SOFTWARE DISTRIBUTION, EXECUTION AND UPGRADING

This invention relates to methods for distribution, execution and upgrading of software. It relates particularly but not exclusively to methods for executing and distributing software over a communication network such as a local area network, a wide area network, a virtual private network, or the Internet.

Most current application software consists of a large number of different files. Such software is typically distributed on one or more CD-ROMs. The software is installed on a user's computer by running a set-up or installation program, which writes different files to different directories or folders on storage media (such as a hard disk) on the user's computer. When the software application is executed, it makes file requests to the various different files as they are needed. The file requests are typically made to a particular file directory location on the user's computer, and the software may cease to operate properly if a file is not found in the expected location. For some software, the file requests are made to a CD-ROM drive on the user's computer, and the software may cease to operate properly if the correct CD-ROM is not present in the drive.

Disadvantages of software distribution using CD-ROMs or other physical distribution media include the costs of creating the CD-ROMs, the effort and expense involved in creating printed manuals and packaging, and the time delays and transportation logistics involved in physically delivering the media to the user. Different users have different configurations of computers and different requirements for software functionality. Accordingly, different users require slightly different (sometimes significantly different) versions of the same software, and in order to achieve this the software manufacturer typically includes all possible variations on the distribution media, with the user selecting the applicable components during installation. From the user's point of view, there is the inconvenience of receiving more materials than may be required. Typically, a user never finds a use for all of the software which is received, and the functionality of the software far exceeds what is required. After a while, superseded CD-ROMs take up useful space. The user may have to pay for and install the whole of a software package when all that was required was a small part of it, and the user may have preferred to pay a lesser amount for software which takes up less data storage space on the user's computer. Moreover, a user who uses the software

only occasionally has to pay the same amount as a person who uses the software frequently.

It is becoming increasingly common to distribute software over the Internet. This is typically done by collecting all the files for a software application together into a compressed archive, making the archive available on an FTP or Web server on the Internet, and downloading the archive from the FTP or Web server onto the user's computer. The user then runs an installation program which decompresses the archive to extract the files, and then writes different files to different directories on the hard disk of the user's computer in much the same way as the installation program on a CD-ROM operates.

Apart from the issues of wastage and delivery of physical media, this method of delivery of software includes many of the disadvantages of conventional physical delivery. It is still usually necessary for the user to download more files than are actually required for the user's particular configuration, and this can be problematic in view of the time costs typically associated with connecting to the Internet. There is no means for charging the user only for the parts of the software which are actually useful to the user, or for charging the user on the basis of the frequency with which the software is used. Moreover, once a complete archive of the relevant software has been downloaded by the user, it is relatively easy for the user to make and distribute unauthorised copies, so piracy issues arise.

A further problem associated with conventional software distribution methods relates to the manner in which the user decides whether the software is suitable for the user's purposes. It is typically very difficult for a user to decide whether software is suitable merely from reading advertising material, and the suitability can in many cases only be verified some time after the user has purchased the software and gone through the effort necessary to learn how to use it. This can result in a user purchasing unsuitable software. One answer to this problem is provided by cut-down "demonstration" versions of the software which may be distributed at low cost or free of charge via physical media (such as on CD-ROMs sold with computer magazines) or over the Internet. However, because it is usually necessary for a user to invest a significant amount of time in installation and learning how to use a new software application, there is often a reluctance to spend the time necessary to install and learn how to use a cut-down software product which will never in itself be useful.

Another type of software distribution is involved in client-server networking, typically in the context of local area networks. A typical local area network involves a number of client computers connected to one or more file servers. Networking software located on a client computer and the file server enables the client computer to gain access to files on the file server as if the files were located on a disk drive on the client computer. From the client computer's point of view, the file server simply appears to be one or more additional file volumes on the client computer. As a result, it is possible to run client-server software applications, in which some of the files are located on the client computer and some are located on the file server. In peer-to-peer networking, a similar result is achieved in a system where each computer can be configured to allow the other computers in the network to have access to part or all of its disk volumes.

