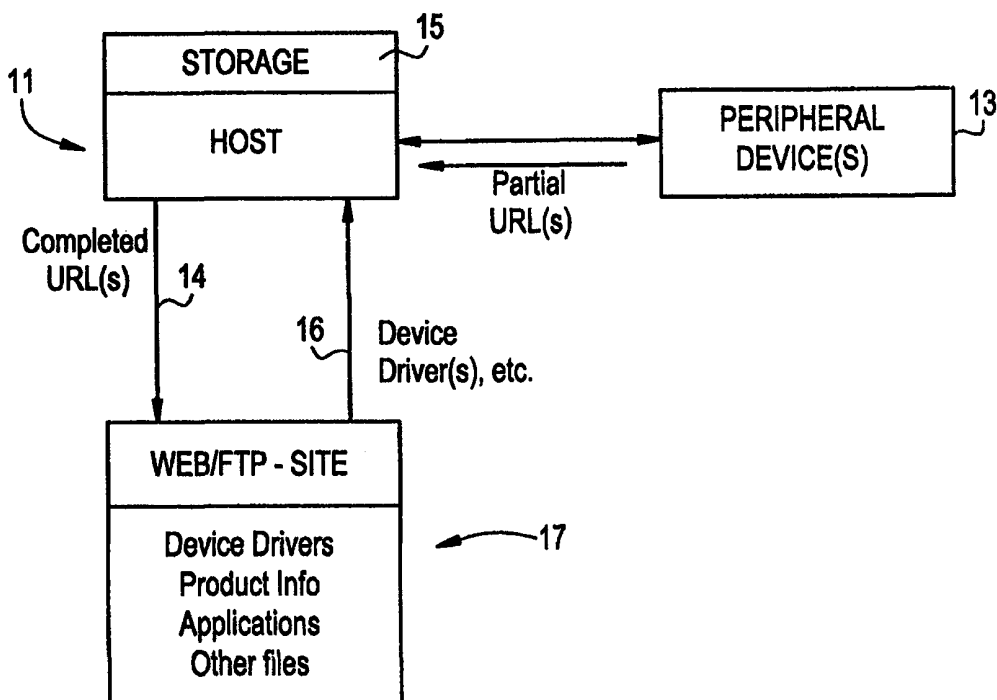




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<b>(51) International Patent Classification <sup>7</sup> :</b> <b>G06F 13/15, 9/445</b>	<b>A2</b>	<b>(11) International Publication Number:</b> <b>WO 00/33232</b> <b>(43) International Publication Date:</b> 8 June 2000 (08.06.00)
<b>(21) International Application Number:</b> PCT/SE99/02160 <b>(22) International Filing Date:</b> 23 November 1999 (23.11.99) <b>(30) Priority Data:</b> 60/110,226                      30 November 1998 (30.11.98)    US 09/388,499                      1 September 1999 (01.09.99)        US <b>(71) Applicant:</b> TELEFONAKTIEBOLAGET LM ERICSSON (publ) [SE/SE]; S-126 25 Stockholm (SE). <b>(72) Inventor:</b> ELG, Johannes; Valborgsgatan 43, S-216 13 Malmö (SE). <b>(74) Agent:</b> ERICSSON MOBILE COMMUNICATIONS AB; IPR Department, S-221 83 Lund (SE).		<b>(81) Designated States:</b> AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, 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, GW, ML, MR, NE, SN, TD, TG).  <b>Published</b> <i>Without international search report and to be republished upon receipt of that report.</i>

(54) Title: IMPROVEMENTS IN HOST COMPUTER ACCESS TO PERIPHERAL DEVICE DRIVERS



## (57) Abstract

A peripheral device (13) is cooperable with a host computer (11) to direct the host computer (11) to device information stored at an external location (17) accessible to the host computer (11) via an external data path (14, 16). The device information can then be automatically downloaded to the host computer (11) for use in conjunction with the peripheral device (13).

**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.

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

-1-

## IMPROVEMENTS IN HOST COMPUTER ACCESS TO PERIPHERAL DEVICE DRIVERS

This application claims the priority under 35 USC 119(e)(1) of copending U.S.  
5 Provisional Application No. 60/110,226, filed on November 30, 1998.

### FIELD OF THE INVENTION

The invention relates generally to interfacing between host computers and  
peripheral devices and, more particularly, to providing host computer access to  
10 peripheral device drivers.

