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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 January 2002 (03.01.2002)

PCT

(10) International Publication Number
WO 02/01377 A1

(51) International Patent Classification[7]: G06F 15/00

(21) International Application Number: PCT/KR01/01058

(22) International Filing Date: 21 June 2001 (21.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2000/36621 29 June 2000 (29.06.2000) KR
2001/13220 14 March 2001 (14.03.2001) KR

(71) Applicant *(for all designated States except US)*: **WYSETECH CO., LTD.** [KR/KR]; 7th floor, Woo-Jin Building, 1338-20 Seocho-Dong, Seocho-Ku, Seoul 137-070 (KR).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **KIM, Jong, Jin** [KR/KR]; 106-104, LineDongsan Apt., 930-1 Hwa-jung-Dong, Seo-Ku, Gwangju-City 502-240 (KR). **U, Do, Sam** [KR/KR]; 504-1310, Booyoung Apt., 244 Many-oun-Ri, Hwasoon-Eub, Hwasoon-Kun, Chollanam-do 519-800 (KR).

(74) Agent: **MOON, Kyoung, Jin**; 7th Floor, WooJin Building, 1338-20 Seocho-Dong, Seocho-Ku, Seoul 137-070 (KR).

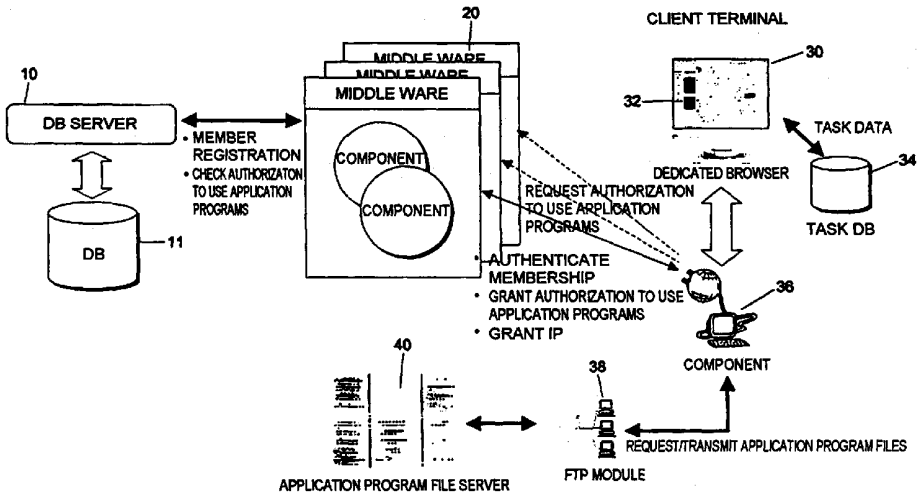
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, 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, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DISTRIBUTED PROCESSING SYSTEM AND METHOD FOR APPLICATION PROGRAMS



(57) Abstract: A distributed processing system and method for application programs, for performing general processes such as user authentication, application program download/update, etc. efficiently through a dedicated browser, when a user wants to execute a desired application program provided on the World Wide Web through the Internet, comprises a DB server (10) which manages a DB (11) having various basic information for performing various processes stored therein; middlewares (20) that communicates with the DB server (10) and performs corresponding processes on the basis of the information provided from the DB server (10); a client terminal (30) which can access the middlewares (20) and includes the dedicated browser (32) for performing the processes; and a file server (40) which manages a plurality of application program files stored therein and provides corresponding application programs to the browser (32) in accordance with the instruction from the middlewares (20).

WO 02/01377 A1

DISTRIBUTED PROCESSING SYSTEM AND METHOD FOR APPLICATION PROGRAMS

Technical Field

The present invention relates to a system and method for application programs, more specifically to a distributed processing system and method for application programs which makes it possible to perform general processes such as user authentication, application program download/update, etc. via a dedicated browser when a user wants to execute desired application programs provided on the World Wide Web through the Internet.

Background Art

In the conventional office work using computers, a user had to purchase software programs necessary for the work and install into his or her own computer system. In this case, it was inconvenient to install all the software programs to be used into a computer system separately and to keep up the programs (such as program re-installation and upgrade, etc).

Nowadays, a variety of software companies develop and sell office work/business software programs in packages, however, all the packages being sold are not associated with the form of the World Wide Web.

Software programs associated with the Web are used from time to time in the computerization of specific massive groups, but there are no Web-associated programs for the work of medium/small-sized groups. In

addition, it is difficult to input/output, analyze and check enormous client data because client data is generally managed by a central system in the software programs associated with the Web at the computerization of specific massive groups.

Alternatively, in the ASP (Application Service Provider), the load on a browser is reduced since pages are created by a Web server, and failures in the network can be minimized by decreasing the need of mutual communications between the browser and the server. Further, security is enhanced by blocking data or contents which are not supposed to be transmitted to the browser. Nevertheless, such attempts result in the over-load on the server and investment of a lot of resources for database management due to the concentration of all sorts of data and contents into the server. Such over-load increases proportionately to the number of clients in service, which often causes installation of additional devices on the server side.

In addition, small-sized companies or private business owners have difficulties in using the ASP due to the cost thereof or the like because the ASP provides mainly for big or medium-sized companies.

As an example of prior art in this field, there is disclosed "A System and Method for Acquiring Remote Programs for Performing a Task" in the PCT Publication of International Publication Number WO 99/22332. According to such system, a user can easily and securely select, configure and access the unique computing resources and applications required to perform a given task. However, the system has all the application programs

including object request broker (ORB) and data in a web server. Accordingly, not only the speed of access to the server decreases due to over-load on the web but also there still exists possible hacking or server failure.

Disclosure of Invention

Therefore, the present invention is designed to overcome such problems, and it is an object of the present invention to provide a system and method for application programs which makes it possible to perform, efficiently in optimum conditions, general processes such as user authentication, authorization to use application programs, application program download/update, etc. when the application programs are provided on line through the Internet.

