

**PCT**WORLD INTELLECTUAL PROPERTY ORGANIZATION  
International Bureau

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

<b>(51) International Patent Classification <sup>6</sup> :</b> <b>G06F 13/00</b>	<b>A1</b>	<b>(11) International Publication Number:</b> <b>WO 98/50861</b> <b>(43) International Publication Date:</b> 12 November 1998 (12.11.98)
<b>(21) International Application Number:</b> PCT/US98/07474 <b>(22) International Filing Date:</b> 29 April 1998 (29.04.98) <b>(30) Priority Data:</b> 08/850,846      2 May 1997 (02.05.97)      US <b>(71) Applicant:</b> WEBTV NETWORKS, INC. [US/US]; 305 Lytton Avenue, Palo Alto, CA 94301 (US). <b>(72) Inventors:</b> PERLMAN, Stephen, G.; 721 Tiana Lane, Mountain View, CA 94041 (US). BUCHER, Tim; 305 Lytton Avenue, Palo Alto, CA 94301 (US). <b>(74) Agents:</b> NYDEGGER, Rick, D. et al.; Workman, Nydegger & Seeley, 1000 Eagle Gate Tower, 60 East South Temple, Salt Lake City, UT 84111 (US).		<b>(81) Designated States:</b> AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).  <b>Published</b> <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>
<b>(54) Title:</b> AUTOMATIC DOWNLOAD OF DEVICE DRIVERS OVER A NETWORK		
<b>(57) Abstract</b>  A method is provided of installing device drivers in a host processing system coupled to a number of peripheral devices. At system initialization, the host processing system requests the peripheral devices to which it is coupled to each provide a device code for identifying the type of the peripheral device. The host processing system receives the device codes from the peripheral devices and transmits the device codes to a remote server over a network, such as the Internet. The remote server selects an appropriate device driver for each peripheral device from a plurality of stored device drivers based on the device codes received from the host processing system and then transmits the device drivers to the host processing system over the network. This driver download process is carried out with no intervention from the user.		

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<b>AL</b>	Albania	<b>ES</b>	Spain	<b>LS</b>	Lesotho	<b>SI</b>	Slovenia
<b>AM</b>	Armenia	<b>FI</b>	Finland	<b>LT</b>	Lithuania	<b>SK</b>	Slovakia
<b>AT</b>	Austria	<b>FR</b>	France	<b>LU</b>	Luxembourg	<b>SN</b>	Senegal
<b>AU</b>	Australia	<b>GA</b>	Gabon	<b>LV</b>	Latvia	<b>SZ</b>	Swaziland
<b>AZ</b>	Azerbaijan	<b>GB</b>	United Kingdom	<b>MC</b>	Monaco	<b>TD</b>	Chad
<b>BA</b>	Bosnia and Herzegovina	<b>GE</b>	Georgia	<b>MD</b>	Republic of Moldova	<b>TG</b>	Togo
<b>BB</b>	Barbados	<b>GH</b>	Ghana	<b>MG</b>	Madagascar	<b>TJ</b>	Tajikistan
<b>BE</b>	Belgium	<b>GN</b>	Guinea	<b>MK</b>	The former Yugoslav Republic of Macedonia	<b>TM</b>	Turkmenistan
<b>BF</b>	Burkina Faso	<b>GR</b>	Greece			<b>TR</b>	Turkey
<b>BG</b>	Bulgaria	<b>HU</b>	Hungary	<b>ML</b>	Mali	<b>TT</b>	Trinidad and Tobago
<b>BJ</b>	Benin	<b>IE</b>	Ireland	<b>MN</b>	Mongolia	<b>UA</b>	Ukraine
<b>BR</b>	Brazil	<b>IL</b>	Israel	<b>MR</b>	Mauritania	<b>UG</b>	Uganda
<b>BY</b>	Belarus	<b>IS</b>	Iceland	<b>MW</b>	Malawi	<b>US</b>	United States of America
<b>CA</b>	Canada	<b>IT</b>	Italy	<b>MX</b>	Mexico	<b>UZ</b>	Uzbekistan
<b>CF</b>	Central African Republic	<b>JP</b>	Japan	<b>NE</b>	Niger	<b>VN</b>	Viet Nam
<b>CG</b>	Congo	<b>KE</b>	Kenya	<b>NL</b>	Netherlands	<b>YU</b>	Yugoslavia
<b>CH</b>	Switzerland	<b>KG</b>	Kyrgyzstan	<b>NO</b>	Norway	<b>ZW</b>	Zimbabwe
<b>CI</b>	Côte d'Ivoire	<b>KP</b>	Democratic People's Republic of Korea	<b>NZ</b>	New Zealand		
<b>CM</b>	Cameroon			<b>PL</b>	Poland		
<b>CN</b>	China	<b>KR</b>	Republic of Korea	<b>PT</b>	Portugal		
<b>CU</b>	Cuba	<b>KZ</b>	Kazakstan	<b>RO</b>	Romania		
<b>CZ</b>	Czech Republic	<b>LC</b>	Saint Lucia	<b>RU</b>	Russian Federation		
<b>DE</b>	Germany	<b>LI</b>	Liechtenstein	<b>SD</b>	Sudan		
<b>DK</b>	Denmark	<b>LK</b>	Sri Lanka	<b>SE</b>	Sweden		
<b>EE</b>	Estonia	<b>LR</b>	Liberia	<b>SG</b>	Singapore		

## **AUTOMATIC DOWNLOAD OF DEVICE DRIVERS OVER A NETWORK**

### **BACKGROUND OF THE INVENTION**

Modern personal computer systems are often characterized by a main computer,  
5 or host device, to which one or more peripheral devices may be coupled. Typical  
peripheral devices that may be used in such a system include keyboards, pointing devices,  
monitors, printers, mass storage devices, and audio or visual input/output devices.  
Generally, for any particular peripheral device, the main computer must be programmed  
10 with special software that permits the main computer to communicate with the peripheral  
device--this software is often referred to as the device driver for the peripheral device.