### BACKGROUND OF THE INVENTION

Conventionally, when a host computer such as a personal computer,  
engineering workstation, etc. is connected to a peripheral device such as a printer,  
15 monitor, or even a PCI or EISA card, the peripheral device identifies itself uniquely  
to the host computer so the host computer can then search for a device driver for use  
with that peripheral device. The term "device driver" conventionally refers to software  
that must be installed in the host computer in order to permit the host computer to use  
the features of the device. The host computer can use the unique identification  
20 received from the peripheral device to search for the appropriate driver on, e.g., a  
floppy disk, CD-ROM or within the operating system utilized by the host computer  
platform. Although the peripheral device uniquely identifies itself to the host  
computer, the peripheral device does not tell the host computer where to find the  
driver. The host's operating system needs some independent knowledge of suitable  
25 places to search for the driver. After the device driver is located and retrieved, the host  
computer can use the device driver to access the features of the peripheral device.

The aforementioned conventional technique of providing host computer access  
to a device driver associated with a connected peripheral device is often referred to as  
Plug-and-Play (PnP). Plug-and-Play assumes a specific driver for each device.  
30 Because drivers are by nature operation system-specific, Plug-and-Play does not

-2-

promote interoperability. Although Plug-and-Play is typically adequate for use with peripheral devices such as EISA and PCI cards, Plug-and-Play's lack of interoperability presents a problem when, for example, the peripheral device is a wireless device such as a mobile telephone, that can move from one host computer to another and can connect to virtually any type of host computer, for example personal computers, personal digital assistants and Smartphones.

The conventional Plug-and-Play technique is not suitable for use with such highly mobile peripheral devices, because each of the various host computer platforms to which the mobile peripheral device can be connected must be able to locate and retrieve the device driver associated with the mobile peripheral device. Conventional Plug-and-Play typically assumes that the peripheral device will be accompanied by a suitable data storage medium (such as a floppy disk or a CD-ROM) when it is first connected/installed at the host computer, and that the host computer will be able to search for, locate and retrieve the device driver stored by that data storage medium. It is disadvantageous to require a user of a mobile peripheral device also to carry a storage medium (such as a floppy disk or CD-ROM) for use when connecting the mobile device to a given host computer.

It is therefore desirable to give a host computer access to a device driver associated with a connected peripheral device, without requiring the peripheral device to be accompanied by a data storage medium having the device driver stored therein, and without requiring the host computer to search for the device driver.

According to the present invention, a peripheral device connected to a host computer is cooperable with the host computer to direct the host computer to the appropriate device driver (or other device-related information), which is stored at an external location accessible to the host computer via an external data path. The host computer can then download the device driver automatically.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 diagrammatically illustrates exemplary cooperation between a host computer and a connected peripheral device to direct the host computer to a corresponding device driver according to the invention.

-3-

FIGURE 2 illustrates an example of a partial URL (Uniform Resource Locator) of FIGURE 1.

FIGURE 3 illustrates an example of a completed URL of FIGURE 1.

FIGURE 4 illustrates an example of a filename that is associated with a peripheral device driver and can be produced by the host computer of FIGURE 1 in response to the partial URL of FIGURE 1.

FIGURE 5 illustrates exemplary operations which the host computer of FIGURE 1 can perform according to the invention.

FIGURE 6 illustrates a partial pointer stored in an information storage portion of a peripheral device according to the invention.

FIGURE 7 diagrammatically illustrates pertinent portions of an exemplary embodiment of a host computer according to the invention.

FIGURE 8 diagrammatically illustrates pertinent portions of an exemplary peripheral device according to the invention.

FIGURE 9 illustrates another example of a partial pointer according to the invention.

FIGURE 10 illustrates another example of a completed pointer according to the invention.

FIGURE 11 diagrammatically illustrates pertinent portions of an exemplary embodiment of a host computer according to the invention.

#### DETAILED DESCRIPTION

According to the present invention, a peripheral device connected to a host can provide to the host a partial pointer from which the host can produce a completed pointer. The completed pointer points to a site where the device driver for the peripheral device is located in a data network. The host computer uses the completed pointer to access the site via the data network and download the device driver.

FIGURE 1 is a block diagram of an exemplary arrangement according to the invention. FIGURE 1 illustrates a host computer 11 coupled to one or more peripheral devices at 13. The host computer can be, for example, a personal computer, an engineering workstation, or any data processing apparatus which has need for

-4-

connection to and interaction with peripheral devices. The peripheral devices at 13 can be virtually any peripheral devices, for example a printer, a monitor or a PCI card. As mentioned above, the invention is particularly useful when applied to relatively mobile peripheral devices such as wireless devices, which can be easily moved about and connected to a variety of host computers.