It is another object of the invention to provide a system and method for application programs that is capable of reducing loads on respective resources and preventing security problems such as data leakage from hacking, by storing all the task-related data prepared in the client terminal by a user into task DB on the client terminal side.

It is yet another object of the invention to provide a system and method for application programs including a dedicated browser for performing above-described various processes.

It is still another object of the invention to provide a system and method for application programs that can operate in an off-line condition.

In order to accomplish the above objects, the invention provides a

system for application programs, the system comprising a DB server for managing a DB which has a variety of basic information to perform various processes stored therein; middlewares for communicating with the DB server and performing corresponding processes based on the information from the DB server; a client terminal which can access the middlewares and includes a dedicated browser to perform the processes; and a file server for storing and managing a plurality of application program files and providing corresponding application programs to the browser according to the instructions from the middlewares.

Brief Description of the Drawings

The above and other objects and features of the instant invention will become apparent from the following description of preferred embodiments taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a schematic configuration diagram of a system for application programs according to the present invention;

Fig. 2A is a block diagram of a user authentication process through of the system shown in Fig. 1;

Fig. 2B is a block diagram of an authorization process to use application programs through the system shown in Fig. 1;

Fig. 2C is a block diagram of an application program download/update process through the system shown in Fig. 1;

Fig. 2D shows block diagrams of load-balancing process and fail-over process of the system shown in Fig. 1;

Fig. 3A and Fig. 3B are flow charts of a user authentication process according to the present invention;

Fig. 3C is a flow chart of an application program registration process according to the present invention;

Fig. 3D is a flow chart of an application program update process according to the present invention;

Fig. 3E is a flow chart of an application program execution process according to the present invention; and

Fig. 4 is a block diagram of a manual authentication process of the system according to the present invention in an off-line condition.

Best Mode for Carrying out the Invention

Hereinafter, the present invention will be described with reference to the drawings.

The present system consists of a distributed processing-based 3-hierarchy structure, and comprises a DB server 10, middlewares 20, a client terminal 30 and a file server 40. The DB server 10 manages a DB 11 having various basic information stored therein, such as information related to member authentication and registration of users, information related to application programs, information related to application program download/update, information related to data backup, etc, form performing processes of the present system and comprises such as MS NT 4.0/2000 series and an MS SQL server. The middlewares 20 communicate with the DB server 10 and perform various processes on the basis of various

information of the DB 11 received from the DB server 10. The client terminal 30 makes it possible to access the middlewares on-line or off-line by the user and to perform all user-wanted tasks in itself. Finally, the file server 40 stores and manages various application program files, and provides corresponding application programs to the client terminal 30 according to the instructions from the middlewares 20.

The client terminal 30 includes a dedicated browser 32, the key component of the present invention. The dedicated browser 32 has all functions of any means used to access the Internet such as Internet Explorer of Microsoft or Netscape, and, for example, provides options that make it possible to use the environment of the Internet Explorer without changing the environment. The browser 32 includes a component 36 comprising a socket, a member administrator and a balance broker. The component 36, the middlewares 20 and the DB server 10 communicate each other so as to perform various processes which will be described in detail below. The component 36 is connected to the application program file server 40 through a file transfer protocol (FTP) module 38. The function of the balance broker will be described in detail below, in conjunction with a load balancing process and a fail-over process.

The browser 32 can implement multi-browsing and automatic memorization of opened URL. With such multi-browsing function, a new Internet site can be opened within the browser which is being executed already without starting the browser again when opening a new window or inputting a new Internet site address. With such function of automatic

memorization of URL, it is possible to transfer to the already opened page without opening a new window when Internet addresses for several pages are inputted. Therefore, users can enjoy surfing the Internet with such multi-browsing function.

The client terminal 30 includes a task data DB 34 for storing data of the task carried out in the client terminal 30 by the user. The task data DB 34 is arranged in the client terminal 30 as such in order to perform quick and efficient input/output, check, and analysis of the data made by the user. Accordingly, it is possible to prevent security problems such as leakage of user data by storing the data made by the user into the task data DB 34 in the client terminal 30. Also, it is possible to store the data made by the user as backup into specified positions according to respective users in a backup file server (not shown) in accordance with the instructions from the middlewares if the user wants to store the data as backup.

The middlewares 20 each includes a component comprising a socket and a license administrator. With such configuration, each middleware 20 communicates with the component 36 of the dedicated browser 32, and manages general processes of the present invention, such as a membership authentication and registration process when a user accesses through the browser 32, an authorization process to use application programs by the user, an application program download/update process, load-balancing and fail-over processes, a data back-up process, a manual authentication process for the user in an off-line condition, etc.

Now, each process performed by the system according to the

present invention configured as Fig. 1 will be described in detail, with reference to the Figs. 2A to 2D, Figs. 3A to 3E, and Fig. 4. It should be noted that the Figs. 2A to 2D are block diagrams of several processes among a plurality of processes according to the invention in order to help readers understand the present invention, and that the flows of these and other processes are referred to Figs. 3A to 3E and Fig. 4.

Figs. 2A to 2D show a user authentication process, an authorization process to use application programs, an application program download/update process, and load-balancing and fail-over processes, respectively.

First, in a user authentication process as shown in Fig. 2A, a user inputs ID and password at log-in, and executes a dedicated browser 32 in a client terminal 30. Subsequently, a member administrator in a component 36 (see Fig. 1) of the browser 32 transmits user information (that is, user ID and password) to a license administrator in each middleware 20 so as to request authentication of the user. Then, the license administrator in each middleware 20 requests a DB server 10 to check on the user. In response to such request, the DB server 10 checks DB 11 (see Fig. 1) in the DB server 10, and notifies the license administrator of the check result. Accordingly, the license administrator notifies the member administrator whether the user has been authenticated or not, on the basis of the result, thereby completing the user authentication process.