Many computer systems rely upon the user to manually load the appropriate  
device drivers whenever a new peripheral device is connected to the main processing  
system. This approach places an undesirable burden upon the user. In other cases, a  
computer system might be sold to the user with a number of device drivers already  
15 installed. Some of those drivers may simply reside in memory without being used if the  
corresponding peripheral devices are not purchased or are simply not connected by the  
user, thus wasting valuable memory space. This approach becomes especially undesirable  
as the number of different peripheral devices which could potentially be used with the  
computer system increases. A third approach which avoids this type of waste is to pre-  
20 load the appropriate driver into a non-volatile memory within the peripheral device; the  
main computer could then load the appropriate driver from the peripheral device when the  
peripheral device is connected to the system. This approach has the disadvantage,  
however, of requiring additional memory space to be provided in the peripheral device,  
which adds to the cost of the device.

25 Therefore, it is desirable to provide a technique by which the appropriate drivers  
for all peripheral devices connected to a computer system are automatically installed in  
the computer system. It is further desirable that only those device drivers for peripheral  
devices that are actually connected are so installed, so as to avoid unnecessary  
consumption of memory in the computer system. It is further desirable that such a  
30 technique avoid storing device drivers on individual peripheral devices.

### **SUMMARY OF THE INVENTION**

The present invention includes a method of installing a device driver in a host  
processing device coupled to a peripheral device. In the method, a device code is received  
35 from the peripheral device. The device code is then transmitted to a remote processing

system. An appropriate device driver is then received from the remote processing system in response to the transmission of the device code to the remote processing system.

In another aspect of the present invention, the remote processing system receives the device code from the host processing device and uses the device code to select an appropriate device driver for the peripheral device from a number of stored device drivers. In particular embodiments, communication between the host processing device and the remote processing system is implemented over a network, such as the Internet.

Other features of the present invention will be apparent from the accompanying drawings and from the detailed description which follows.

### BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

Figure 1 illustrates a WebTV client system connected to a WebTV server system over the Internet.

Figure 2 illustrates a WebTV client system.

Figure 3 illustrates the internal features of a WebTV box in block diagram form.

Figure 4 illustrates the internal features of a WebTV server in block diagram form.

Figure 5 illustrates a WebTV box connected to a number of peripheral devices in a daisy chain.

Figure 6 is a flow diagram illustrating a routine for automatically downloading from a remote server the appropriate drivers for connected peripheral devices.

### DETAILED DESCRIPTION

A method and apparatus are described for automatically installing appropriate device drivers for all peripheral devices connected to a host processing system over a network. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be evident, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate description of the present invention.

The present invention includes steps which, in one embodiment, are embodied in machine-executable software instructions, such that the present invention is carried out in a processing system as a result of a processor executing the instructions. In other

embodiments, hardwired circuitry may be used in place of, or in combination with, software instructions to implement the present invention.

In one embodiment, the present invention is included in a system known as WebTV™ (hereinafter "WebTV"), which uses a standard television set as a display device for browsing the World Wide Web ("the Web") and which connects to the Internet using a standard telephone, ISDN (Integrated Services Digital Network), or other similar communication path. A user of a WebTV client system can access, via the Internet (generally), WebTV Network services provided by one or more remote WebTV servers. The WebTV Network services are used in conjunction with software running in a WebTV client system to enable the user to browse the Web, send electronic mail, and to make use of the Internet in various other ways.

Although the present invention is described herein as implemented in the WebTV system for illustrative purposes, the present invention can also be implemented in other contexts, such as in a conventional personal computer (PC), workstation, etc.

## **I. SYSTEM OVERVIEW**

Figure 1 illustrates a configuration of a WebTV network according to one embodiment. A WebTV client 1 is coupled to a modem pool 2 via a direct-dial, bi-directional data connection 29, which may be a conventional telephone, i.e., "plain old telephone service" (POTS) or ISDN link, or any other suitable type of data connection. The modem pool 2 may be provided by a local Internet Service Provider (ISP). The modem pool 2 is coupled typically through a router, such as that conventionally known in the art, to at least one remote Web server (hereinafter WWW server) 4 via a conventional network infrastructure 3, such as the Internet. The WebTV system also includes a WebTV server 5, which implements WebTV Network services and specifically supports the WebTV client 1. The WebTV server 5 generally includes one or more conventional computer systems. The server 5 may actually comprise multiple physical and logical devices connected in a distributed architecture. The client 1 can connect to the server 5 either through direct data connection (i.e., telephone, ISDN, etc.) or through the Internet 3 via the modem pool 2. Note that the modem pool 2 is a conventional modem pool, such as those found today providing access throughout the world to the Internet and private networks.

### **A. WEBTV CLIENT SYSTEM**

Figure 2 illustrates a WebTV client system 1 according to one embodiment. The client system 1 includes an electronics unit 10 (hereinafter "the WebTV box 10" or "the

box 10"), an ordinary television set 12, and a hand-held remote control 11. In an alternative embodiment (not shown), the WebTV box 10 is built into the television set 12 as an integral unit. The box 10 includes hardware and software for providing the user with a graphical user interface by which the user can browse the Web, send e-mail, and access other Internet services. The client system 1 uses the television set 12 as both a display device and an audio output device. The box 10 is coupled to the television set 12 by a link 6. The link 6 includes an audio channel for generating sound from the television's speaker and an RF (radio frequency), S-video, composite video, or other equivalent form of video channel. The data link 29 between the box 10 and the WebTV server 5 can be a conventional telephone (POTS, i.e., Plain Old Telephone Service), ISDN, or other suitable data connection. The box 10 receives alternating-current (AC) power through a conventional AC power line 7. Remote control 11 is operated by the user in order to control the client system 1 to browse the Web and perform other functions. The box 10 receives commands from remote control 11 via an infrared (IR) communication link. In alternative embodiments, the link between the remote control 11 and the box 10 may be an RF link or any other suitable type of link.

Figure 3 illustrates the internal components of the WebTV box 10, according to one embodiment. Operation of the client system 1 is controlled by a central processing unit (CPU) 21, which is coupled to a bus 29. It will be recognized that functions described herein as being performed by the WebTV box 10 may result from the CPU 21 of the WebTV box 10 executing software instructions, from operation of hardwired circuitry, or both. It will also be recognized that bus 29 may represent multiple physical buses connected by various bridges and/or adapters, the details of which are not required for an understanding of the present invention.