An advantage of client-server networking and peer-to-peer networking is that it is not necessary for each computer in the network to have a complete copy of all the files of the software application; shared files need only be present on one computer. Accordingly, computer storage media space is not wasted, and upgrading can be a simpler process. However, this type of networking can typically only be achieved within an organisation which has the same networking software on all computers. Efficiencies can be gained in distributing software within an organisation, but not between organisations or to individuals which are not part of the organisation.

US Patent 5,752,005 describes a foreign file system establishing method which uses a native file system virtual device driver. A computer has a native file system which provides various native function routines. A virtual driver communicates with the native file system and intercepts some of the function routines, transferring them to a particular location on another ("foreign") computer. This allows the computer to gain some of the advantages of client-server networking, but it is necessary for the device driver to emulate the foreign file system.

An object of the present invention is to provide new methods of distributing, executing and upgrading software which overcome some or all of the difficulties described above.

According to a first aspect of the invention, there is provided a method of distributing software over a communication network including the steps of:

(a) making the software available on a file server attached to the network;

(b) providing and executing an installation application on a user's computer attached to the network;

(c) monitoring file requests made by the installation application;

(d) identifying file requests which relate to files which are not present on the user's computer;

(e) downloading the identified files from the file server;

(f) storing the downloaded files in storage media on the user's computer; and

(g) directing the file requests for the identified files, and any future file requests for those files, to the downloaded versions of those files.

The communication network may be any suitable form of communication network. Suitable communication networks include the Internet, a local area network, a wide area network, and a virtual private network. It is currently envisaged that the invention will be used most extensively in the context of distribution of software over the Internet. It will be seen that one of the advantages of this aspect of the invention is that a user downloads only the files which are required by the user.

The file server may be any suitable type of file server. Suitable types of file servers, with particular reference to Internet applications, include an FTP server, and HTTP server and a Gopher server.

The method of this aspect of the invention may further include the steps of:

(h) monitoring the number of times that files are requested and/or the particular files requested by the user's computer from the server; and

(i) requiring the owner of the user's computer to make payments based on the number of times that files are requested and/or the particular files requested.

These steps enable the software manufacturer to charge a differential price to users based on the amount of the software which the user actually uses and/or on the frequency of use. Accordingly, a person who uses only a small part of the software or who uses it only occasionally can be charged a smaller fee, whereas a person who uses more extensive parts of the software or uses it frequently can be charged a higher fee.

During the installation process, the software may optionally require that a key be entered.

One of the advantages of the first aspect of the invention is that it can allow distribution of currently available software without any modification of the file requests made by that software. This is because file requests are intercepted. If the file is already on the user's computer, the file request is directed to that file. If the file is not on the user's computer, the file is downloaded from the file server and then accessed on the user's computer. Currently available software might make numerous file requests to the local CD-ROM drive, because the software has been programmed with the assumption that the software is being installed from a local CD-ROM. All of these requests are monitored and converted into file download requests directed to the file server or redirected to downloaded files located on the user's computer.

According to a second aspect of the invention, there is provided a method of executing computer application software on a user's computer which is connected to a communication network including the steps of:

(a) providing on the user's computer a version of the software which does not include all the files necessary for complete operation of the software;

(b) making the missing files available on a file server attached to the network;

(c) executing the software on the user's computer;

(d) monitoring file requests made by the software;

(e) identifying file requests which relate to files which are not present on the user's computer;

(f) downloading the identified files from the file server and storing them in volatile or non-volatile storage media on the user's computer;

(g) directing the file requests for the identified files to the downloaded versions of those files; and

(h) deleting one or more of the downloaded files from the user's computer.

The communication network and the file server may have the same features as described above in relation to the first aspect of the invention.

The step of deleting one or more of the downloaded files may be done at any suitable time, including:

(i) as the files are closed by the software;

(ii) when execution of the software terminates;

(iii) after a pre-determined time or number of days has elapsed; and/or

(iii) after the software has been executed a pre-determined number of times.