Next, in an authorization process to use application programs as shown in Fig. 2B, the user requests to use specific application programs on

a screen menu of the dedicated browser 32 in the client terminal 30, then the member administrator transmits the user information and application program codes to the license administrator in each middleware 20 so as to request authentication. Accordingly, the license administrator transmits the user information (such as ID and password) and the application program codes to the DB server 10 so as to check the user information and the application program codes, and to transmit a list of the application programs requested by the user. In response to such request, the DB server 10 checks and searches the DB 11 in the DB server 10, and notifies the license administrator of the result. Then, the license administrator notifies the member administrator of such result, thereby completing the authorization process to use the application programs. In this authentication process, it is checked generally for important matters of authentication such as whether the user is a charged-member, whether the user is in a complimentary period, whether the user has paid the bill or not, etc.

Fig. 2C shows an application program download/update process. In this process, the user requests for download/update of the application programs shown on the screen menu of the dedicated browser 32 in the client terminal 30, then the member administrator transmits the user information and the application program codes to the license administrator in each middleware 20. Subsequently, the license administrator transmits the user information (such as ID and password) and the application program codes to the DB server 10, so as to check whether the user has

authorization to download/update the application programs. So, the DB server 10 searches and checks the DB 11 and notifies the check result to the license administrator, and then the license administrator notifies the member administrator of the result. If the member administrator is notified that the user has authorization to download/update the application programs, the member administrator requests again the license administrator for IP, account and password. Next, if the member administrator receives the IP, account and password, the member administrator requests an application program file server 40 for corresponding application programs through a FTP module 38. Where, the IP is an address for connecting to the FTP module 38, thereby enabling to connect to the application program file server 40. Then, the application program file server 40 transmits files corresponding to the requested application programs to the dedicated browser 32 in the client terminal 30 via the FTP module 38, thereby completing the application program download/update process.

The load-balancing process and fail-over process for performing various processes efficiently in the system of the present invention are illustrated in Fig. 2D. In the load-balancing process shown in the Fig. 2D, the user accesses the dedicated browser 32 in the client terminal 30, then the balance broker 22 checks the number of current users connected to each of the plurality of middlewares 20-1, 20-2, ……, and 20-n. In accordance with the check result, the balance broker 22 connects automatically the user to a middleware 20-n with the smallest number of

users connected thereto, as indicated by the arrow of a solid line. In the fail-over process, when a user accesses the dedicated browser 32 in the client terminal 30, the balance broker 22 connects automatically the user to middlewares 20-1 and 20-2 operating in normal state among the plurality of middlewares 20-1, 20-2, ……, and 20-n, by avoiding the middleware 20-n which is in failure. With such load-balancing and fail-over function, the balance broker 22 in the dedicated browser 32 connects the user to a optimum middleware 20 which has least members or guests connected and operates normally, so that general logic procedures such as user authentication, authorization to use application programs, application program download/update, etc can be performed quickly and accurately through information exchange with the DB server 10. Therefore, it is possible not only to alleviate the over-load on the servers but also to make the resources in the middlewares 20 and file server 40 optimum states.

Hereinafter, the flows of the above processes of the system according to the present invention will be described in detail, with reference to Figs. 3A to 3E.

User Authentication Process

Fig. 3A shows a user authentication process for a new user. First, the user executes a dedicated browser 32 in the client terminal 30 in step S10. Then, a middleware 20 requests a DB server 10 to check whether the user has membership S20. Then, the DB server 10 searches and checks DB 11, if the user does not have membership, the middleware 20 requests

the user to input general information for acquiring membership such as private and/or company information S30. Next, the middleware 20 checks whether the user has inputted information normally S40. If the input information is normal, the user is requested to input information related to the application programs the user wants to use S50. Such input information is based on the various application programs shown on the application program menu on the browser 32. After the information is inputted, it is determined whether the input information will be stored or not S60. If the response is 'No', this process is ended. If the response is 'Yes', the user is given ID and password is encoded S70, and then the ID and encoded password are stored in the DB 11 in step S80. As such, the user authentication process is completed, and step A in Fig. 3B follows next. The step A also follows the step S20 if the user has membership.

Fig. 3B shows a user authentication process performed in case of the registered members. First, the user is requested to input ID and password S90. After inputting the ID and password, the user is subjected to user authentication S100. The operation process for the components of the present system related to this user authentication was already described with reference to Fig. 2A. Next, it is checked whether the user authentication is performed normally S110. If the result of the user authentication is not normal, the user returns to the step S90 to input user ID and password again, and repeats the steps until the user authentication is performed normally. If the user authentication is performed normally, it is checked whether the user has applied for the use of application

programs before, S120. If the use has applied for the use of application programs before, the step B of application program registration shown in Fig. 3C follows. On the other hand, if the user has not applied for the use of application programs before, the user is caused to apply for the use of application programs S130. After that, it is determined whether such result will be stored S140. If the response is 'Yes', the result is stored in the DB 11 in step S150, and the step B of application program registration follows next. On the contrary, if the response is 'No', the procedure ends.

Application Program Registration Process

Fig. 3C shows a process for downloading the user-wanted application programs and registering the downloaded application programs into the application program menu of the dedicated browser 32. First, it is checked whether the user has authorization to use the application programs requested by the user S160. After such check, if the user has no authorization to use, the step of authorization to use the application programs follows so that the user is given authorization to use the application programs. On the contrary, if the user is checked as having authorization to use the application programs, lists of the application programs are requested S180. The above request for the lists of the application programs is performed in such a way that the member administrator of the browser 32 requests the license administrator of the middleware 20 for the lists of the application programs requested by the user in the above-described user authentication process. Next, the number of application programs requested by the user is counted S190. Then, the

steps included in a larger block shown in Fig. 3E are repeated in accordance with the number of counted application programs. In such repetition, first, the download of the requested application programs is performed S200. Then, it is checked whether the download of the requested application programs is performed normally S210. If the result of the download is not normal, the download of the requested application programs is performed again S205. On the contrary, if the result of the download is normal, it is checked whether the downloaded application programs have been registered in the application program menu of the browser S220. If registered, step C of the application program update in Fig. 3D follows next. On the other hand, if not registered, the downloaded application programs are registered in the application program menu of the browser S230, and then step C of the application program update in Fig. 3D also follows next. Where, the process for registering application programs in the application program menu also creates icons of execution files for the application programs on the screen of the browser, so that the user can execute the application programs directly on the browser 32.