5 In FIGURE 1, when the host computer 11 requires a device driver for a given peripheral device, for example, when the host first communicates with a recently connected peripheral device at 13, the peripheral device communicates to the host computer 11 a partial pointer (a portion of a pointer), for example a partial URL as shown in FIGURE 1. The host computer receives the partial pointer and produces therefrom a completed pointer, for example a completed URL as shown in FIGURE 1. The host computer 11 can then use a suitable output 14 to apply the completed pointer to a data network that is accessible to the host and has stored at a site 17 therein the device driver associated with the peripheral device. The completed pointer points to the site where the device driver is located, so the host can download the driver automatically via an input 16 thereof.

15 In the example of FIGURE 1, the device driver can be located at, for example, a web site that is accessible to the host computer via the Internet and is pointed to by the completed pointer (in this example the completed URL). Once the web site has been accessed by the completed URL, the device driver can be downloaded to the host computer via the Internet (not shown in FIGURE 1).

20 As shown in FIGURE 1, the completed pointer (e.g., URL) can also point, for example, to product information about the peripheral device, other applications associated with the peripheral device, or any other downloadable files associated with the peripheral device.

25 Also as shown in FIGURE 1, in another embodiment, the completed pointer can point to an FTP (file transfer protocol) site accessible to the host via the Internet and capable of storing device drivers, product information, other applications, etc.

FIGURE 2 illustrates one example of a partial URL (a portion of a URL) which can be output from one of the peripheral devices 13 to the host computer 11. The asterisk "\*" in FIGURE 2 represents a missing portion of the partial URL in

-5-

FIGURE 2, which will be provided by the host computer in order to produce a completed URL.

FIGURE 3 illustrates an example of a completed URL produced by the host computer 11 in response to receiving the exemplary partial URL of FIGURE 2. In the completed URL of FIGURE 3, the host computer has replaced the asterisk "\*" with a platform/operating system identifier, namely Win98\_x86, which identifies the host computer platform as an x86 platform and the host computer operating system as Windows 98. The completed URL such as shown in FIGURE 3 points to the web/FTP site of FIGURE 1 where, for example, the manufacturer of the peripheral device has stored one or more of the device drivers, product information, applications and other downloadable files associated with that particular peripheral device. Because of the platform/operating system identifier Win98\_x86 in the URL of FIGURE 3, the URL points to the correct drivers, applications, etc. which are intended by the peripheral device manufacturer to operate with the platform/operating system of the host computer 11. Other forms of identification of the platform/operating system can be used, provided they are known by the manufacturer of the peripheral device so the site 17 can be set up accordingly.

FIGURE 4 illustrates a filename which can be constructed by the host computer 11 from the information in the exemplary partial URL of FIGURE 2. Such a filename can be constructed, for example, in the event that the device driver is not available at the site referenced by the URL of FIGURE 3. Starting from the leftmost character in the partial URL of FIGURE 2, the host can delete all characters up to and including the asterisk "\*", and then insert "file:/a:/Win98\_x86" in place of the deleted characters to produce the filename of FIGURE 4. This filename can then be used by the host computer to attempt to retrieve the appropriate device driver from a storage medium accessible to the host computer (see 15 in FIGURE 1), such as a floppy disk or CD-ROM provided with the peripheral device and inserted into the host.

The above-described operations of the host computer in the arrangement of FIGURE 1 are further understood with reference to the exemplary flow diagram of FIGURE 5. When the driver is to be installed in a given peripheral drive, the host computer 11 selects that device at 51, and thereafter at 53 requests and receives from

-6-

the selected peripheral device the partial pointer, for example the partial URL of FIGURE 2. At 55, the host computer completes the URL by adding the operating system/platform identification information. Thereafter, at 57 the host computer uses the completed URL to download the device driver (or product information, etc.)  
5 automatically from, for example a web site or FTP site. If it is determined at 58 that the driver has been obtained, then the next device can be selected at 51.

If it is determined at 58 that the driver has not been obtained using the URL, it can be next determined at 59 whether or not to use conventional Plug-and-Play techniques to obtain the driver. If so, then conventional Plug-and-Play techniques are  
10 used at 56. Decision block 59 also provides an alternative to conventional Plug-and-Play, namely the path leading to block 54 wherein the host computer uses the partial URL to produce a filename such as discussed above and illustrated in FIGURE 4. The host computer then uses the filename at 52 to access a storage medium such as a floppy disk or CD-ROM from which to load the appropriate device driver (or product  
15 information, etc.).