The method of this aspect of the invention may further include the steps of:

(i) monitoring the number of times that files are requested and/or the particular files requested by the user's computer from the server; and

(j) requiring the owner of the user's computer to make payments based on the number of times that files are requested and/or the particular files requested.

It will be seen that this method requires the user's computer to make contact with the file server at regular intervals in order to replace deleted files, and this facilitates the pay-per-use type of charging. This method is also useful for overcoming the purchasing barrier – allowing users to try the actual software first before buying it. There is no need for the software to be a cut-down version.

According to a third aspect of the invention, there is provided a method of automatically upgrading software on a user's computer over a communication network including the steps of:

(a) making the upgrade software available on a file server attached to the network;

(b) ascertaining that the software on the user's computer is an older version than the upgrade software;

(c) executing the software on the user's computer and monitoring file requests made by the software;

(d) identifying file requests which relate to files which have been upgraded in the upgrade software;

(e) downloading the identified files from the file server;

(f) storing the downloaded files in storage media on the user's computer; and

(g) directing the file requests for the identified files, and any future file requests for those files, to the downloaded versions of those files.

The communication network and the file server may have the same features as described above in relation to the first aspect of the invention.

The step of ascertaining that the software on the user's computer is an older version than the upgrade software may occur at any suitable time, including:

(i) each time the software on the user's computer is executed;

(ii) after the software on the user's computer has been executed a predetermined number of times; or

(iii) after a predetermined period of time.

When it is ascertained that the software on the user's computer is an older version than the upgrade software, an upgrade agent may be downloaded from the file server to the user's computer. The upgrade agent performs the function of monitoring file requests made by the software to identify file requests which relate to files which have been upgraded in the upgrade software.

The invention will hereinafter be described in greater detail by reference to the attached drawings which show an example form of the invention. It is to be understood that the particularity of those drawings does not supersede the generality of the preceding description of the invention.

Figure 1 is a block diagram showing schematically the software components in a conventional networking system.

Figure 2 is a block diagram showing schematically the software components in a networking system according to the present invention.

Figure 3 is a schematic diagram of the components in a preferred embodiment of the present invention.

Referring firstly to Figure 1, there is shown the components in conventional directory mounting/mapping products such as NFS, SMB and NetWare. On the host machine, application software runs on top of an operating system. The application software communicates with the operating system through various application program interfaces (APIs). Included within the operating system are a native file system, which includes a multiplicity of function routines, and network drivers. On the remote computer, the operating system also includes a file system and network drivers. When the application software on the host machine accesses a file on the remote machine, it does so by making a request to the file system on the host machine. This request then passes to the network driver on the host machine, which passes the request to the network driver on the remote machine, which in turn passes it to the file system on the remote machine to access the file. The response is sent back through the same communication chain. There is no trapping of file requests made from the application software before they reach the native file system. Instead, the native file system invokes special drivers called network redirectors, which co-operate with their counterparts on the remote machine to make the directories and files visible to the native file system.

Referring now to Figure 2, the present invention enables software to be installed and run on a host machine, without requiring the underlying files to be

physically available on any of the local storage drives. Instead, every file request from the software is trapped by an installable monitor, which dynamically downloads the target file from a remote file server to a local storage medium if the file has not already been downloaded. The installable monitor thereafter directs the native file system to operate on the downloaded file in order to satisfy the file request. Even requests for a file at an absolute path on a specific drive (e.g. the CD-ROM drive) are redirected to the downloaded file, which is likely to reside at a different path on a different drive. The downloaded file can optionally be removed when the software closes the file, or when the software terminates. Consequently, a host machine can run application software without having a physical medium containing all the underlying files.

In a preferred embodiment as illustrated in Figure 3, the file server is a standard FTP server using the FTP communication protocol. The communication network is the Internet. The user's computer has Windows 95 as its operating system, and the FTP server has Solaris as its operating system. However, those skilled in the art should appreciate that different operating systems have similar features and functions, so that the present invention is not limited to use only in connection with these operating systems.