Application Program Update Process

Fig. 3D shows an application program update process. As described above, after the application program registration process is completed, the user can execute the application programs directly S240. In this case, it is checked whether the version of an application program to be executed by the user is the latest S250. If the version is the latest, step D of application program execution shown in Fig. 3E follows next, so that the user can

perform his/her task with the latest version of the application program. If the version of the application program to be executed by the user is not the latest, it is checked whether the user has authorization to update the application program S260. If the user has no authorization to update the application program, the user is made to request authorization to update S270. If the user has authorization to update, the user can download and install the latest version of the application program S290, thereby completing the application program update. Then, step D of Fig. 3E in which the user does his/her tasks using the application program follows next.

Application Program Execution Process

As shown in Fig. 3E, now the user executes an application program so as to do his/her tasks. The authorization to use the application program is performed in step S300. If the authentication is normal, the user ultimately executes the application program so as to do his/her tasks S330. On the contrary, if the authentication is not normal, step S320 follows next, and the process ends with only browser open.

It should be noted that, immediately after the completion of the application program registration process, the application program execution process of Fig. 3E might follow directly. Only when the version of the application program stored in the application program file server 40 is later than that of the application program stored and used currently in the client terminal 30, the application program update process is performed.

In addition, it should be noted that, although several authentication steps are performed in each process, it is possible to change, modify, add or subtract properly such authentication steps in accordance with the operator's intention, and it is not intended to limit the present invention to the flows of the processes described with the preferred embodiments.

Fig. 4 shows a manual authorization process to use the present system in off-line condition, when it is difficult to access the present system due to failure of the middlewares 20. For such manual authentication, the present system includes a local authentication module, which comprises a serial number issue module at the dedicated browser 32 and an authentication number generator module at the middlewares 20. First, in case of middleware failure, the user checks a serial number on the dedicated browser, and notifies the customer service team of the present system of the serial number by phone so as to request an authentication key. Then, the customer service team retrieves the authentication module to input the serial number of the user, confirms and notifies the user of an authentication key. Accordingly, the user can log in the dedicated browser with the authentication key so as to execute a desired application program. Where, the authentication key generated in accordance with the serial number cannot apply to other computer systems. That is, the authentication key cannot apply to other computer systems with different serial number because the serial number is generated by referring to, for example, a serial number on a hard disk of a user PC. In addition, the authentication key given once cannot be used permanently. That is, the

authentication key can be used only for a predetermined time, for example one hour, after the key is given, since the authentication key is generated while checking the date and time of the user PC when the user executes the serial number issue module. The time limit is set as such because the failure of the middlewares does not last long and quick recovery by spot-side is made immediately after occurrence of the failure. Furthermore, the authentication key does not expose to outside by configuring and applying an encode algorithm internally.

As described above, the present invention makes it possible to perform general processes such as user authentication, authorization to use application programs, application program download/update, etc. efficiently in optimum condition, through a dedicated browser. In addition, all task data made by the user is stored in a task DB on the client terminal side other than a central server, thereby alleviating load on respective resources and preventing security problems such as data leakage from hacking.

Although the preferred embodiments of the invention have been disclosed for illustrative purposes, those skilled in the art will appreciate that various modifications, additions and substitutions are possible, without departing from the scope and spirit of the invention as disclosed in the accompanying claims.

Claims

1. A distributed processing system for application programs comprising;

a DB server for managing a DB in which a variety of basic information to perform a plurality of processes is stored;

a plurality of middlewares for communicating with the DB server and performing corresponding processes on the basis of information from the DB server;

a dedicated browser stored in a client terminal for accessing each middleware and performing the processes; and

a file server for managing a plurality of application program files stored therein, and providing corresponding application programs to the dedicated browser in accordance with instructions from one of the middlewares.

2. The system as claimed in claim 1, wherein the dedicated browser includes a component comprising a socket for accessing the middlewares, a member administrator and a balance broker.

3. The system as claimed in claim 1, wherein the dedicated browser functions as World Wide Web browser and has a multi-browsing function.

4. The system as claimed in claim 1, the middlewares include a component comprising a socket for accessing the dedicated browser and a license administrator.

5. The system as claimed in claim 1 or 2, wherein the dedicated browser performs a load-balancing function for automatically connecting a user to a middleware with the smallest number of users connected thereto among the plurality of middlewares while communicating with the middlewares and a fail-over function for automatically connecting the user to middlewares in normal operation by excluding middlewares in failure state among the plurality of middlewares.

6. The system as claimed in claim 1, wherein the file server accesses the dedicated browser via a file transfer protocol (FTP).

7. The system as claimed in claim 1, wherein the client terminal further comprises a task data DB for storing data prepared on the dedicated browser by the user.

8. The system as claimed in claim 1, further comprising a local authentication module for performing a manual authentication process in off-line condition.

9. The system as claimed in claim 8, the local authentication

module comprises a serial number issue module at the dedicated browser and an authentication key generator module at the middlewares.

10. The system as claimed in claim 9, wherein the serial number is generated by referring to a serial number of a user PC.

11. A method for application programs for performing general steps including a user authentication step, an authorization step to use application programs, an application program download/update step, a load-balancing/fail-over step, a manual authentication step in off-line condition, etc through a system comprising a DB server for managing a DB in which a variety of basic information to perform a plurality of processes is stored; a plurality of middlewares for communicating with the DB server and performing corresponding processes on the basis of information from the DB server; a dedicated browser stored in a client terminal for accessing each middleware and performing the processes; and a file server for managing a plurality of application program files stored therein and providing corresponding application programs to the dedicated browser in accordance with instructions from one of the middlewares.

