving content from at least one user computer;

- 34 -

traversing a schedule associated with the content to determine if an update is to be sent to at least one display computer;

if an update is to be communicated, communicating by the server an update command to the at least one display computer over a network, the display computer
5 having a display device and running a web browser software program having an associated cache memory;

receiving, in response to the communicated update command, a download communication from the at least one display computer; and

downloading, in response to the received download communication, the content
10 into the cache memory.

11. A system for displaying content, said system comprising:

a plurality of memory modules for storing a software program comprising a plurality of program modules; and

15 a plurality of processors communicating with the plurality of memory modules for executing the program, said program directing the plurality of processors to execute a method for displaying content, the method comprising the steps of:

creating by one or more user computers the content and uploading the content to a server;

20 communicating by the server an update command to a plurality of display computers over a network, the display computers having respective display devices and running respective web browser software programs having associated cache memories;

receiving by the web browsers the update command and in response thereto:

communicating with the server;

- 35 -

downloading the content into the respective cache memories; and
displaying the content on the respective display devices.

12. A computer readable non-transitory tangible storage medium having a computer
5 program recorded therein, the program being executable by a server to perform a method
for displaying content, the method comprising the steps of:

receiving content from at least one user computer;

traversing a schedule associated with the content to determine if an update is to be
sent to at least one display computer;

10 if an update is to be communicated, communicating by the server an update
command to the at least one display computer over a network, the display computer
having a display device and running a web browser software program having an associated
cache memory;

receiving, in response to the communicated update command, a download
15 communication from the at least one display computer; and

downloading, in response to the received download communication, the content
into the cache memory.

13. A system for displaying content comprising one or more playlists, the system
20 comprising one or more servers, a plurality of user computers, and a plurality of display
computers communicating over a telecommunications network, wherein:

the user computers and the display computers run respective web browser
software applications having associated cache memories;

- 36 -

the user computers are configured for creating or modifying the content and downloading the content to the servers;

the servers are configured for storing the content in one or more databases and sending an update command to those display computers having corresponding display
5 schedules which are affected by the content; and

the display computers are configured for uploading the content into their respective cache memories in response to receiving the respective update commands, and displaying the content.

10 14. A server configured for (a) storing content received from one or more user computers in one or more databases, (b) sending an update command to display computers having corresponding display schedules which are affected by the content, and (c) downloading the content into respective cache memories of the display computers.

15 15. A server according to claim 14, wherein the content is stored as an unpublished playlist in the one or more databases, and the server is configured for publishing the availability of the playlist in response to a publish command communicated to the server by a user computer after the unpublished playlist has been reviewed.

20 16. A display computer running a web browser software application having an associated cache memory, the display computer being configured for (a) receiving an update command from a server, (b) uploading content into the cache memory from the server in response to the update command, and (c) displaying the content.

- 37 -

17. A display computer according to claim 16, wherein the display computer is configured for, prior to receiving the update command, displaying either previously uploaded content or pre-loaded default content.

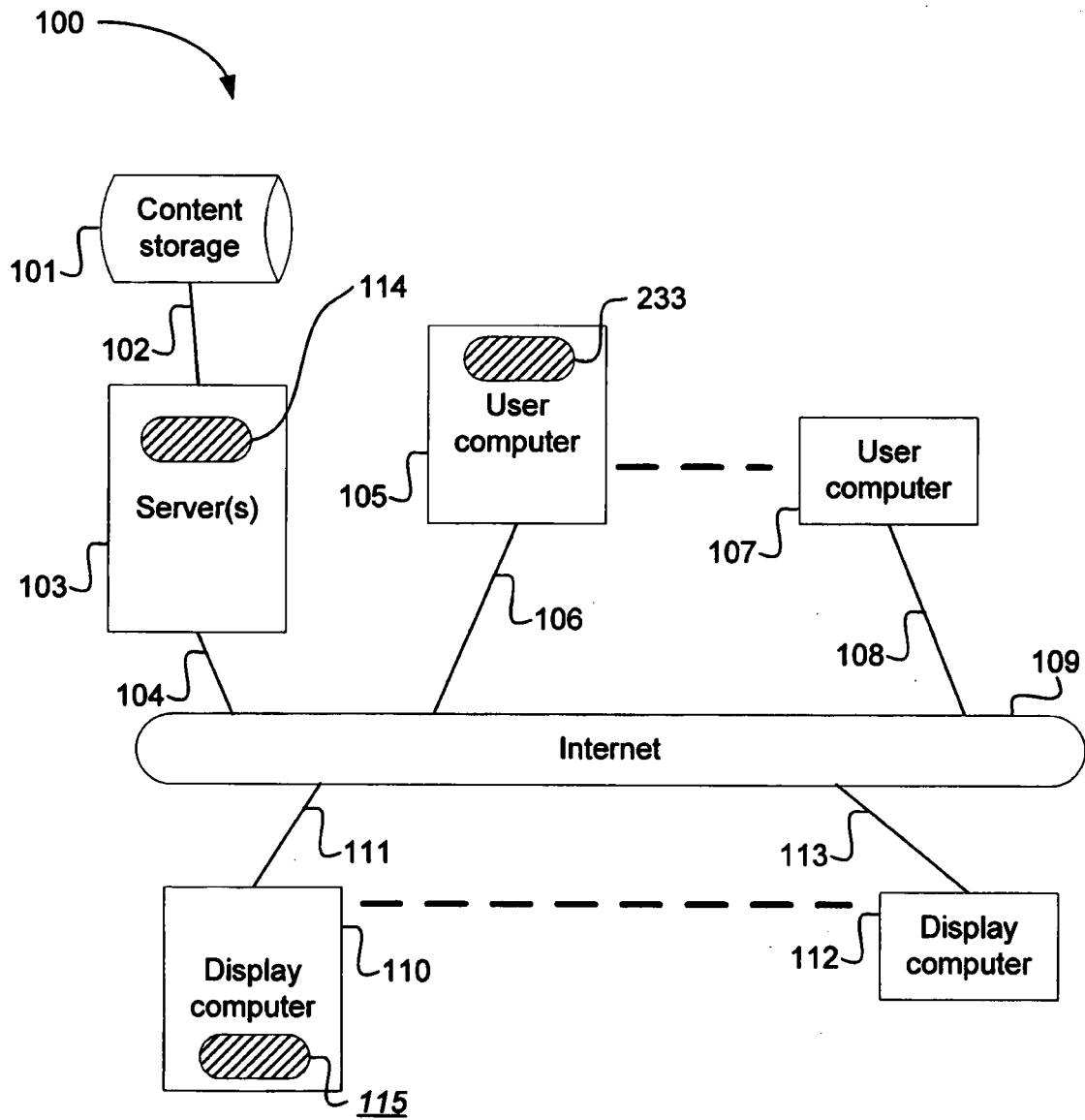
5 18. A display computer according to claim 16, wherein (a) if the uploaded content is in a priority playlist then the display computer substitutes the priority playlist for a current playlist and displays the priority playlist immediately, and (b) if the uploaded content is not in a priority playlist, then the display computer schedules the content for display as soon as the current playlist item has finished.

10

19. A display computer according to claim 16, wherein the display computer is configured for, if a polling timeout time stamp has expired without an update command being received from the server, polling the server to determine if an update command is forthcoming.