The installable monitor on the user's computer has two components: a virtual device driver (VxD) inside the operating system kernel, and a Windows application at the user level, which includes an FTP client (the user mode application). The VxD is implemented in such a way that it layers itself on top of the native file system. It is implemented as a hook that gets installed below the installable file system (IFS) manager interface. A hook is a programming device that alters program flow away from its original intent. Consequently, for selected program flow threads progressing through the IFS manager interface, program flow is routed to the VxD rather than to the native file system as originally intended. The VxD is then free to either satisfy the request and return program flow back to the requesting thread or modify the request before directing it to the native file system.

The native file system may include function routines directed towards file rights, file locking, file transaction tracking and the like. These are of little importance to the installable monitor of the invention, and hence hooks are omitted

for such functions. The selected functions for which hooks are implemented in the preferred embodiment are indicated in the following table:

CreateFile	WriteFile	FindNextFile
CloseFile	GetFileSize	GetFileAttributes
ReadFile	FindFirstFile	GetFileTime
CreateDirectory	DeleteFile	

The precise processes and program flow used in hooking these function routines are unimportant for the purposes of the present invention. Such hooking may be implemented any number of different ways, as will be known to those skilled in the art. For example, published information describing the operating system may indicate the definition of the IFS manager interface for various function routines supported by the native file system. To the extent that published information is lacking, hook locations may be determined empirically.

For each function routine that is hooked, the VxD blocks the execution of the request inside the kernel, determines the path name of the request issued by the CD application and translates it into the equivalent remote path name on the CD drive attached to the file server. This remote path name is then passed back to the user mode application using known event notification mechanisms. The user mode application then translates the request into an FTP request and forwards it to the FTP server. The user mode application waits for a response from the FTP server and, once the file has been downloaded onto the user's machine, it saves the file to another location on the user's machine's local storage media. Confirmation of the download is passed to the VxD, together with the location of the local copy of the file. The VxD then redirects the native file system to satisfy the request made by the application using the local copy, and the application continues as normal. The application itself is unaware that its files and data are being accessed on-line rather than on a local CD-ROM. After the application finishes execution, or closes the file, or after a pre-determined number of uses or after a pre-determined number of days, (these options may be configurable) one or more or all of the downloaded files are deleted from the user's machine.

Typical applications for this embodiment include games traditionally distributed via CD-ROM disks. The start-up program corresponds to the game

application that is launched on the user's machine. Typically these applications store their data in the form of movie files on the CD-ROM disk. These are downloaded on demand by the installable monitor on the user's computer, while the user continues to play the game.

It is to be understood that various alterations, modifications and additions can be made to the parts previously described without departing from the ambit of the invention.

Claims:

1. A method of distributing software over a communication network including the steps of:

(a) making the software available on a file server attached to the network;

(b) providing and executing an installation application on a user's computer attached to the network;

(c) monitoring file requests made by the installation application;

(d) identifying file requests which relate to files which are not present on the user's computer;

(e) downloading the identified files from the file server;

(f) storing the downloaded files in storage media on the user's computer; and

(g) directing the file requests for the identified files, and any future file requests for those files, to the downloaded versions of those files.

2. A method according to claim 1 further including the steps of:

(h) monitoring the number of times that files are requested and/or the particular files requested by the user's computer from the server; and

(i) requiring the owner of the user's computer to make payments based on the number of times that files are requested and/or the particular files requested.

3. A method according to claim 1 wherein the software requires a key to be entered during installation.

4. A method according to claim 1 wherein the communication network is the Internet.

5. A method according to claim 1 wherein the communication network is a local area network or a private wide area network.

6. A method according to claim 1 wherein the file server is:

(i) an FTP server;

(ii) an HTTP server; or

(iii) a Gopher server.

* C L M S *

7. A method of executing computer application software on a user's computer which is connected to a communication network including the steps of:

(a) providing on the user's computer a version of the software which does not include all the files necessary for complete operation of the software;

(b) making the missing files available on a file server attached to the network;

(c) executing the software on the user's computer;

(d) monitoring file requests made by the software;

(e) identifying file requests which relate to files which are not present on the user's computer;

(f) downloading the identified files from the file server and storing them in volatile or non-volatile storage media on the user's computer;

(g) directing the file requests for the identified files to the downloaded versions of those files; and

(h) deleting one or more of the downloaded files from the user's computer.