12. The method for application programs as claimed in claim 11, wherein the user authentication step comprises:

a step in which a user inputs user information at log-in and

executes the dedicated browser in the client terminal;

a step in which the dedicated browser transmits the user information to the middlewares so as to request user authentication;

a step in which the middlewares request the DB server to check on the user; and

a step in which the DB server checks the DB in response to such request and notifies the middlewares of such check result, and the middlewares notify the dedicated browser whether the user is authenticated or not on the basis of the result.

13. The method for application programs as claimed in claim 11, wherein the authorization step to use the application programs comprises:

a step in which the user requests to use application programs on a screen menu of the dedicated browser;

a step in which the dedicated browser transmits the user information and application program codes to the middlewares so as to request authentication to use the application programs;

a step in which the middlewares transmit the user information and the application program codes to the DB server so as to request to check the DB for the user information and the application program codes, and a list of the application programs requested by the user;

a step in which the DB server checks the DB and transmits the check result to the middlewares; and

a step in which the middlewares notify the dedicated browser of such result.

14. The method for application programs as claimed in claim 11, wherein the application program download/update step comprises:

a step in which the user requests for download/update of application programs shown on a screen menu of the dedicated browser;

a step in which the dedicated browser transmits user information and application program codes to the middlewares;

a step in which the middlewares transmits the user information and the application program codes to the DB server so as to request to check whether the user has authorization for the application program download/update;

a step in which the DB server checks the DB whether the user has the authorization to download/update the application programs and notify the middlewares of the check result, and the middlewares transmit the check result again to the dedicated browser;

a step in which the dedicated browser requests the middlewares for an IP, an account and a password, if the dedicated browser is notified that the user has authorization to download/update the application programs;

a step in which the dedicated browser requests the application program file server for corresponding application programs via the FTP (file transfer protocol) module, after receiving the IP, account and

password; and

a step in which the application program file server transmits the requested application programs to the dedicated browser via the FTP module.

15. The method for application programs as claimed in claim 11, wherein the load-balancing step comprises:

a step in which the user accesses the dedicated browser;

a step in which the dedicated browser checks the number of current users connected to each of the plurality of middlewares; and

a step in which the dedicated browser connects automatically the user to a middleware with the smallest number of users connected thereto among the plurality of middlewares.

16. The method for application programs as claimed in claim 11, wherein the fail-over step comprises:

a step in which the user accesses the dedicated browser;

a step in which the dedicated browser checks which middleware(s) is(are) in failure among the plurality of middlewares; and

a step in which the dedicated browser connects automatically the user to middlewares in normal operation among the plurality of middlewares by avoiding the middleware(s) in failure.

17. The method for application programs as claimed in claim 11,

wherein the manual authentication step in off-line condition comprises:

a step in which the user inputs and transmits a serial number of the dedicated browser to an operator so as to request an authentication key;

a step in which the operator retrieves an authentication module to input the serial number of the user, and confirms and notifies the user of the authentication key; and

a step in which the user logs in the dedicated browser with the authentication key.

18. A computer-readable recording medium for storing code programs that can be transmitted to a client terminal through communications network by download or upload or can be accessed by a computer via disk drives, the programs are encoded so as to perform general processes including Ⅰ) a user authentication process, Ⅱ) an authorization process to use application programs, Ⅲ) an application program download/update process, Ⅳ) a load-balancing process, Ⅴ) a fail-over process, Ⅵ) a manual authentication process in off-line, through a system comprising a DB server for managing a DB in which a variety of basic information to perform a plurality of processes is stored; a plurality of middlewares for communicating with the DB server and performing corresponding processes on the basis of information from the DB server; a dedicated browser stored in a client terminal for accessing the middlewares and performing the processes; and a file server for managing a plurality of application program files stored therein and

providing corresponding application programs to the dedicated browser in accordance with instructions from one of the middlewares.

19. The computer-readable recording medium as claimed in claim 18, wherein the process Ⅰ) comprises:

a process in which a user inputs user information at log-in and executes the dedicated browser in the client terminal;

a process in which the dedicated browser transmits the user information to the middlewares so as to request user authentication;

a process in which the middlewares request the DB server to check on the user; and

a process in which the DB server checks the DB in response to such request and notifies the middlewares of such check result, and the middlewares notify the dedicated browser whether the user is authenticated or not on the basis of the result.

20. The computer-readable recording medium as claimed in claim 18, wherein the process Ⅱ) comprises:

a process in which the user requests to use application programs on a screen menu of the dedicated browser;

a process in which the dedicated browser transmits the user information and application program codes to the middlewares so as to request authentication to use the application programs;

a process in which the middlewares transmit the user information

and the application program codes to the DB server so as to request to check the DB for the user information and the application program codes, and a list of the application programs requested by the user;

a process in which the DB server checks the DB and transmits the check result to the middlewares; and

a process in which the middlewares notify the dedicated browser of such result.

21. The computer-readable recording medium as claimed in claim 18, wherein the process Ⅲ) comprises:

a process in which the user requests for download/update of application programs shown on a screen menu of the dedicated browser;

a process in which the dedicated browser transmits user information and application program codes to the middlewares;

a process in which the middlewares transmits the user information and the application program codes to the DB server so as to request to check whether the user has authorization to download/update the application programs;

a process in which the DB server checks the DB whether the user has the authorization to download/update the application programs and notify the middlewares of the check result, and the middlewares transmit the check result again to the dedicated browser;

a process in which the dedicated browser requests the middlewares for an IP, an account and a password, if the dedicated browser is notified

that the user has authorization to download/update the application programs;

a process in which the dedicated browser requests the application program file server for corresponding application programs via the FTP (file transfer protocol) module, after receiving the IP, account and password; and

a process in which the application program file server transmits the requested application programs to the dedicated browser via the FTP module.

22. The computer-readable recording medium as claimed in claim 18, wherein the process Ⅳ) comprises:

a process in which the user accesses the dedicated browser;

a process in which the dedicated browser checks the number of current users connected to each of the plurality of middlewares; and

a process in which the dedicated browser connects automatically the user to a middleware with the smallest number of users connected thereto among the plurality of middlewares.