15

1/15

**Fig. 1**

2/15

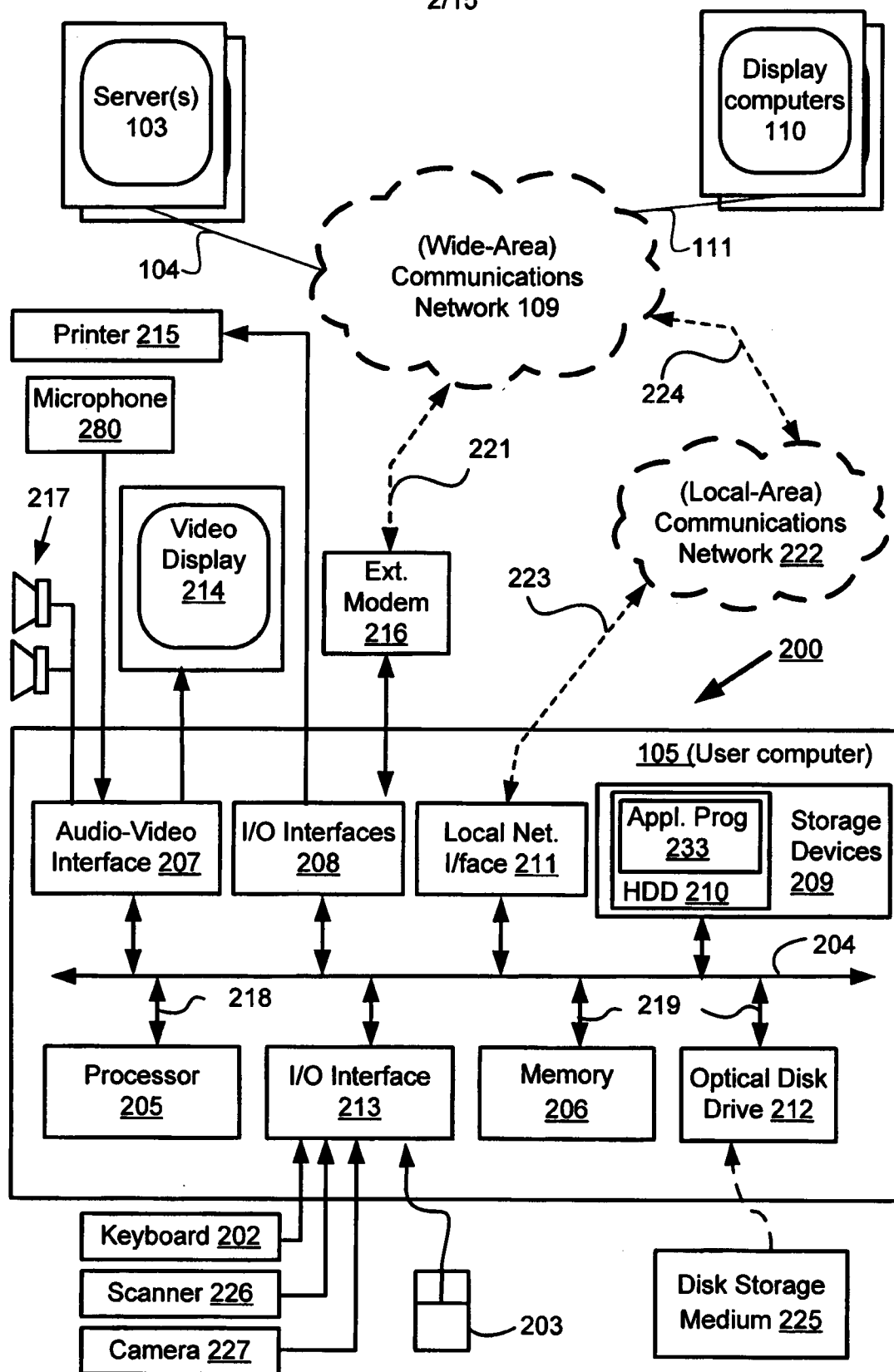


Fig. 2A

3/15

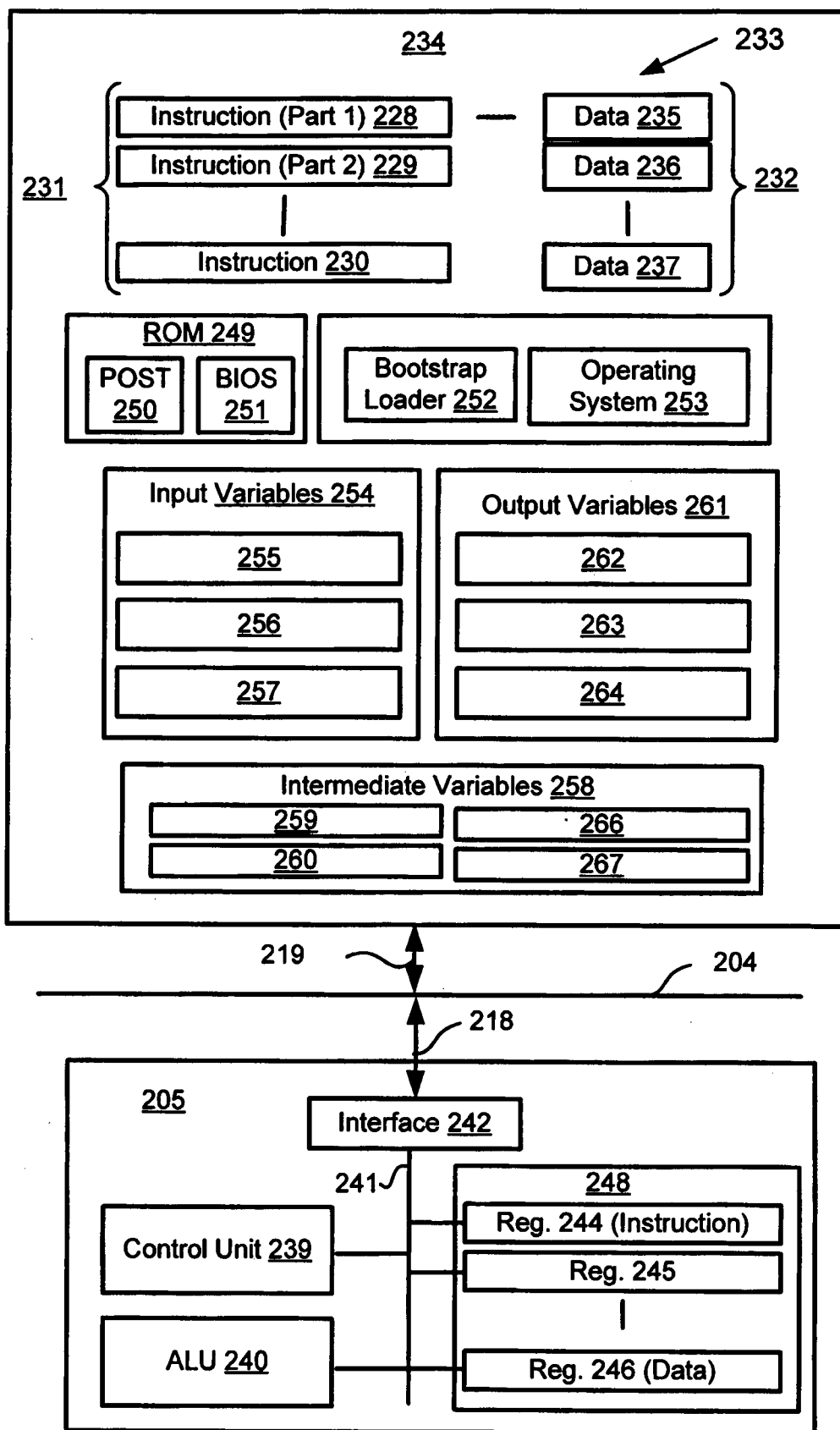
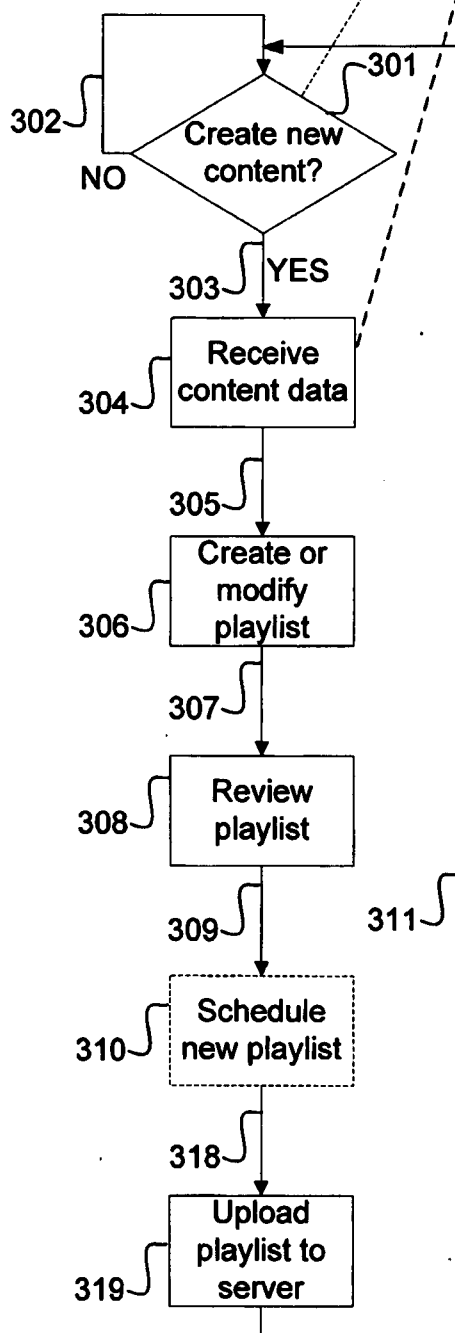