8. A method according to claim 7 wherein the step of deleting one or more of the downloaded files is done:

(i) as the files are closed by the software;

(ii) when execution of the software terminates;

(iii) after a pre-determined time or number of days has elapsed; and/or

(iii) after the software has been executed a pre-determined number of times.

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

(i) monitoring the number of times that files are requested and/or the particular files requested by the user's computer from the server; and

(j) requiring the owner of the user's computer to make payments based on the number of times that files are requested and/or the particular files requested.

10. A method according to claim 7 wherein the communication network is the Internet.

11. A method according to claim 7 wherein the communication network is a local area network or a private wide area network.

* C L M S *

12. A method according to claim 7 wherein the file server is:

(i) an FTP server;

(ii) an HTTP server; or

(iii) a Gopher server.

13. A method of automatically upgrading software on a user's computer over a communication network including the steps of:

(a) making the upgrade software available on a file server attached to the network;

(b) ascertaining that the software on the user's computer is an older version than the upgrade software;

(c) executing the software on the user's computer and monitoring file requests made by the software;

(d) identifying file requests which relate to files which have been upgraded in the upgrade software;

(e) downloading the identified files from the file server;

(f) storing the downloaded files in storage media on the user's computer; and

(g) directing the file requests for the identified files, and any future file requests for those files, to the downloaded versions of those files.

14. A method according to claim 13 wherein the step of ascertaining that the software on the user's computer is an older version than the upgrade software occurs:

(i) each time the software on the user's computer is executed;

(ii) after the software on the user's computer has been executed a predetermined number of times; or

(iii) after a predetermined period of time.

15. A method according to claim 13 wherein, when it is ascertained that the software on the user's computer is an older version than the upgrade software, an upgrade agent is downloaded from the file server to the user's computer and the upgrade agent performs the function of monitoring file requests made by the

software to identify file requests which relate to files which have been upgraded in the upgrade software.

16. A method according to claim 13 wherein the communication network is the Internet.

17. A method according to claim 13 wherein the communication network is a local area network or a private wide area network.

18. A method according to claim 13 wherein the file server is:

(i) an FTP server;

(ii) an HTTP server; or

(iii) a Gopher server.