Also coupled to the bus 29 are an audio digital-to-analog converter (DAC) 25, which provides audio output to the television 12, and a video encoder 26, which provides video output to the television set 12. The outputs of the audio DAC 25 and the video encoder 26 are combined by conventional circuitry (not shown) to produce the link 6 to the television 12. An IR interface 24 coupled to the bus 29 detects IR signals transmitted by remote control 11 and, in response, provides corresponding electrical signals onto the bus 29. A communication device 27 is coupled to the bus 29 to provide bi-directional data communication with the remote WebTV server 5 and WWW servers 4. The communication device 27 may be a conventional telephone modem, an ISDN adapter, or any other suitable data communication device. A read-only memory (ROM) 22 and a random access memory (RAM) 23 are also coupled to the bus 29. The ROM 22 may

include programmable-erasable memory, such as electrically-erasable programmable ROM (EEPROM) or flash memory.

Also coupled to the bus 29 is an expansion bus 28. The expansion bus 28 is a high-speed, multiplexed (address and data), parallel bus that is used to connect various peripheral devices to the WebTV box 10; the WebTV box 10 thereby functions as the host device of a processing system. Peripheral devices which may be connected to the WebTV box 10 using expansion bus 28 include devices such as printers, mass storage devices (e.g., CD-ROM, digital versatile disk (DVD), magnetic disk, etc.), microphones, video cameras, video tuners, and additional data communication devices. It will be recognized that various adapters may be used to interface the expansion bus 28 with bus 29, the details of which are not required for an understanding of the present invention.

## **B. WEBTV SERVER SYSTEM**

Certain steps of the present invention may be performed by the server 5. As in the case of the client 1, such steps generally may be implemented using software, hardwired circuitry, or a combination thereof. The WebTV server 5 generally includes one or more computer systems having the architecture illustrated in Figure 4. Note, however, that the illustrated architecture is only one possible architecture; a WebTV server 5 is not restricted to such an architecture for purposes of practicing the present invention. The illustrated architecture includes a CPU 50, ROM 51, RAM 52, a mass storage facility 53, a communication device 55, and various input/output (I/O) devices 56. It will be recognized that functions described herein as being performed by the WebTV server 5 may result from the CPU 50 of the WebTV server 5 executing software instructions, from operation of hardwired circuitry, or both.

ROM 51 may include programmable and/or programmable-erasable storage. Mass storage facility 53 may include a magnetic, optical, magneto-optical, or other equivalent mass storage medium. It should be noted that the WebTV server system 5 includes substantially more data storage capacity than the WebTV client system 1. I/O devices 56 may include a monitor, a keyboard, a pointing device, and other similar devices, for example. Communication device 55 may be a conventional telephone (POTS) modem, an ISDN adapter, or any other suitable data communication device. Communication device 55 may be used to communicate data between the server 5 and the remote WWW server 4.

Note that the WebTV server 5 may actually comprise multiple devices connected in a distributed architecture; accordingly, communication device 55 may be used to communicate data between the server 5 and other physical devices used to implement

WebTV server functions. The WebTV server 5 may also include a network interface card (NIC) or other similar device for communicating with other physical devices that provide WebTV server functions.

## II. CONNECTION OF PERIPHERAL DEVICES

Figure 5 illustrates a WebTV client system 1a according to an embodiment of the present invention in which the WebTV box 10 is connected to a number of peripheral devices 30 on the expansion bus 28 in a daisy chain configuration. The peripheral devices 30 may include keyboards, pointing devices, monitors, printers, mass storage devices, and audio or visual input/output devices, for example. It is useful at this point to define certain terminology to be used in this description relating to the topological position (i.e., slot position) of devices connected in the daisy chain. For purposes of this description, the phrase "more senior in the daisy chain" refers to devices that are topologically closer to the WebTV box 10 in the daisy chain than a given device, while the phrase "more junior in the daisy chain" refers to devices that are topologically farther from the WebTV box 10 in the daisy chain than a given device. Thus, the greater the number of devices that are connected in the daisy chain between a given device and the WebTV box 10, the "more junior" in the daisy chain the given device is considered to be; the fewer the number of devices that are connected in the daisy chain between a given device and the WebTV box 10, the "more senior" in the daisy chain the given device is considered to be.

### A. TOPOLOGY DETERMINATION

At system initialization (i.e., upon power-up or in response to a reset command), the WebTV box 10 determines the topological configuration of the processing system. That is, the WebTV box 10 initiates the process of assigning a slot position to each peripheral device and determines the number of peripheral devices on the expansion bus 28. Generally, in this process, each peripheral device 30 in the daisy chain receives the slot number of the next more senior device in the daisy chain and assigns itself that slot number incremented by one. This action occurs by the WebTV box 10 initially assigning itself a slot number of zero and then transmitting that slot number onto the bus 28 to the most senior peripheral device 30 in the daisy chain. The most senior peripheral device 30 in the daisy chain receives slot number zero from the WebTV box 10 and accordingly assigns itself a slot number of one. That peripheral device then passes its slot number on to the next most senior device in the daisy chain, and this process repeats for each peripheral device 30 until all peripheral devices 30 have been assigned a unique slot number. In response to a query from the WebTV box 10, each peripheral device 30 then



transmits its slot number onto the bus 28 back to the WebTV box 10. Based on the responses to the queries, the WebTV box 10 determines the number of devices on the bus 28 and the slot positions of the devices.

## **B. INSTALLATION OF DEVICE DRIVERS**

For each of the peripheral devices 30 in the daisy chain, an appropriate device driver must be stored somewhere within the processing system. However, as mentioned above, it is not desirable to store such drivers within the peripheral devices themselves, nor is it desirable to pre-load into the WebTV box 10 a device driver for every possible peripheral device which might be connected to the expansion bus 28. Further, it is not desirable to require a user of the client system 1a to manually load the appropriate device drivers. Consequently, the present invention includes a technique for automatically installing in the WebTV client system 1a only the appropriate device drivers for the particular system configuration being used, as will now be described.