FIGURE 6 illustrates pertinent portions of an exemplary embodiment of a peripheral device 13 according to the present invention. As shown in FIGURE 6, the peripheral device includes an information storage portion (for example a ROM) wherein the partial pointer (for example a partial URL) is stored. The partial pointer  
20 can then be provided to the host, for example, upon initial connection/communication with the host.

FIGURE 7 illustrates pertinent portions of an exemplary embodiment of a host computer according to the invention. The host computer of FIGURE 7 includes completion logic 71 which receives as an input the partial pointer information from  
25 the peripheral device. The completion logic inserts the operating system/platform identification information (OS/PLATFORM ID) into the partial pointer to produce a completed pointer which is output from the completion logic 71. The completion logic 71 is also selectively operable to produce a filename (such as shown in FIGURE 4) from the partial pointer information. The filename can then be used to access the  
30 information storage medium at 15 and obtain therefrom a device driver, as discussed above with reference to FIGURE 5.



-7-

As shown in FIGURE 2, the partial pointer can include a predetermined symbol or character such as an asterisk "\*", such that the completion logic 71 can find and remove the asterisk "\*" from the partial pointer, and insert the operating system/platform identification information at that point in the partial pointer. In other  
5 embodiments, the completion logic can locate the insertion point by other techniques, for example by a known character offset count from the beginning or end of the partial pointer.

The completed pointer is input to a network interface 73 which interfaces the host computer to a data network 75. In the aforementioned example wherein the data  
10 network is the Internet, the network interface 73 can be, for example, a web browser. The data network 75 includes the driver site 77 (for example the web site 17 of FIGURE 1) where the driver, product information, application, etc. is located. The network interface 73 applies the completed pointer to the data network 75, and the device driver from site 77 is returned automatically via the data network 75 to the  
15 network interface 73 in response to the completed pointer. The driver can then be supplied to an execution unit 79 which will execute the driver software. FIGURE 7 also illustrates at 72 that the completed pointer may also point to (and be used to retrieve) other information such as product information, applications useable with the peripheral device, or other downloadable files related to the peripheral device.

FIGURE 8 illustrates pertinent portions of another exemplary peripheral device according to the invention. FIGURE 8 illustrates that the peripheral device itself can include, in an information storage portion (e.g., a ROM) thereof, its own device drivers, product information, applications, etc. That is, the driver site illustrated in FIGURES 1 and 7 is provided in the peripheral device itself. In this embodiment, the  
20 peripheral device includes a network interface 81 that interfaces the peripheral device to a data network. For example, if the network interface 81 supports TCP/IP and e.g. HTTP, then the host computer can utilize the Internet to retrieve the device driver, product information, applications, etc. automatically from the peripheral device itself.

FIGURE 11 is similar to FIGURE 7, but illustrates exemplary embodiments wherein the completed pointer can be used to access the desired peripheral device  
30 driver from an external peripheral device 83 that is accessible to the host computer via

-8-

an external data path provided by any desired communication medium 82. The communication medium 82 can include interfaces 82a and 82b. The interface 82a of the host computer communicates with the interface 82b of the peripheral device 83. The communication medium 82 can be, for example, a point-to-point medium. Specific examples of the communication medium 82 include: an RF link, using a point-to-point protocol such as PPP (on top of which may run HTTP/TCP/IP); a serial cable; an infrared interface using the IrDA (Infrared Data Association) protocols; a Universal Serial Bus (USB); and a PC-Card (e.g., PCMCIA). In some embodiments (e.g., the RF link embodiment), internet protocols or other desired protocols may run on top of the interfaces 82a and 82b.

Additionally, a communication medium such as shown at 82 can be used by a peripheral device 13 of FIGURES 1, 7 or 11 to communicate the partial pointer to the host computer. Moreover, in some embodiments wherein the driver site is provided in the same peripheral device 13 that provides the partial pointer, the communication medium 82 can be used to communicate both the partial pointer and the device driver from the peripheral device 13 to the host computer (such embodiments are designated by the use of reference character "83(13)" and the broken line in FIGURE 11).

Examples of the peripheral devices 13 or 83 in FIGURES 1, 7 or 11 include a cellular telephone, a printer, a camera, a speaker, a laptop PC or any information storage device.