23. The computer-readable recording medium as claimed in claim 18, wherein the process Ⅴ) comprises:

a process in which the user accesses the dedicated browser;

a process in which the dedicated browser checks which middleware(s) is(are) in failure among the plurality of middlewares; and

a process in which the dedicated browser connects automatically the user to middlewares in normal operation among the plurality of middlewares by avoiding the middleware(s) in failure.

24. The computer-readable recording medium as claimed in claim 18, wherein the process VI) comprises:

a process in which the user inputs and transmits a serial number of the dedicated browser to an operator so as to request an authentication key;

a process in which the operator retrieves an authentication module to input the serial number of the user, and confirms and notifies the user of the authentication key; and

a process in which the user logs in the dedicated browser with the authentication key.

FIG. 1

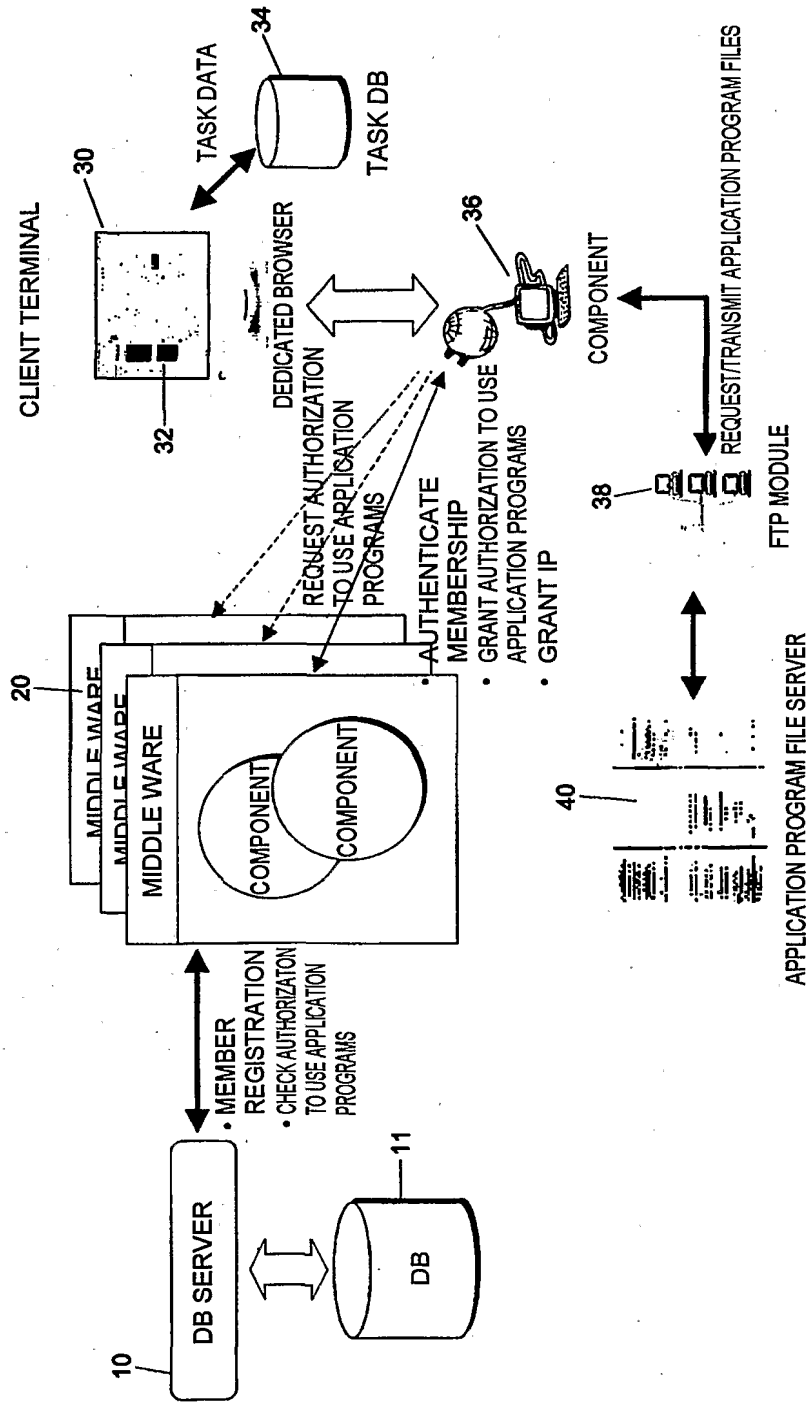

CLIENT TERMINAL
30
32
TASK DATA
34
TASK DB
DEDICATED BROWSER
36
COMPONENT
REQUEST/TRANSMIT APPLICATION PROGRAM FILES
38
FTP MODULE
REQUEST AUTHORIZATION TO USE APPLICATION PROGRAMS
AUTHENTICATE MEMBERSHIP
• GRANT AUTHORIZATION TO USE APPLICATION PROGRAMS
• GRANT IP
20
MIDDLE WARE
MIDDLE WARE
MIDDLE WARE
COMPONENT
COMPONENT
40
APPLICATION PROGRAM FILE SERVER
• MEMBER REGISTRATION
• CHECK AUTHORIZATON TO USE APPLICATION PROGRAMS
DB SERVER
10
DB
11