Fig. 2B

4/15

(Performed by the
user computer 105)

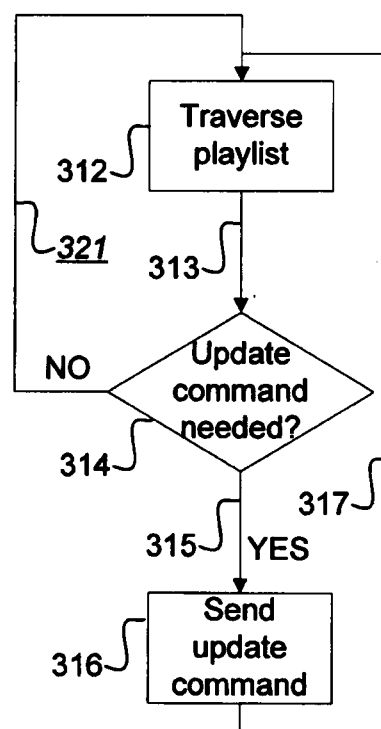
300



Manual command
via keyboard 202
or mouse 203, or
input data over
network 109

(Performed by server 103)

320

**Fig. 3**

5/15

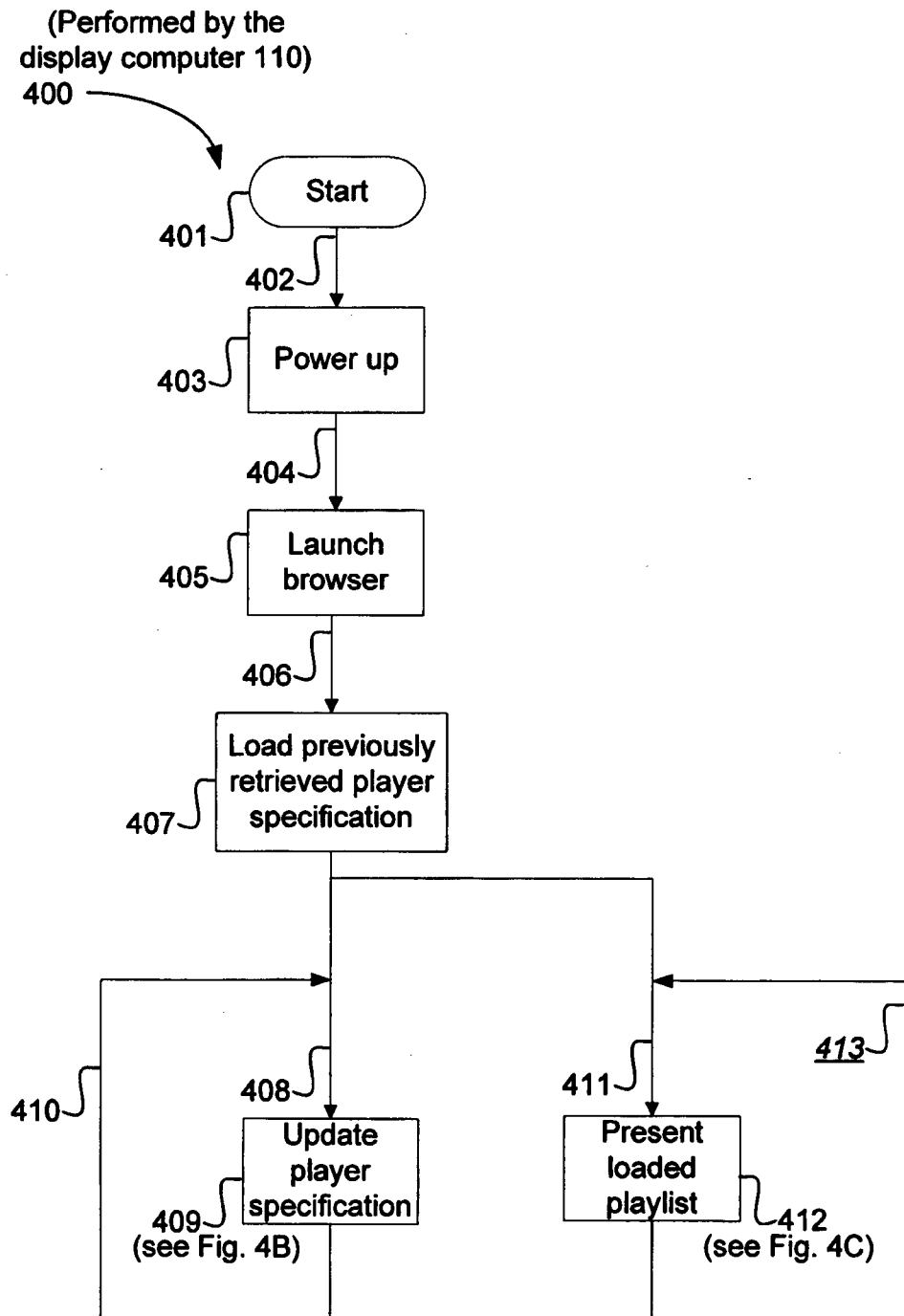


Fig. 4A

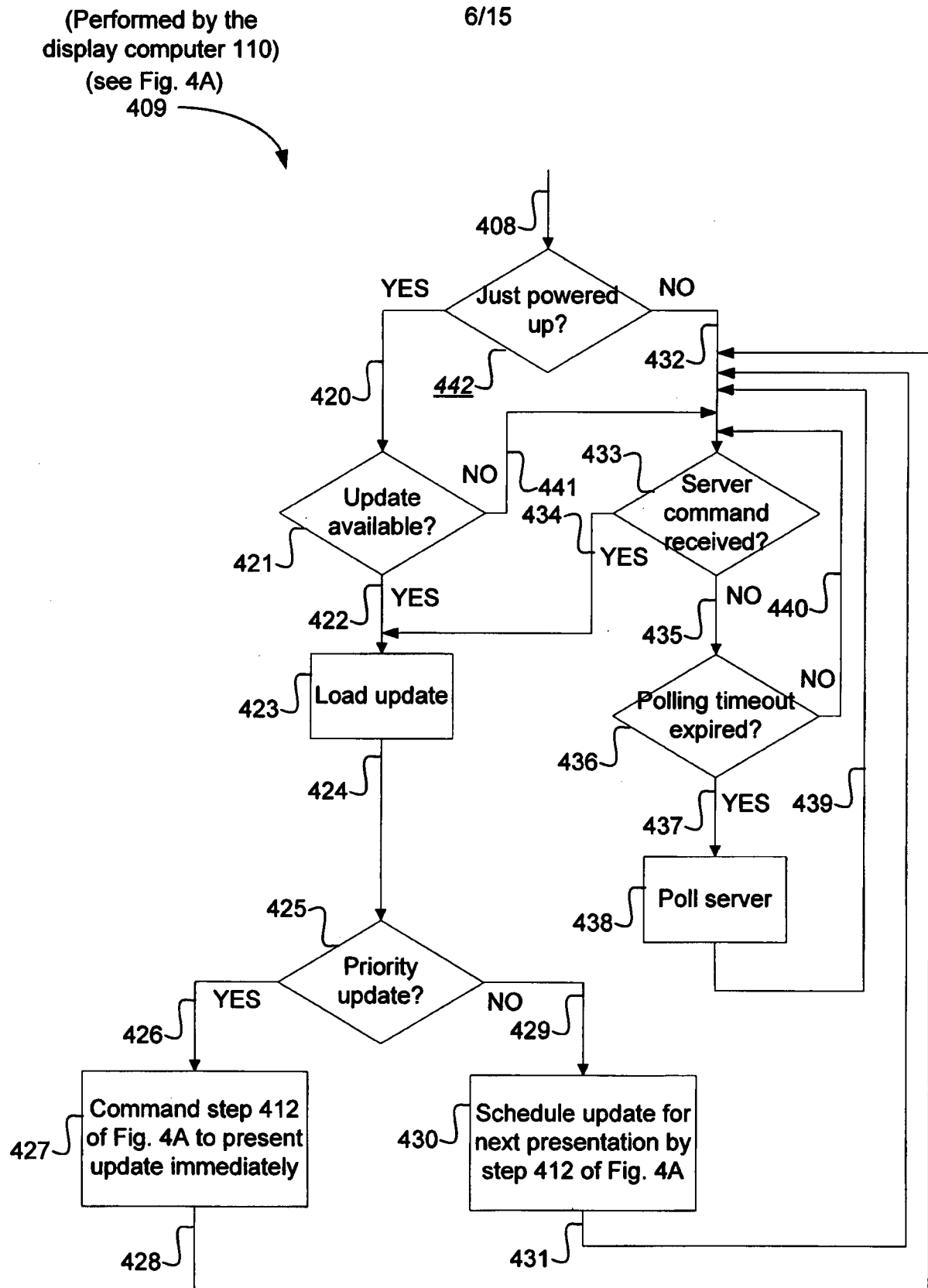
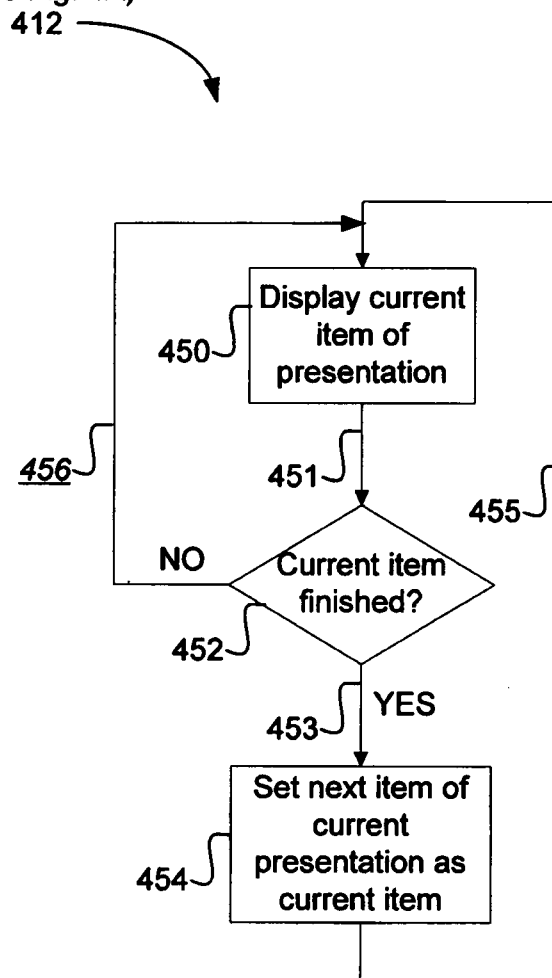


Fig. 4B

7/15

(Performed by the
display computer 110)
(see Fig. 4A)