FIGURE 9 illustrates an example of a partial URL which can be provided to a host computer by a peripheral device, and which is intended to permit the host computer to access product information about the peripheral device. Again, the asterisk "\*" is used to identify the insertion point for the platform/operating system identifier.

FIGURE 10 illustrates a completed URL which can be produced by the host computer in response to receiving the partial URL of FIGURE 9. In the example of FIGURE 10, platform/operating system identifier Win2000\_x86 indicates that the operating system is Windows 2000 and the platform is an x86 platform. Furthermore,

-9-

in the example of FIGURES 9 and 10, the peripheral device is an Ericsson mobile telephone, model no. GH198.

Although exemplary embodiments of the present invention have been described above in detail, this does not limit the scope of the invention, which can be  
5 practiced in a variety of embodiments.

-10-

## WHAT IS CLAIMED IS:

1. A method for a host computer to access device information corresponding to a peripheral device that is to be used in cooperation with the host computer, comprising:

5                   the peripheral device providing to the host computer information indicative of a pointer;

                  the host computer producing in response to the information a pointer that points to a location from which the device information can be automatically downloaded to the host computer; and

10                  the host computer using the pointer to download automatically the device information.

2. The method of Claim 1, wherein said providing step includes the peripheral device providing to the host computer a first portion of the pointer.

15

3. The method of Claim 2, wherein said producing step includes the host computer providing a second portion of the pointer.

4. The method of Claim 3, wherein said producing step includes the host  
20                  computer combining said first and second portions to form the pointer.

5. The method of Claim 4, wherein said combining step includes the host computer inserting the second portion at a predetermined location in the first portion.

25                  6. The method of Claim 5, wherein said first portion includes a predetermined symbol at said predetermined location thereof.

7. The method of Claim 6, wherein said combining step includes the host computer detecting and removing the predetermined symbol from the first portion.

30

-11-

8. The method of Claim 3, wherein the second portion includes information indicative of a computing platform of the host computer and an operating system of the host computer.

5 9. The method of Claim 1, wherein said using step includes the host computer applying the pointer to a data network and downloading the device information from a storage site accessible to the data network.

10 10. The method of Claim 9, wherein said downloading step includes downloading the device information from the peripheral device.

11. The method of Claim 9, wherein said downloading step includes downloading the device information from a web site.

15 12. The method of Claim 1, wherein said device information includes one of a device driver associated with the peripheral device, product information about the peripheral device, and an application associated with the peripheral device

20 13. The method of Claim 1, wherein said pointer is a URL.

14. A host computing apparatus connectable to a peripheral device for cooperation therewith, comprising:

an input for receiving from a peripheral device pointer information indicative of a pointer;

25 logic coupled to said input and responsive to said pointer information for producing a pointer which points to an external storage location that is accessible via an external data path and has stored therein device information that the host computing apparatus can use to interface with the peripheral device;

30 an output coupled to said logic for applying said pointer to the external data path; and

an input for receiving the device information from the external data path.

-12-

15. The apparatus of Claim 14, wherein said device information includes one of a device driver associated with the peripheral device, product information about the peripheral device, and an application associated with the peripheral device.

5 16. The apparatus of Claim 14, wherein said logic is selectively operable to produce from the pointer information a file name for use in accessing a storage medium in the host apparatus to retrieve therefrom the device information.

10 17. The apparatus of Claim 14, wherein said pointer is a URL.

18. The apparatus of Claim 14, wherein said pointer information includes a first portion of the pointer, and said logic is operable to insert into the first portion, at a predetermined location in the first portion, a second portion of the pointer including information indicative of a computing platform and an operating system of the host computing apparatus.

15 19. The apparatus of Claim 14, wherein said input for receiving pointer information is the same input as said input for receiving device information.

20 20. The apparatus of Claim 14, wherein said input for receiving pointer information is separate from said input for receiving device information.

21. A peripheral device for connection to and cooperation with a host computing apparatus, comprising:

25 a storage apparatus for storing therein information from which the host computing apparatus can produce a pointer which points to a storage location that is located externally of the host computing apparatus and is accessible to the host computing apparatus via an external data path and has stored therein device information that the host computing apparatus can use to interface with the peripheral device; and

30

-13-

an output coupled to said storage apparatus for outputting said information to said host computing apparatus.

5           22.     The device of Claim 21, including an interface for permitting said peripheral device to be accessed from the data path, and wherein said storage location is provided in said peripheral device.