Each peripheral device that is designed for use with a WebTV client system is provided with a device code that is used to precisely identify the peripheral device. For example, this code may identify the type of device (e.g., printer), the manufacturer, and the model number. In one embodiment, the device code is a 64-bit code stored within a register in the peripheral device. In accordance with the present invention, these device codes are used by the WebTV box 10 to request and receive only the appropriate drivers for the peripheral devices 30 that are actually connected in the system 1a, as will now be described with reference to Figure 6.

Figure 6 illustrates a routine for installing the appropriate device drivers in client system 1a. In response to system initialization, the WebTV box 10 requests the device codes from all peripheral devices 30 connected in the daisy chain in step 601. Each peripheral device 30 that is designed to be used in a WebTV client system is designed to respond to such a request by transmitting its device code and slot number onto the bus 28 to the next most senior device in the daisy chain. Each peripheral device relays responses received from more junior devices in the daisy chain toward the WebTV box 10. Hence, in step 602, the WebTV box 10 receives the device codes from all connected peripheral devices 30. In step 603, the WebTV box 10 transmits all of the received device codes to the WebTV server 5 over the network connection 29.

Note that although Figure 6 illustrates an embodiment in which all device codes are transmitted to the server 5 in one step and then all device drivers are downloaded in one step, such a sequence is not required to practice the present invention. For example,

in an alternative embodiment, device codes and device drivers may be transmitted and downloaded (respectively) one at a time, i.e., on a piecemeal basis.

Stored in the WebTV server 5 are the device drivers for all peripheral devices that may be used in a WebTV client system and a database including the device codes of all such peripheral devices. The database is set up to allow device codes to be associated with the appropriate stored device drivers. Note that because the WebTV server system 5 provides much larger storage capacity than a WebTV client system does, it is practical to store the device drivers in the WebTV server system 5. Note also that, using this technique, the device drivers only need to be stored once, rather than storing them in every WebTV client system.

Hence, in step 604, the WebTV server 5 receives the device codes that were transmitted by the WebTV client 1 over the network. In step 605, in response to receiving the device codes, the WebTV server 5 automatically uses the received codes to reference the database to determine the appropriate device drivers for the particular peripheral devices 30 connected to the WebTV box 10. Once the appropriate drivers are identified, in step 606 the WebTV server 5 automatically downloads these device drivers to the WebTV client 1 over the network connection 29. In step 607, the WebTV box 10 receives and automatically installs the device drivers transmitted from the WebTV server 5.

Hence, the present invention provides a technique for installing in a main processing device (e.g., the WebTV box 10) only the appropriate drivers for the particular system configuration being used, in a manner that is automatic and transparent to the user. The technique provides the additional advantages of reducing consumption of memory in the main processing device with unused code while avoiding the requirement of providing additional memory in individual peripheral devices.

Thus, a method and apparatus have been described for automatically installing appropriate device drivers for all peripheral devices connected to a host processing system over a network. Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention as set forth in the claims. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A method of installing a device driver in a host processing device coupled to a peripheral device, the method comprising the steps of:

receiving a device code from the peripheral device;

transmitting the device code to a remote processing system; and

receiving the device driver from the remote processing system in response to transmitting the device code to the remote processing system.

2. A method according to claim 1, wherein the device code is for indicating a device type of the peripheral device.

3. A method according to claim 1, wherein the step of transmitting the device code to the remote processing device comprises the step of transmitting the device code to the remote processing device over a network.

4. A method according to claim 3, wherein the network comprises a wide area network.

5. A method according to claim 3, wherein the network comprises the Internet.

6. A method according to claim 1, wherein the step of receiving the device driver from the remote processing system comprises the step of receiving the device driver from the remote processing system via a network.

7. A method according to claim 6, wherein the network comprises a wide area network.

8. A method according to claim 6, wherein the network comprises the Internet.

9. A method of installing device drivers in a host processing device coupled to at least one peripheral device, each peripheral device having an associated device driver for enabling the host processing device to communicate with said peripheral device, each peripheral device having a device code stored therein for indicating a device type of the peripheral device, the method comprising the steps of:

in response to an initialization of the host processing device, requesting the device code of each peripheral device;

receiving the device code of each peripheral device;

transmitting each received device code to a remote server via a network;

and

receiving a responsive transmission from the remote server via the network, the responsive transmission including an appropriate device driver for each of said at least one peripheral devices.

10. A method according to claim 9, wherein the network comprises the Internet.

11. A machine-implemented method of installing a device driver in a remote host processing device coupled to a peripheral device, the method comprising the steps of:

receiving a device code of the peripheral device from the remote host processing device;

automatically identifying the device driver for the peripheral device based on the device code; and

causing the device driver for the peripheral device to be transmitted to the remote host processing system.

12. A method according to claim 11, wherein the device code received from the remote processing system originates from, and is stored in, the peripheral device, wherein the device code is for indicating a device type of the peripheral device.

13. A method according to claim 11, wherein the step of receiving the device code of the peripheral device comprises the step of receiving the device code of the peripheral device over a network.

14. A method according to claim 13, wherein the network comprises a wide area network.

15. A method according to claim 13, wherein the network comprises the Internet.

16. A method according to claim 11, wherein the step of causing the device driver to be transmitted to the remote host processing system comprises the step of causing the device driver to be transmitted to the remote host processing system over a network.

17. A method according to claim 16, wherein the network comprises a wide area network.

18. A method according to claim 16, wherein the network comprises the Internet.

19. A method of installing a device driver in a host processing device coupled to at least one peripheral device, each peripheral device having an associated device driver for allowing the host processing device to communicate with said peripheral device, each peripheral device having a device code stored therein for indicating a device type of the peripheral device, the method comprising the steps of:

receiving from the host processing device via a network the device code of each peripheral device coupled to the host processing system;

automatically identifying, based on the received device codes, a device driver for each peripheral device coupled to the host processing system;

automatically retrieving each identified device driver from a storage device; and

5 automatically causing each identified device driver to be transmitted to the host processing system via the network.