**Fig. 4C**

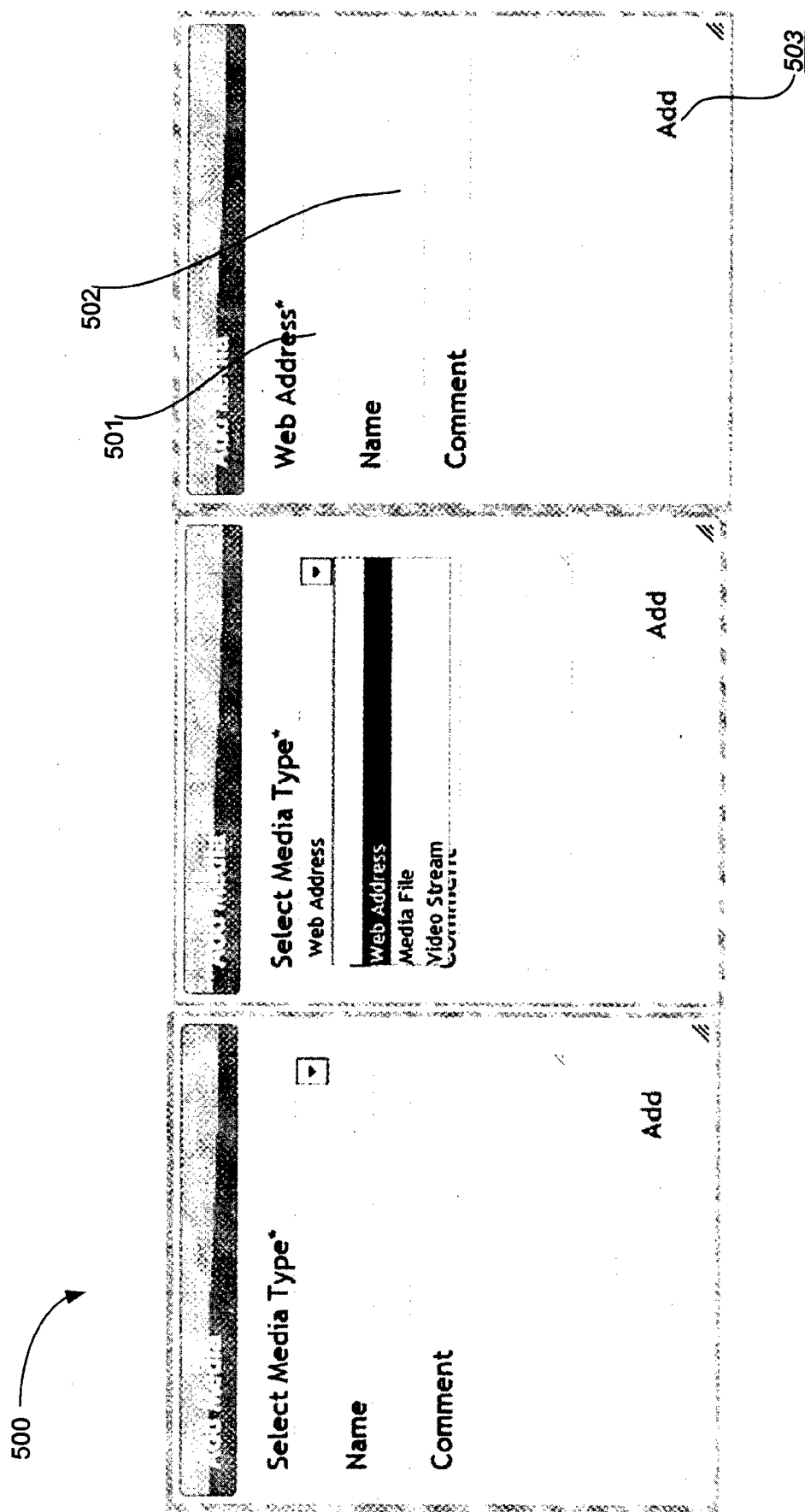


Fig. 5

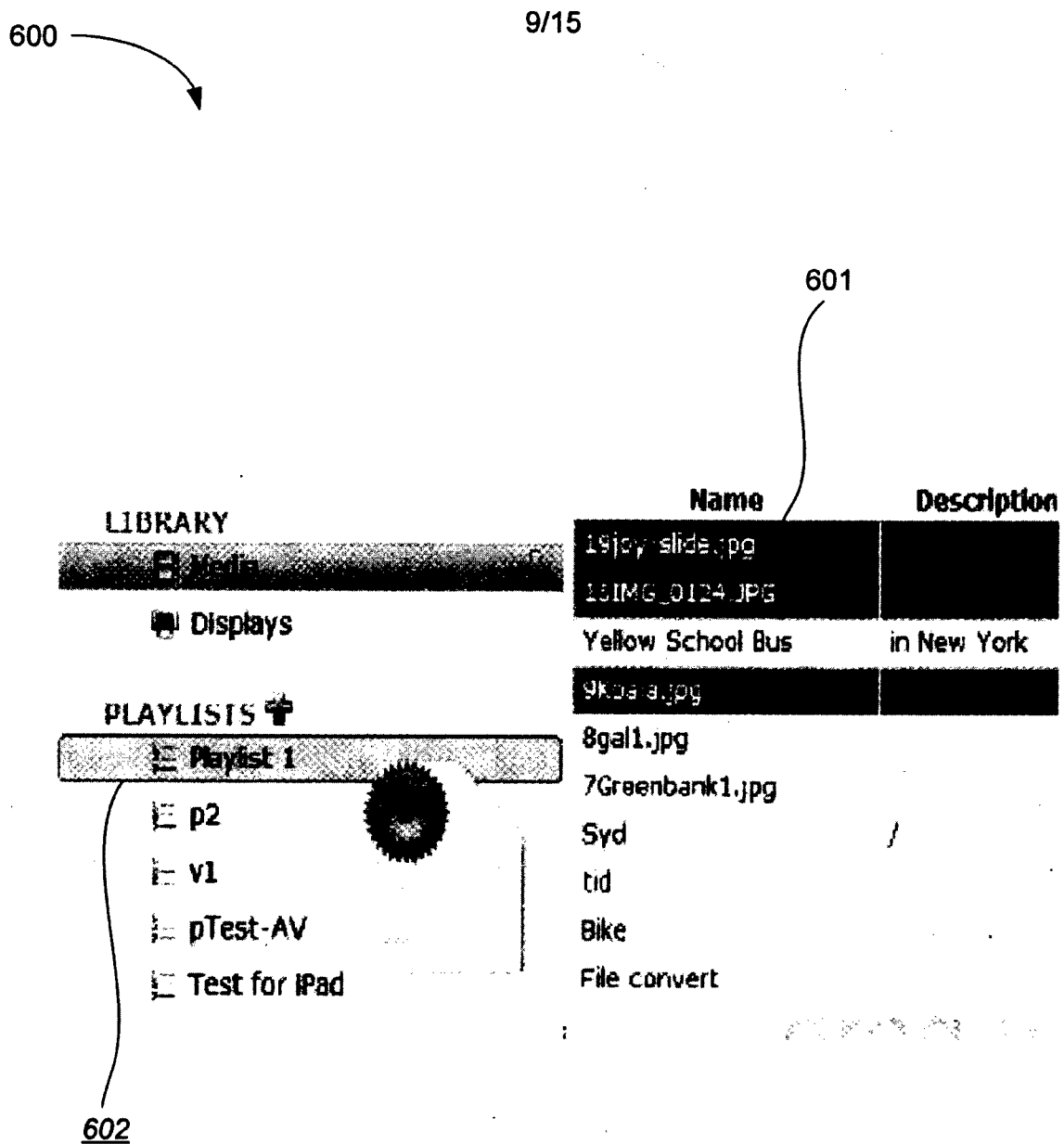