          23.     The device of Claim 21, wherein the peripheral device is a wireless mobile communication device.

10

          24.     The device of Claim 21, wherein said pointer is a URL.

          25.     The device of Claim 21, wherein said device information includes one of a device driver associated with the peripheral device, product information about the peripheral device, and an application associated with the peripheral device

15

FIG.1

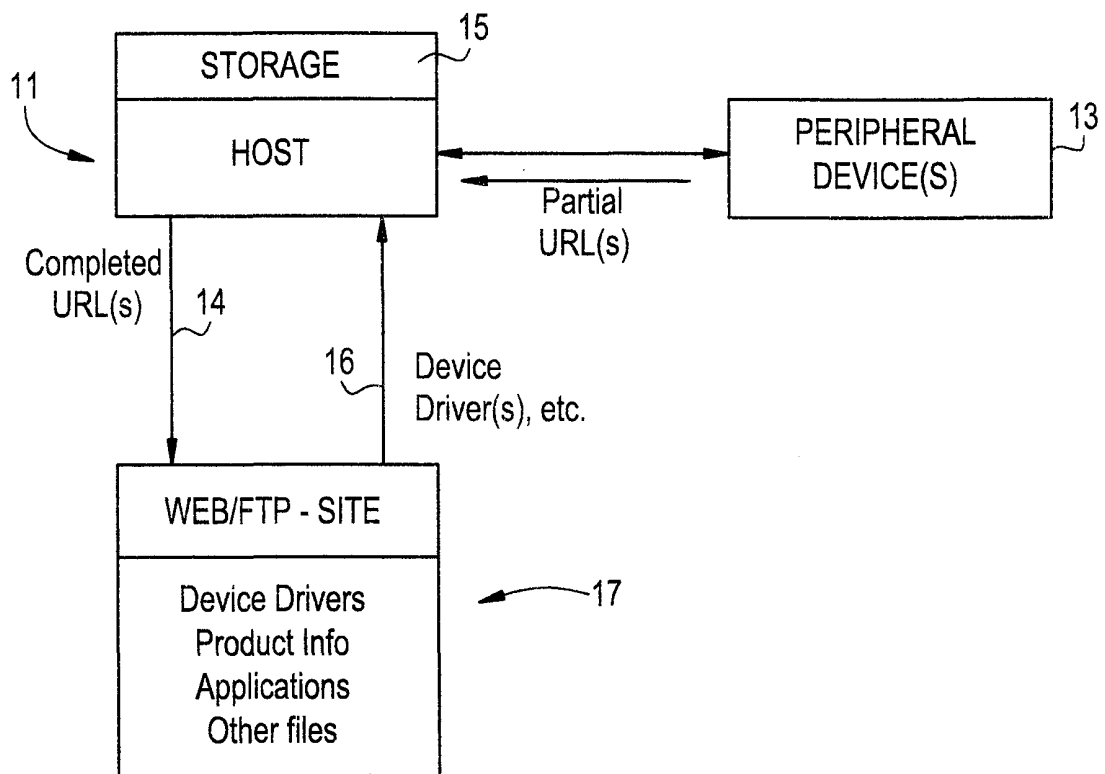


FIG.2

[http://www.ericsson.com/drivers/\\*/SH888.BIN](http://www.ericsson.com/drivers/*/SH888.BIN)

FIG.3

[http://www.ericsson.com/drivers/Win98\\_x86/SH888.BIN](http://www.ericsson.com/drivers/Win98_x86/SH888.BIN)

FIG.4

[file:/a:/Win98\\_x86/SH888.BIN](file:/a:/Win98_x86/SH888.BIN)