DRAWINGS

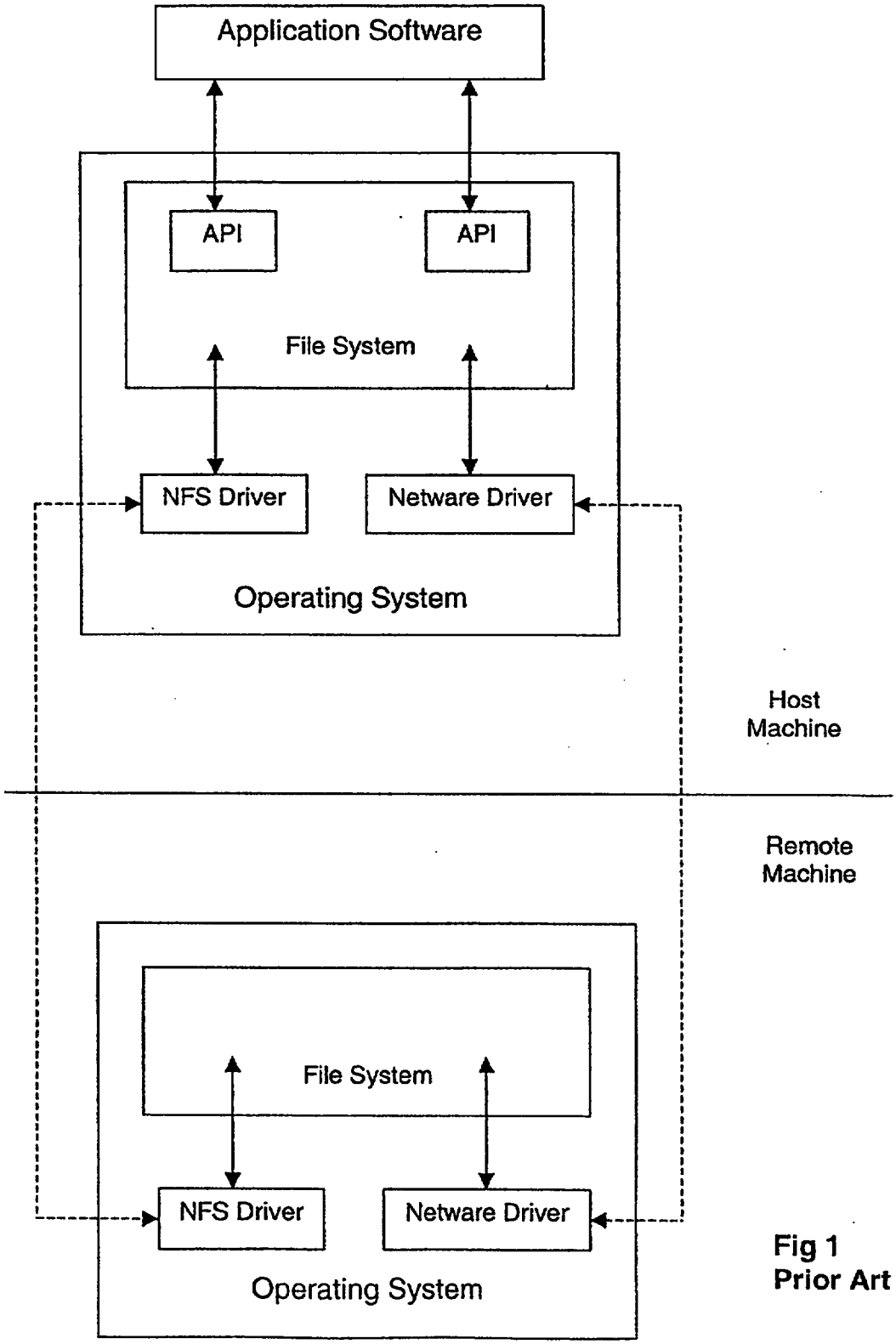


Fig 1
Prior Art

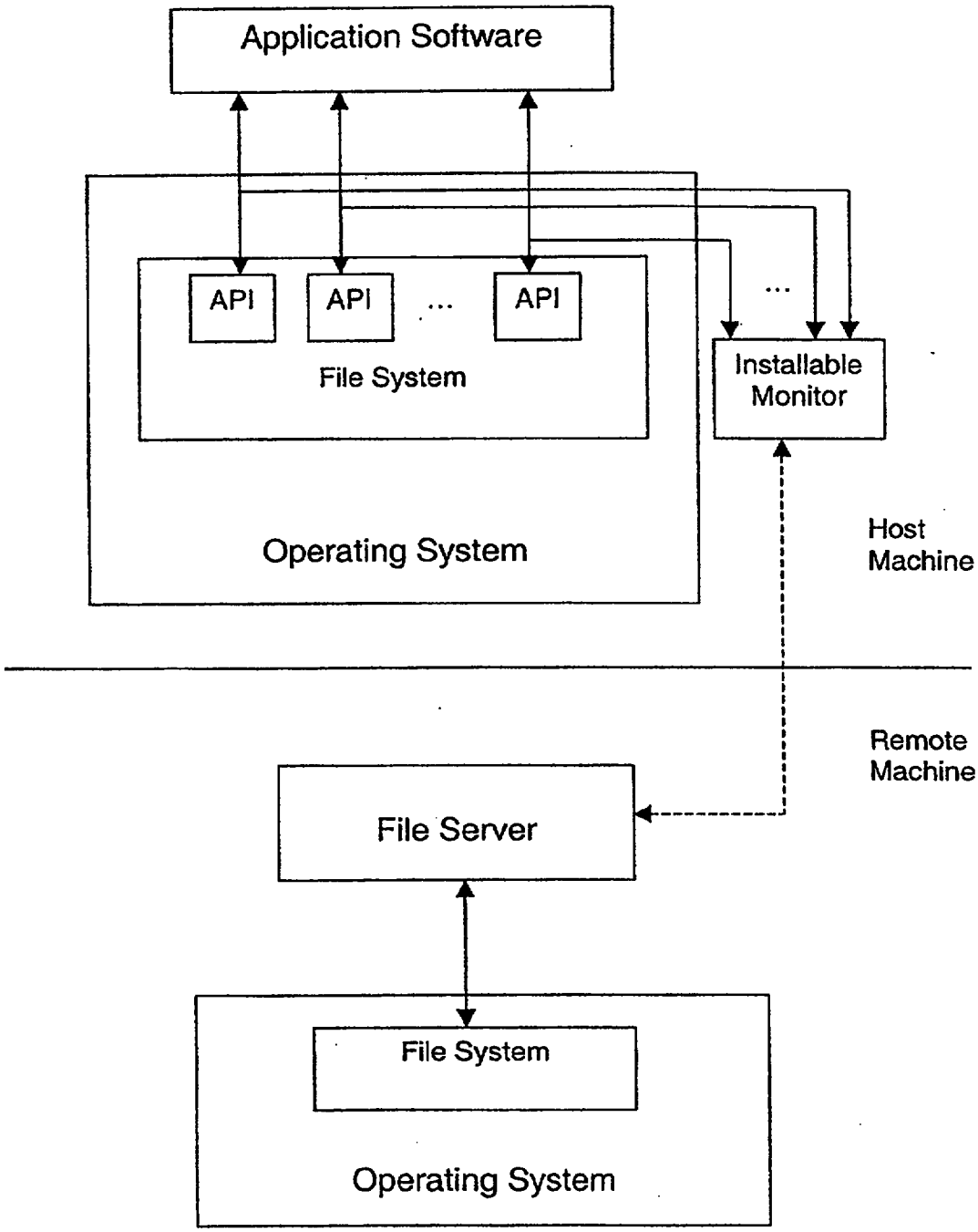

Application Software
API
API
...
API
File System
...
Installable Monitor
Operating System
Host Machine
Remote Machine
File Server
File System
Operating System