Fig. 6

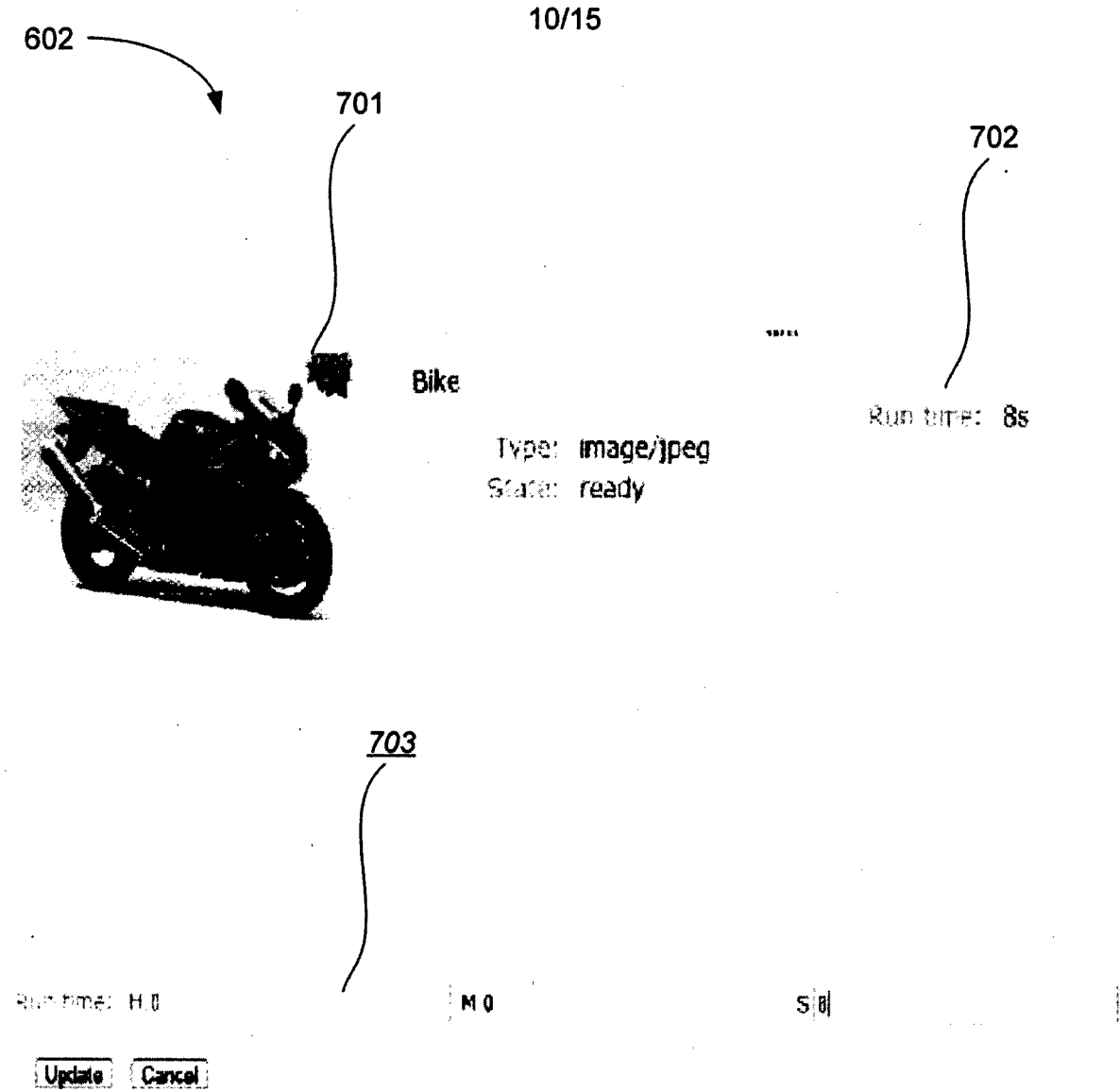


Fig. 7

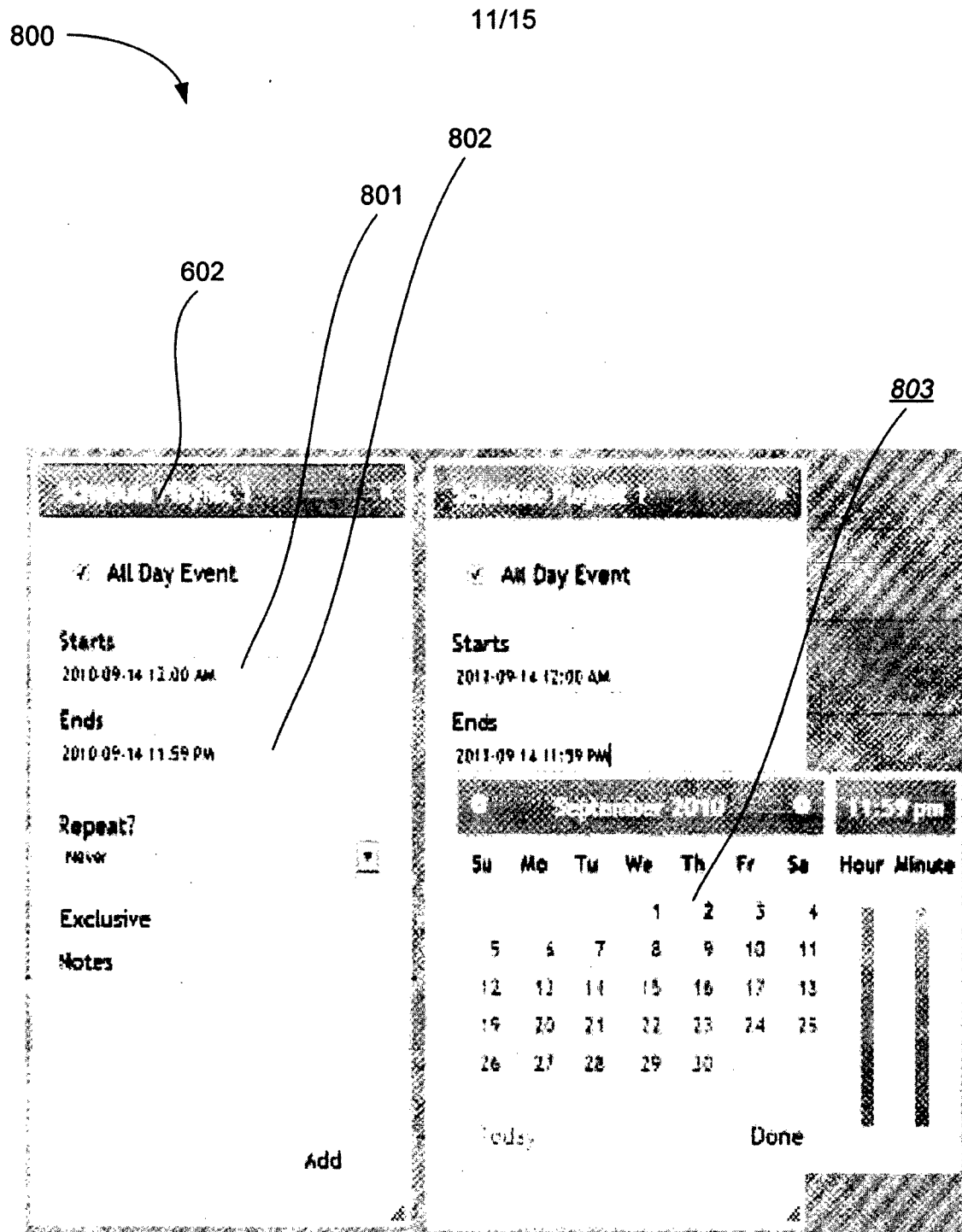