FIG. 2A

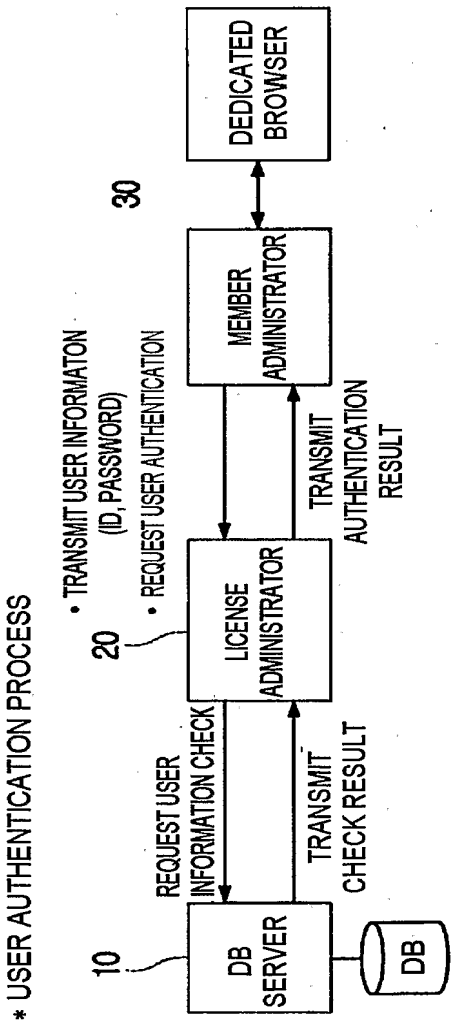

* USER AUTHENTICATION PROCESS
10
DB SERVER
DB
REQUEST USER INFORMATION CHECK
TRANSMIT CHECK RESULT
20
LICENSE ADMINISTRATOR
• REQUEST USER AUTHENTICATION
• TRANSMIT USER INFORMATON (ID, PASSWORD)
TRANSMIT AUTHENTICATION RESULT
30
MEMBER ADMINISTRATOR
DEDICATED BROWSER

FIG. 2B

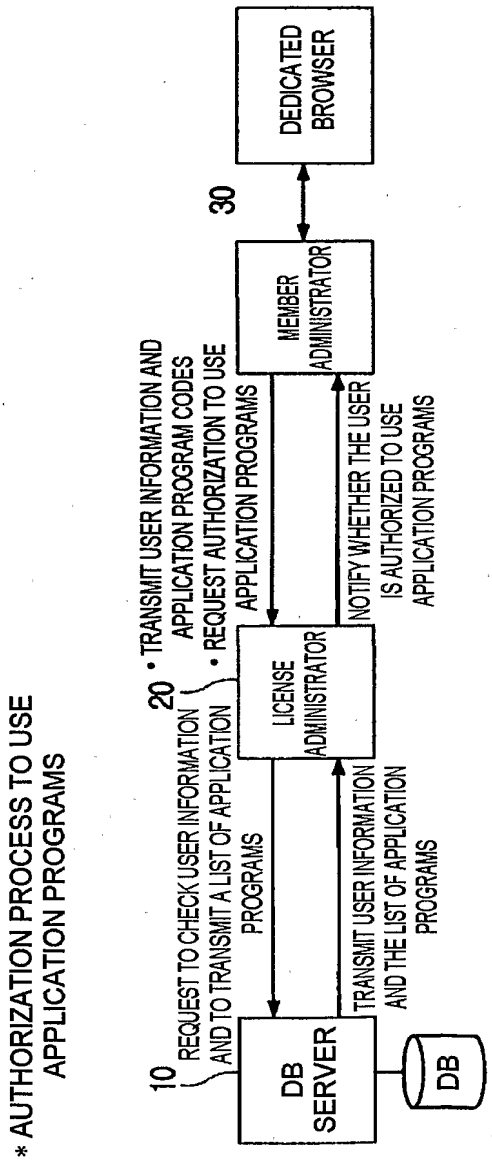

* AUTHORIZATION PROCESS TO USE APPLICATION PROGRAMS
DEDICATED BROWSER
30
MEMBER ADMINISTRATOR
• TRANSMIT USER INFORMATION AND APPLICATION PROGRAM CODES
• REQUEST AUTHORIZATION TO USE APPLICATION PROGRAMS
NOTIFY WHETHER THE USER IS AUTHORIZED TO USE APPLICATION PROGRAMS
20
LICENSE ADMINISTRATOR
REQUEST TO CHECK USER INFORMATION AND TO TRANSMIT A LIST OF APPLICATION PROGRAMS
TRANSMIT USER INFORMATION AND THE LIST OF APPLICATION PROGRAMS
10
DB SERVER
DB