FIG. 5

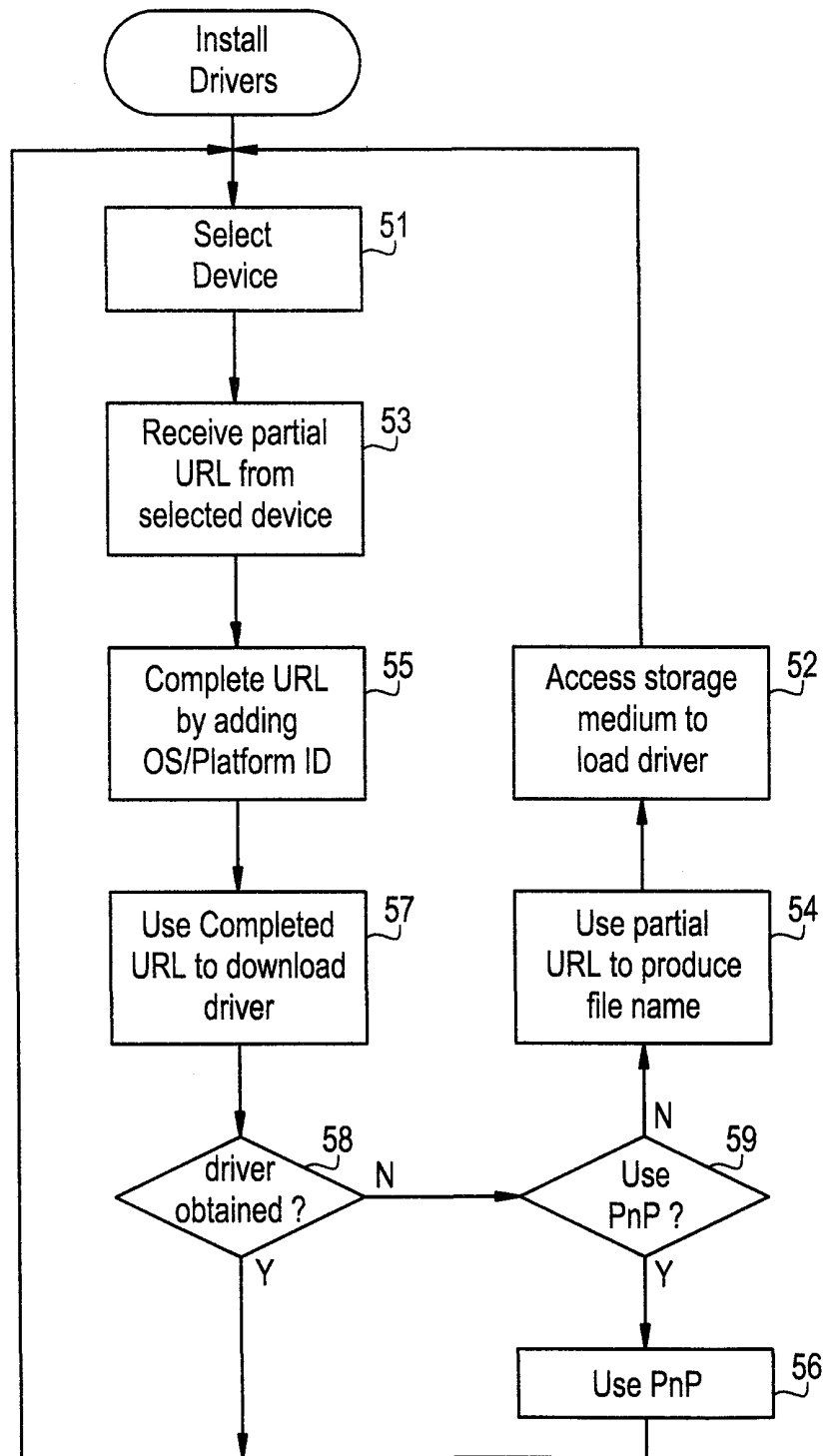


FIG. 6

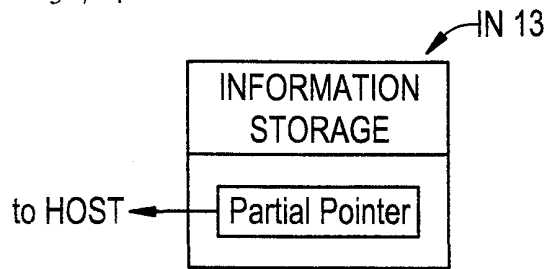


FIG. 7

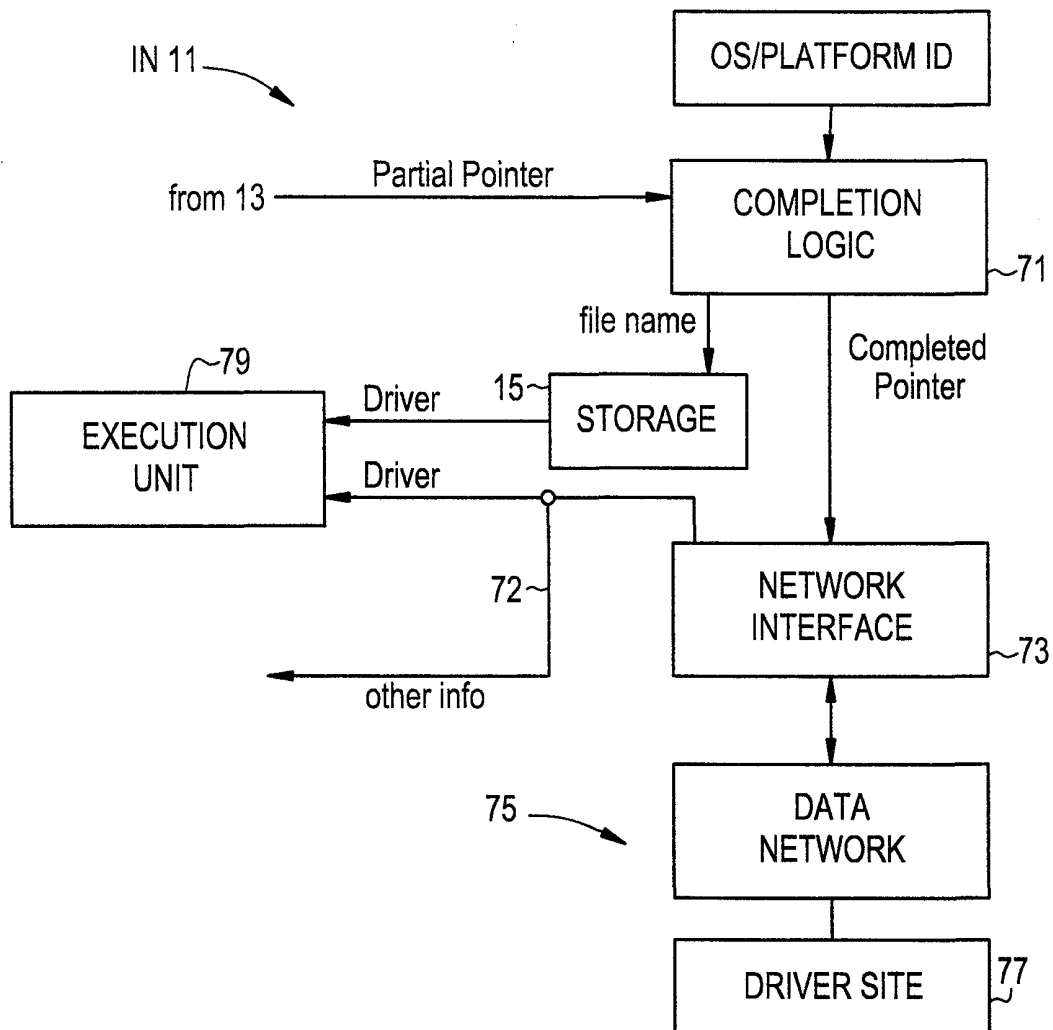
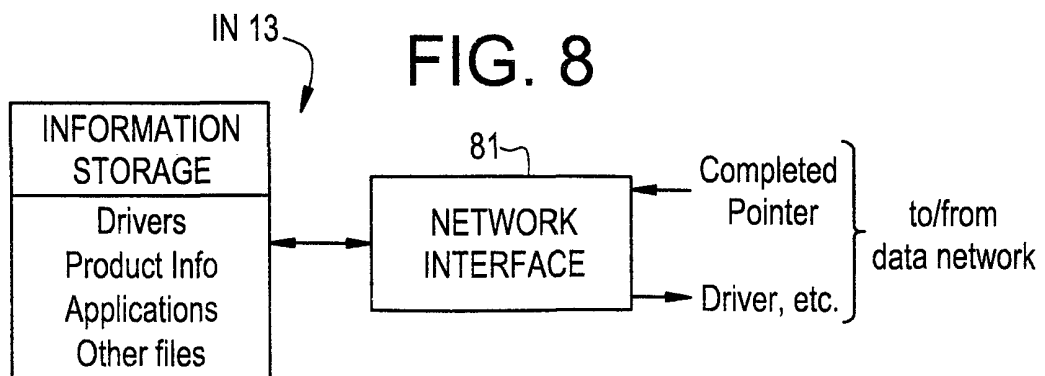


FIG. 8



## FIG.9

[http://www.ericsson.com/prodinfo/\\*/GH198.html](http://www.ericsson.com/prodinfo/*/GH198.html)

## FIG.10

[http://www.ericsson.com/prodinfo/Win2000\\_x86/GH198.html](http://www.ericsson.com/prodinfo/Win2000_x86/GH198.html)

## FIG.11