Fig. 8

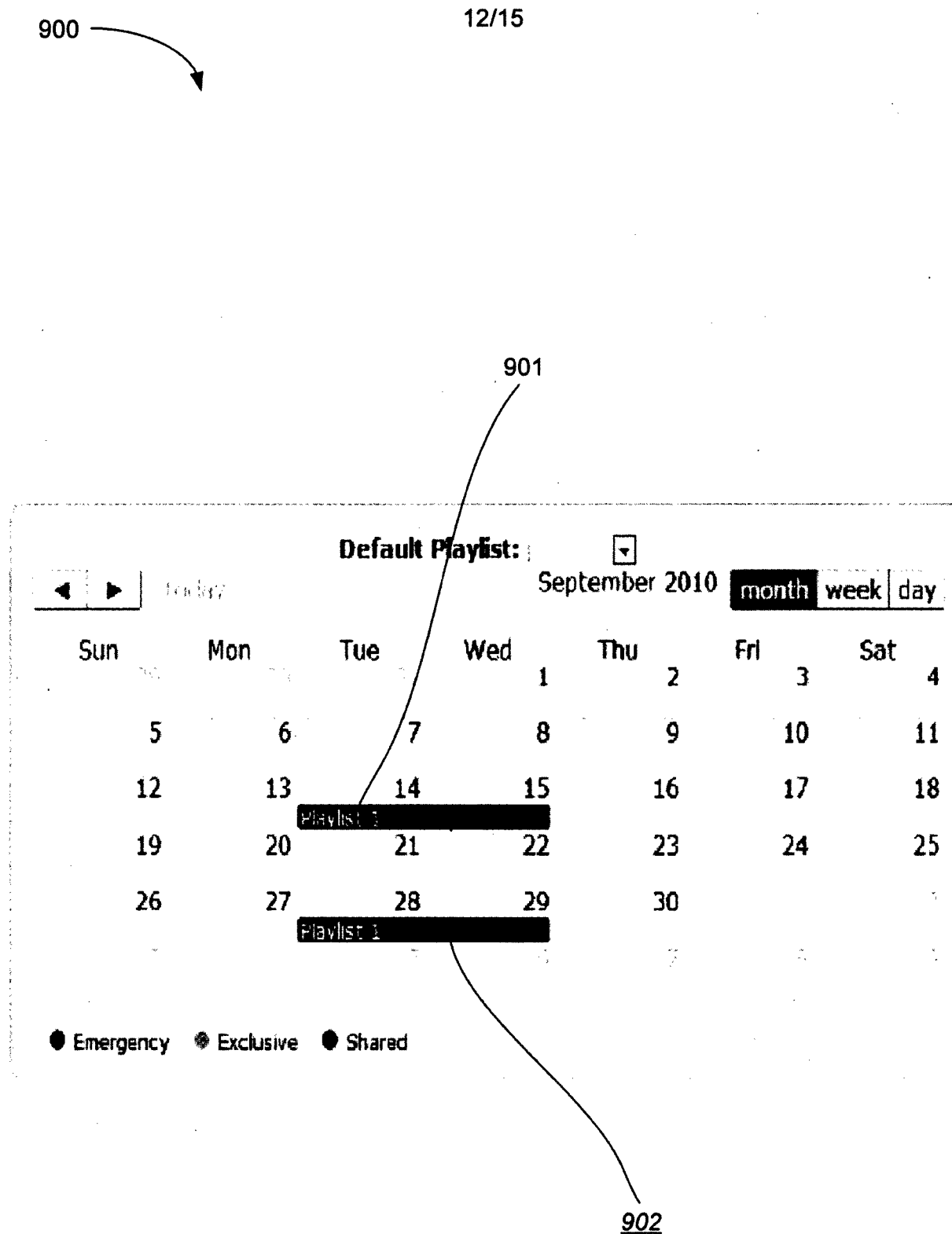
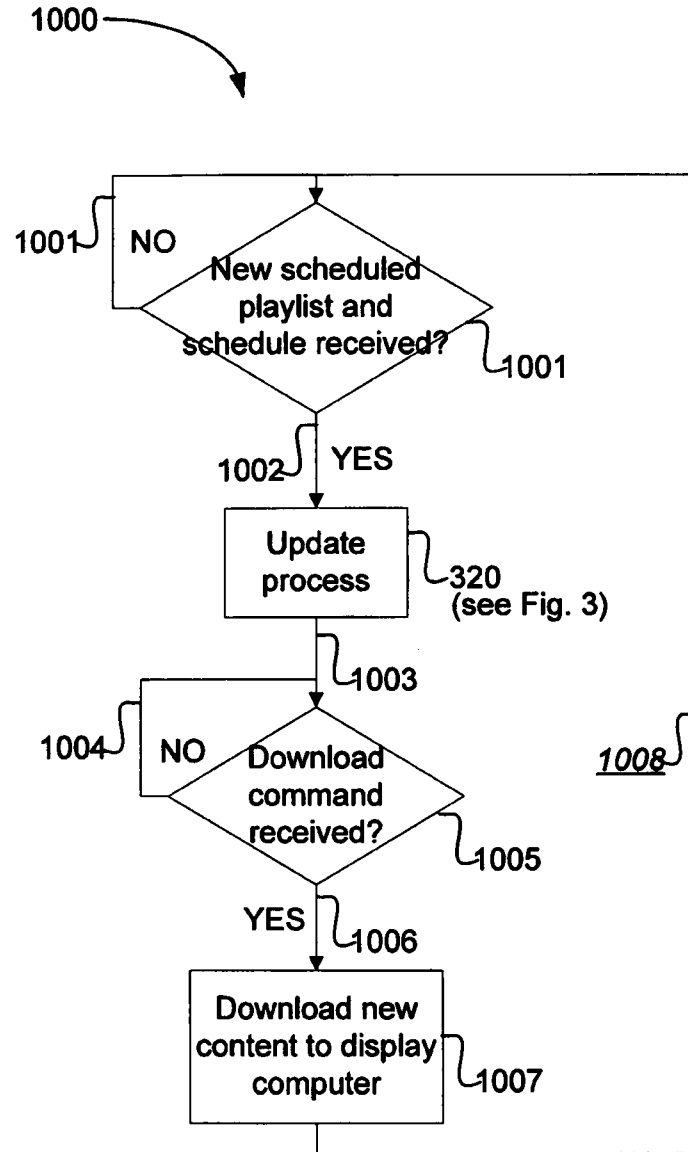