20. A method according to claim 19, wherein the network comprises the Internet.

10 21. A processing system capable of being coupled to a peripheral device, the processing system comprising:

a processor; and

a memory coupled to the processor, the memory having instructions stored therein which, when executed by the processor, cause the processing system to:

15 request a device code from the peripheral device, the device code for indicating a device type of the peripheral device;

receive the device code from the peripheral device;

transmit the device code to a remote processing system; and

receive the device driver from the remote processing system in response to transmitting the device code.

20 22. A processing system according to claim 21, wherein the instructions which, when executed, cause the processing system to transmit the device code to the remote processing system comprise instructions which, when executed, cause the processing system to transmit the device code to the remote processing system over a network.

25 23. A processing system according to claim 22, wherein the network comprises a wide area network.

24. A processing system according to claim 22, wherein the network comprises the Internet.

30 25. A processing system according to claim 22, wherein the instructions which, when executed, cause the processing system to receive the device driver from the remote processing system comprise instructions which, when executed, cause the processing system to receive the device driver from the remote processing system over the network.

26. A processing system according to claim 25, wherein the network comprises a wide area network.

35 27. A processing system according to claim 25, wherein the network comprises the Internet.

28. A server processing system in communication with a remote client system, the server processing system comprising:

a processor;

a first memory unit coupled to the processor, the first memory unit having a plurality of device drivers stored therein; and

a second memory unit coupled to the processor, the second memory having instructions stored therein which, when executed by the processor, cause the server processing system to:

receive a device code from the remote client processing system, the device code indicating a device type of a peripheral device coupled to the remote client processing system;

select one of the plurality of device drivers for the peripheral device based on the device code; and

cause the selected device driver to be transmitted to the remote client processing system.

29. A server processing system according to claim 28, wherein the instructions which, when executed, cause the server processing system to cause the selected device driver to be transmitted to the remote client processing system comprise instructions which, when executed, cause the server processing system to cause the selected device driver to be transmitted to the remote client processing system over a network.

30. A server processing system according to claim 29, wherein the network comprises a wide area network.

31. A server processing system according to claim 29, wherein the network comprises the Internet.

32. A server processing system according to claim 29, wherein the instructions which, when executed, cause the server processing system to receive the device code from the remote client processing system comprise instructions which, when executed, cause the server processing system to receive the device code from the remote client processing system over the network.

33. A server processing system according to claim 32, wherein the network comprises a wide area network.

34. A server processing system according to claim 32, wherein the network comprises the Internet.

35. A local processing system in communication with a remote processing system, the remote processing system coupled to a peripheral device, the local processing system comprising:

means for storing a plurality of selectable device drivers;

means for receiving from the host processing device, via a network, a device code of the peripheral device;

means for selecting one of the plurality of selectable device drivers as an appropriate device driver for the peripheral device based on the device code; and

means for causing the appropriate device driver for the peripheral device to be transmitted to the host processing system via the network.

36. A local processing system according to claim 35, wherein the device code indicates a device type of the peripheral device.

37. A local processing system according to claim 36, wherein the device code is received from the remote processing system after having been received by the remote processing system from the peripheral device.