Fig 2

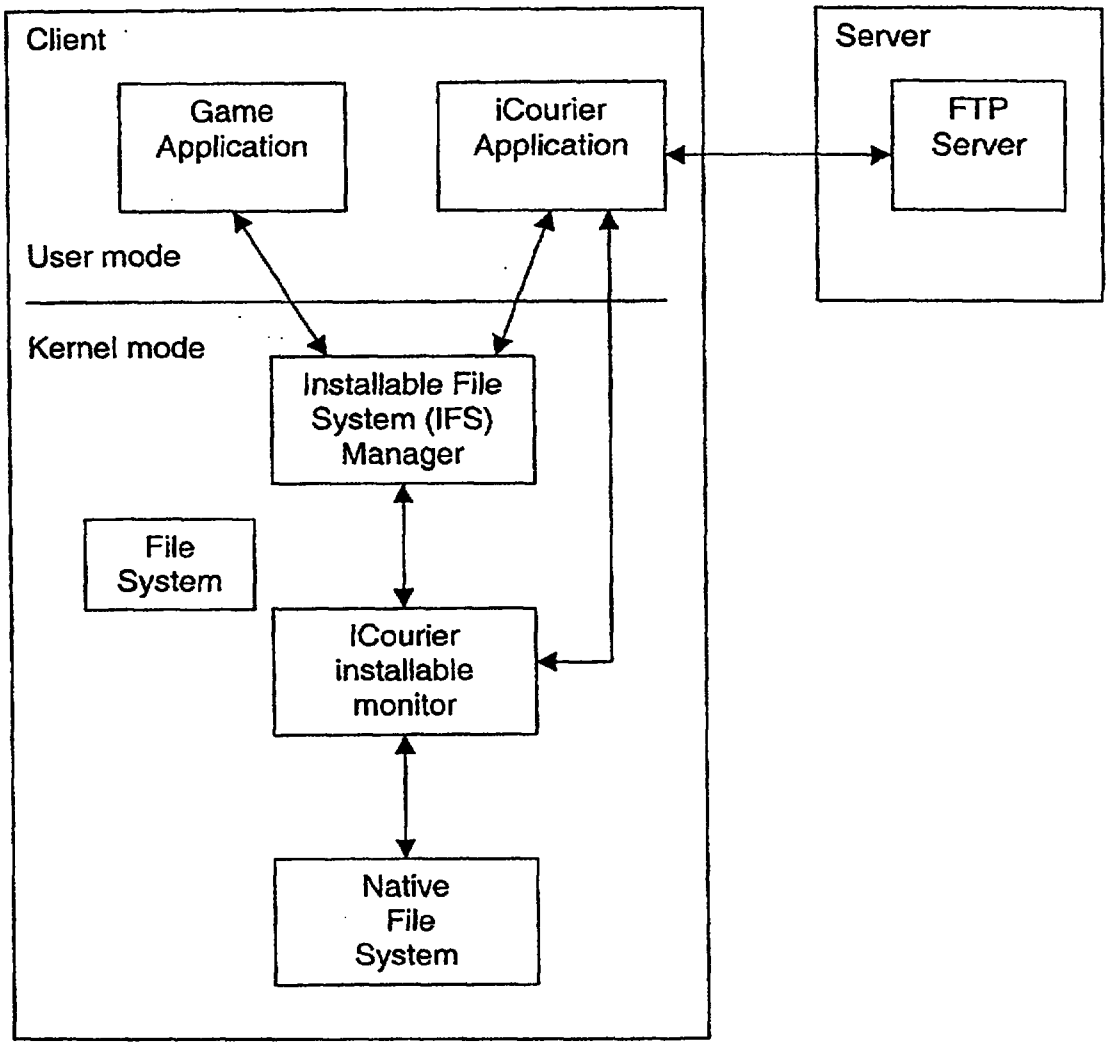

Client
Server
Game Application
iCourier Application
FTP Server
User mode
Kernel mode
Installable File System (IFS) Manager
File System
ICourier installable monitor
Native File System

Fig 3

200104943-6
MRS 10/MY

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 13/00, 15/177	A1	(11) International Publication Number: **WO 00/55740** (43) International Publication Date: 21 September 2000 (21.09.00)

(21) International Application Number: PCT/SG99/00037

(22) International Filing Date: 18 March 1999 (18.03.99)

(71) Applicant *(for all designated States except US)*: KENT RIDGE DIGITAL LABS [SG/SG]; 21 Heng Mui Keng Terrace, Singapore 119613 (SG).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: RANGARAJAN, Sridharan [IN/SG]; Block 379, 5 Clementi Avenue #02–348, Singapore 120379 (SG). LIN, Feng [CN/SG]; Block 110, 13 Woodlands Street #12–20, Singapore 730110 (SG). PANG, Hwee, Hwa [SG/SG]; 19 Shelford Road #01–42, Singapore 288408 (SG).

(74) Agent: HELEN YEO & PARTNERS; 80 Raffles Place, #33–00 UOB Plaza 1, Singapore 048624 (SG).

(81) Designated States: JP, SG, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published

With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SOFTWARE DISTRIBUTION, EXECUTION AND UPGRADING

(57) Abstract

A method of distributing software over a communication network includes the steps of: making the software available on a file server attached to the network; providing and executing an installation application on a user's computer attached to the network; monitoring file requests made by the installation application; identifying file requests which relate to files which are not present on the user's computer; downloading the identified files from the file server; storing the downloaded files in storage media on the user's computer; and directing the file requests for the identified files, and any future file requests for those files, to the downloaded versions of those files. Similar methods are provided for executing software over a network and upgrading software over a network.