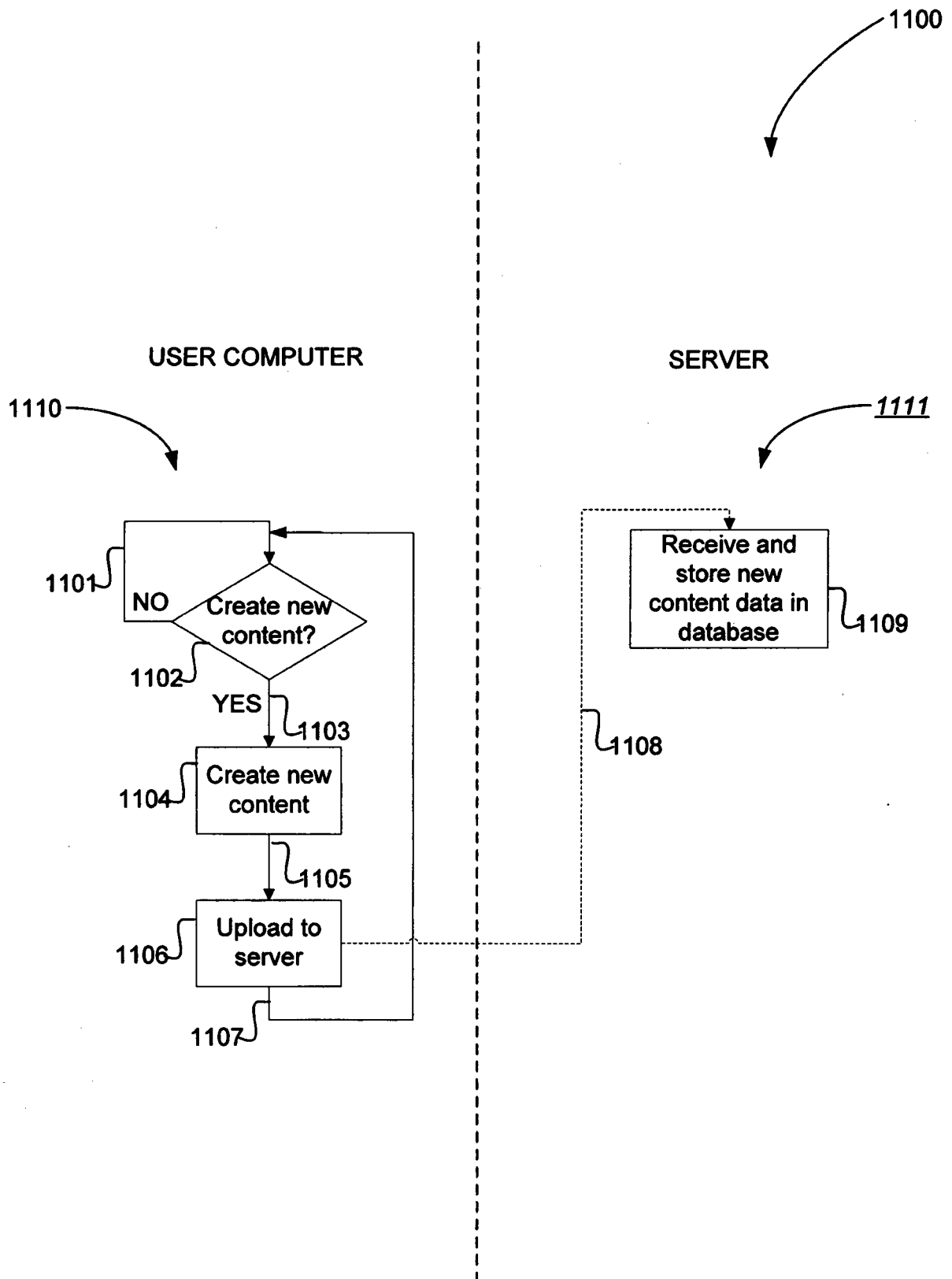
Fig. 9

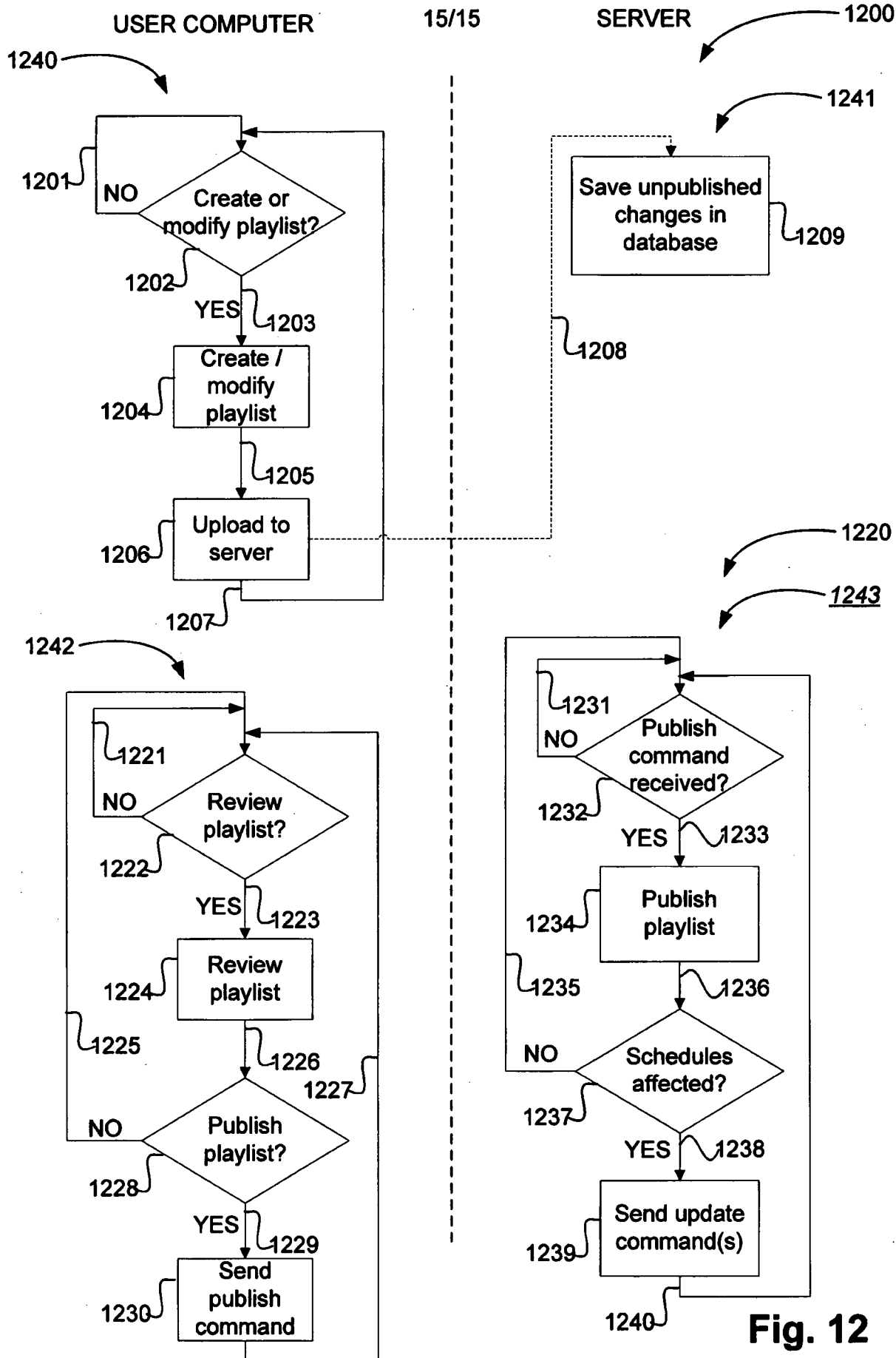
13/15

(Performed by the
server 103)

**Fig. 10**

14/15

**Fig. 11**



INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU2012/000197

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl.

G06F 17/30 (2006.01) **G06F 15/16** (2006.01) **G06Q 30/00** (2012.01)

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

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

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

WPI, Google Patents / Scholars: content, information, media, delivery, distribution, network, CDN, cache, update, new, digital /
electronic display / signage, server, and like keywords

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2010/0325303 A1 (WANG et al.) 23 December 2010 Whole document, in particular: abstract, paragraphs [0008-0010, 0013-0018, 0028, 0030, 0032-0033, 0046-0047, 0053, 0075], figures [2-5, 8]	1-19
A	US 2010/0274819 A1 (LEWIN et al.) 28 October 2010	

☐ Further documents are listed in the continuation of Box C ☒ See patent family annex

* Special categories of cited documents:	
"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E" earlier application or patent but published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search
1 May 2012

Date of mailing of the international search report
1 May 2012

Name and mailing address of the ISA/AU
AUSTRALIAN PATENT OFFICE
PO BOX 200, WODEN ACT 2606, AUSTRALIA
E-mail address: pct@ipaustalia.gov.au
Facsimile No. +61 2 6283 7999

Authorized officer
LILI ENG
AUSTRALIAN PATENT OFFICE
(ISO 9001 Quality Certified Service)
Telephone No : +61 2 6283 3138

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/AU2012/000197

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report				Patent Family Member			
US	20100325303	CA	2720085	CA	2720087	EP	2263159
		EP	2263208	US	20090259611	US	20090282159
		WO	2009126829	WO	2009126839		
US	20100274819	AU	90546/01	EP	1410215	US	20090150518
		US	7752258	WO	0217082		
Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001.							
END OF ANNEX							