38. A local processing system according to claim 35, wherein the network comprises a wide area network.

39. A local processing system according to claim 35, wherein the network comprises the Internet.

1 / 6

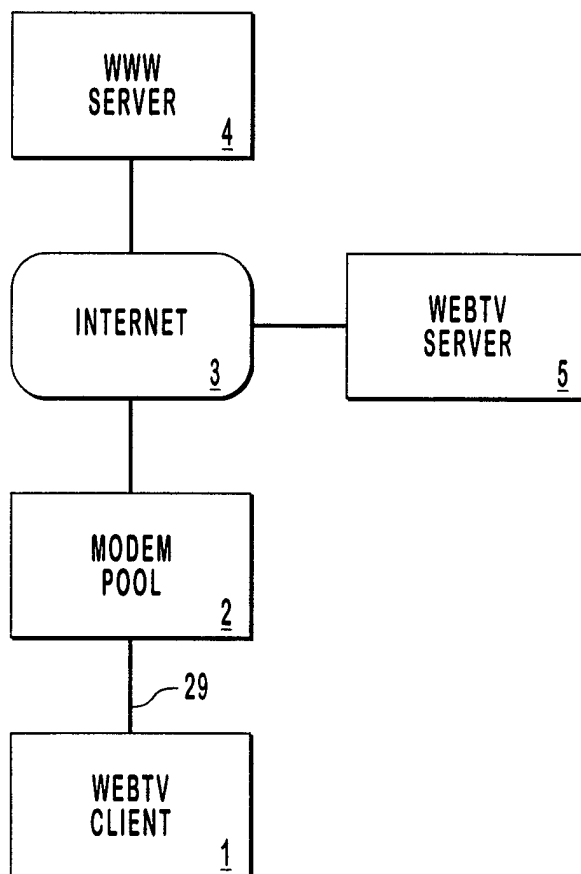


FIG. 1



2 / 6

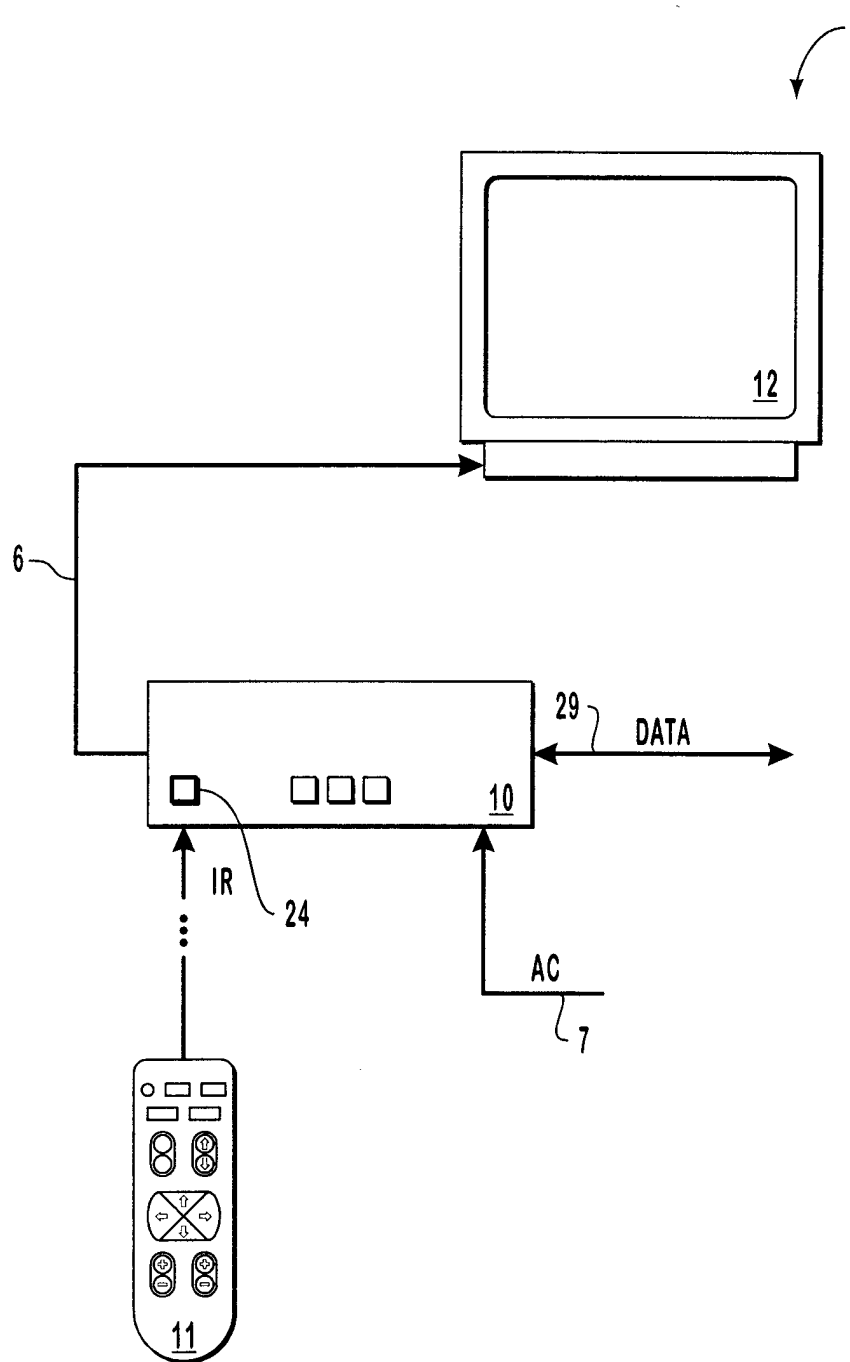


FIG. 2

3 / 6

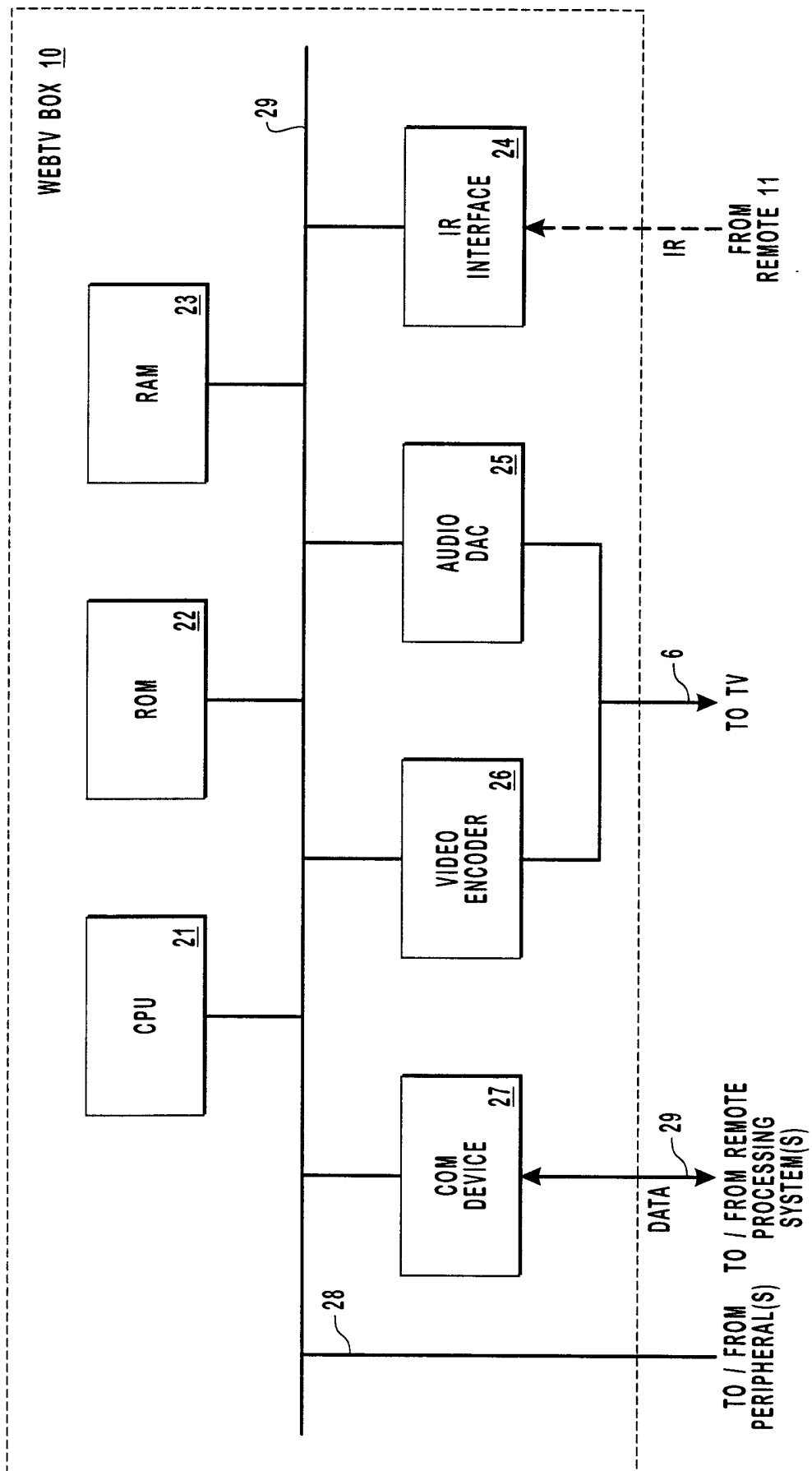


FIG. 3

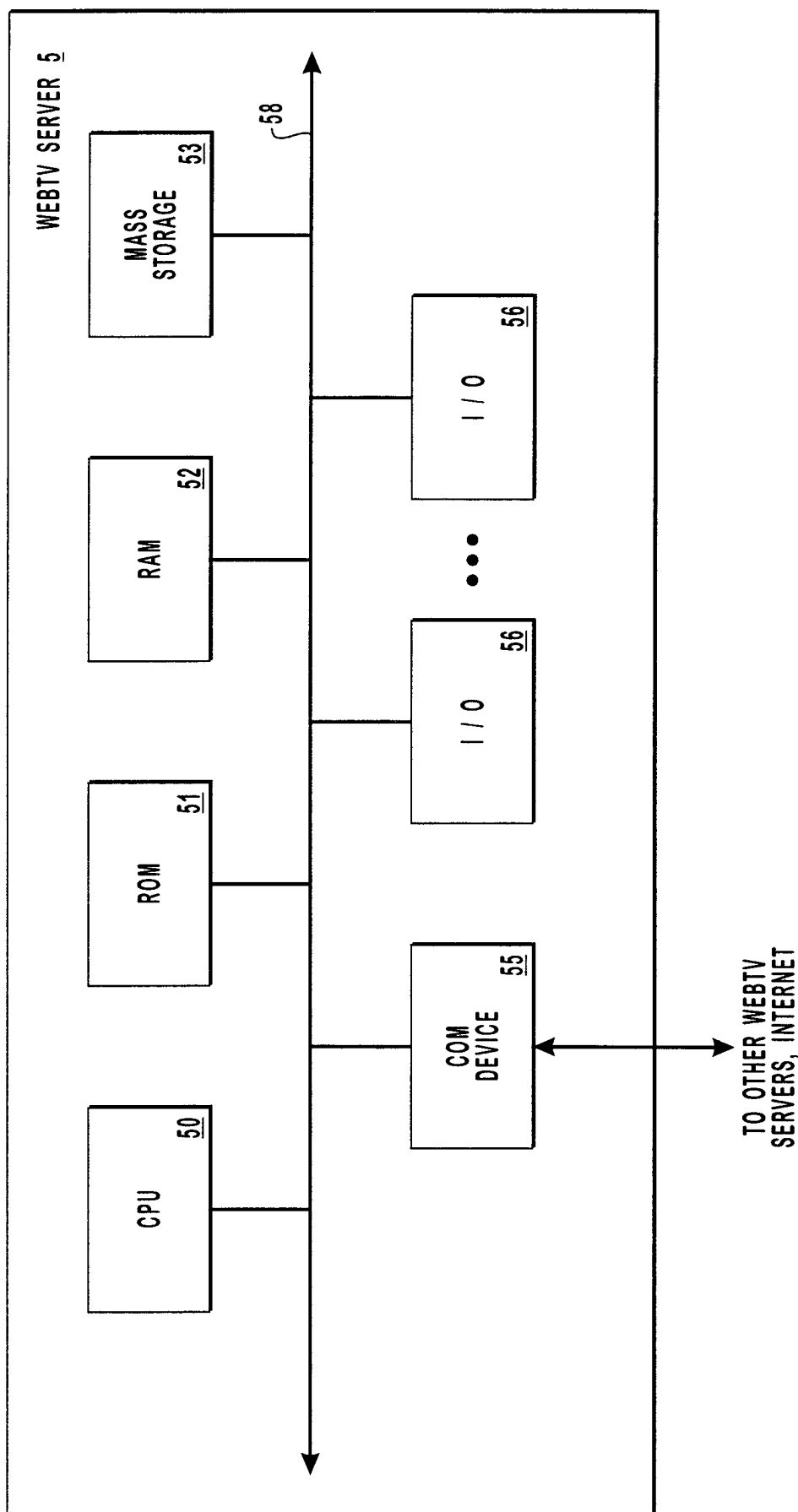


FIG. 4

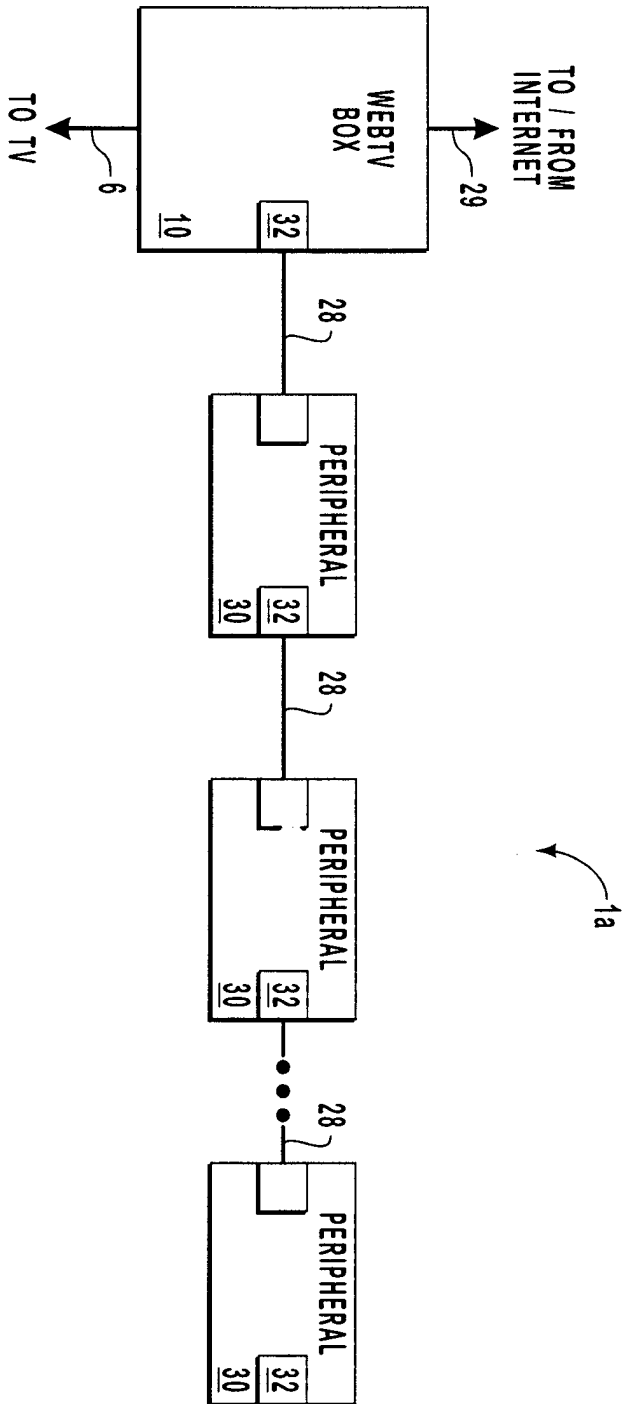


FIG. 5

6 / 6

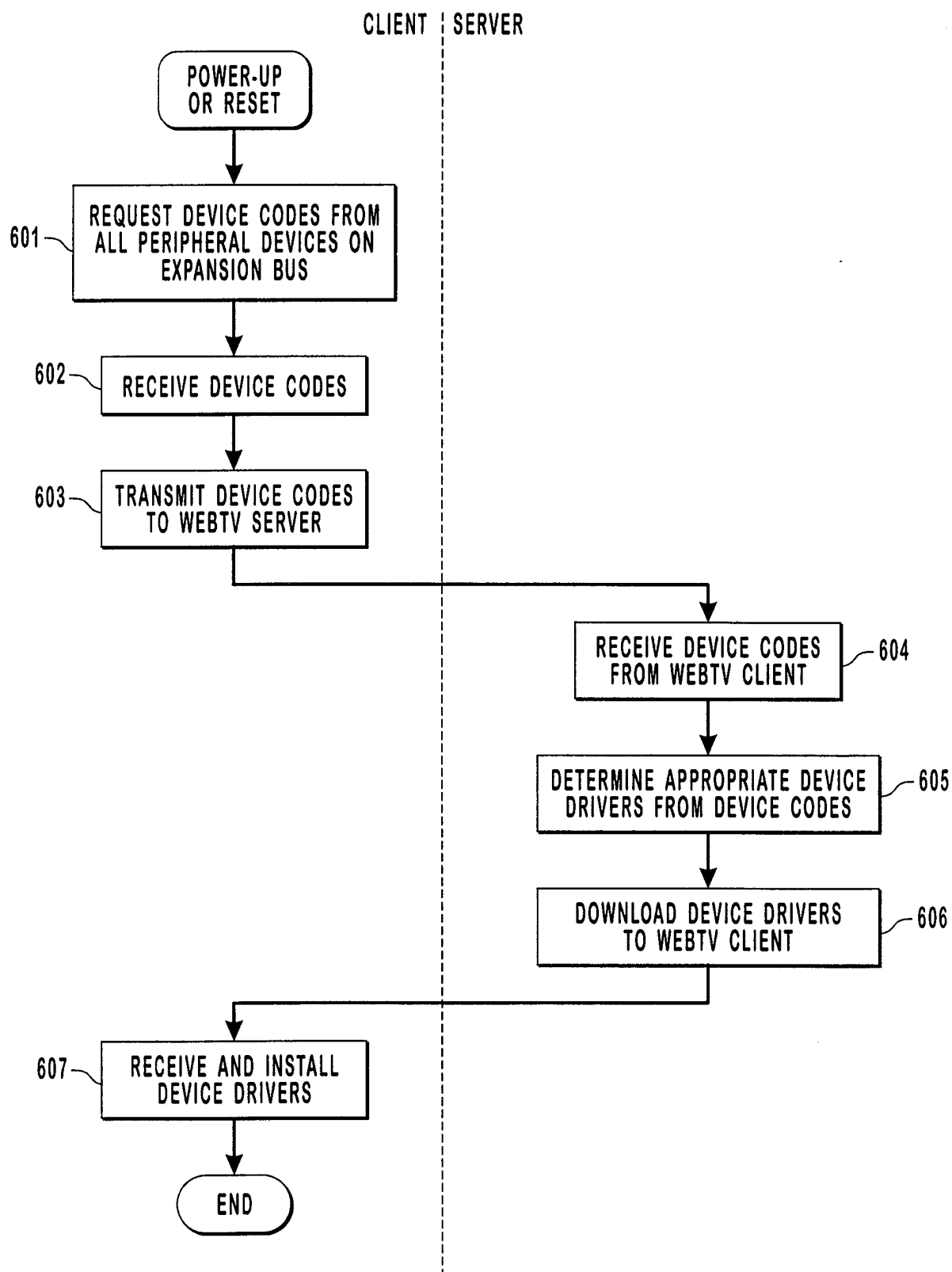


FIG. 6

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/US98/07474

**A. CLASSIFICATION OF SUBJECT MATTER**

IPC(6) :G06F 13/00

US CL :395/680, 681, 685, 652, 712, 828

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)

U.S. : 395/680, 681, 685, 652, 712, 828

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)

APS

search terms: client, server, device driver

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,655,148 A [RICHTMAN et al] 05 AUG 1997, col. 18, lines 59-60, col. 45, lines 10-51, col. 16, lines 36-39, col. 44, lines 65-67, col. 48, lines 13-17 and col. 48, lines 48-65.	1-39

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

* Special categories of cited documents:	"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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier document published on or after the international filing date	"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
"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)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

16 JULY 1998

Date of mailing of the international search report

10 SEP 1998

Name and mailing address of the ISA/US  
Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231

Facsimile No. (703) 308-5357

Authorized officer

TODD VOELTZ

Telephone No. (703) 308-9293