FIG. 2C

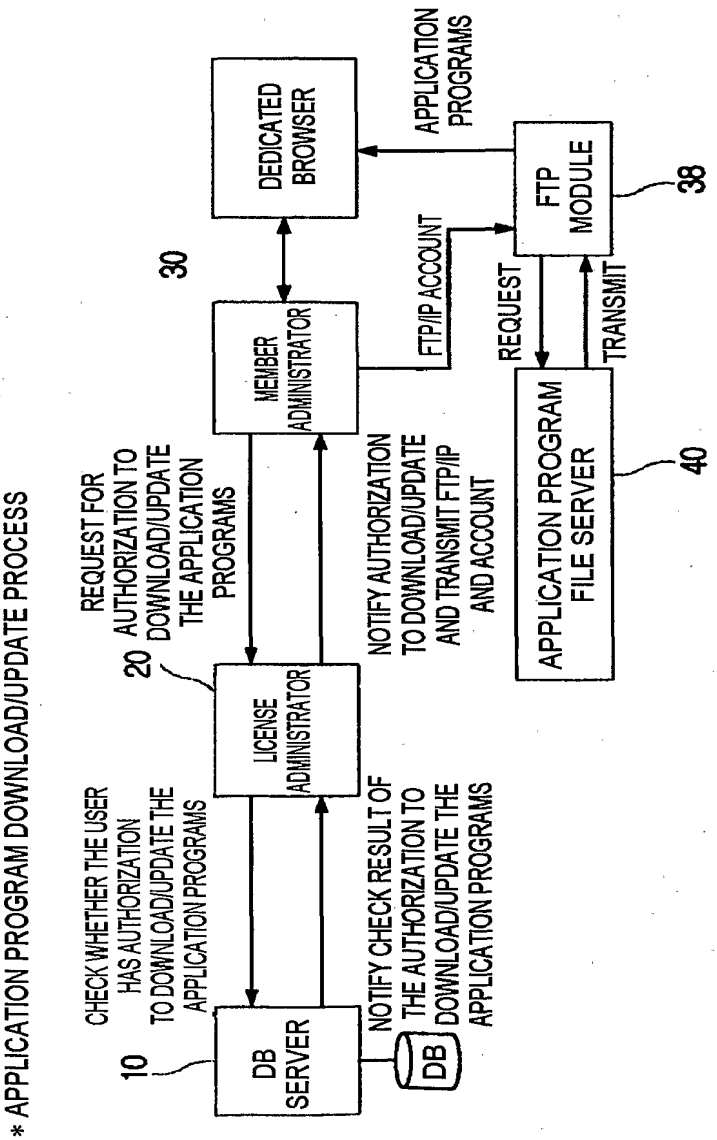

* APPLICATION PROGRAM DOWNLOAD/UPDATE PROCESS
DEDICATED BROWSER
APPLICATION PROGRAMS
FTP MODULE
38
30
MEMBER ADMINISTRATOR
FTP/IP ACCOUNT
REQUEST
TRANSMIT
APPLICATION PROGRAM FILE SERVER
40
REQUEST FOR AUTHORIZATION TO DOWNLOAD/UPDATE THE APPLICATION PROGRAMS
NOTIFY AUTHORIZATION TO DOWNLOAD/UPDATE AND TRANSMIT FTP/IP AND ACCOUNT
20
LICENSE ADMINISTRATOR
CHECK WHETHER THE USER HAS AUTHORIZATION TO DOWNLOAD/UPDATE THE APPLICATION PROGRAMS
NOTIFY CHECK RESULT OF THE AUTHORIZATION TO DOWNLOAD/UPDATE THE APPLICATION PROGRAMS
10
DB SERVER
DB

FIG. 2D

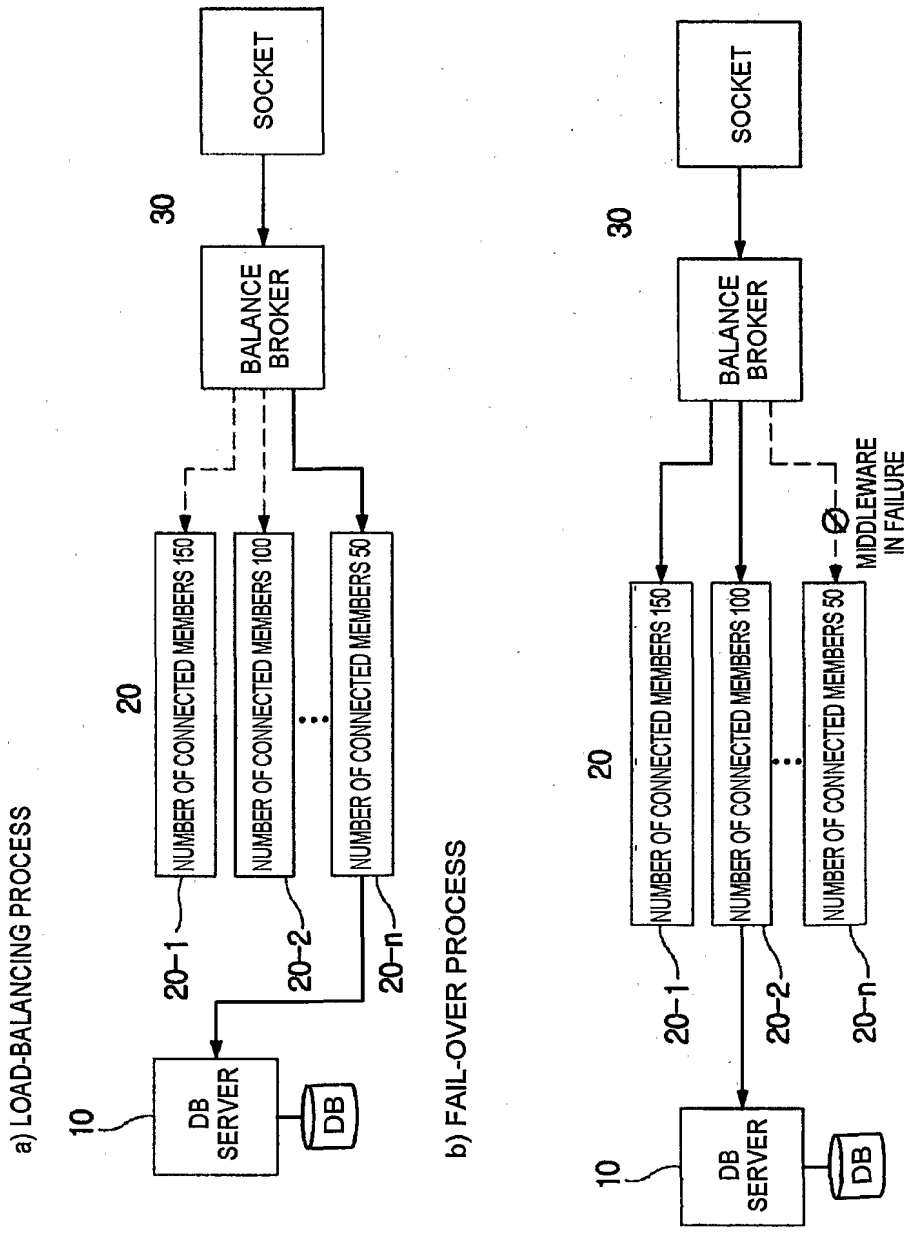

a) LOAD-BALANCING PROCESS
SOCKET
30
BALANCE BROKER
20
NUMBER OF CONNECTED MEMBERS 150
NUMBER OF CONNECTED MEMBERS 100
NUMBER OF CONNECTED MEMBERS 50
20-1
20-2
20-n
10
DB SERVER
DB
b) FAIL-OVER PROCESS
SOCKET
30
BALANCE BROKER
MIDDLEWARE IN FAILURE
20
NUMBER OF CONNECTED MEMBERS 150
NUMBER OF CONNECTED MEMBERS 100
NUMBER OF CONNECTED MEMBERS 50
20-1
20-2
20-n
10
DB SERVER
DB

FIG. 3A

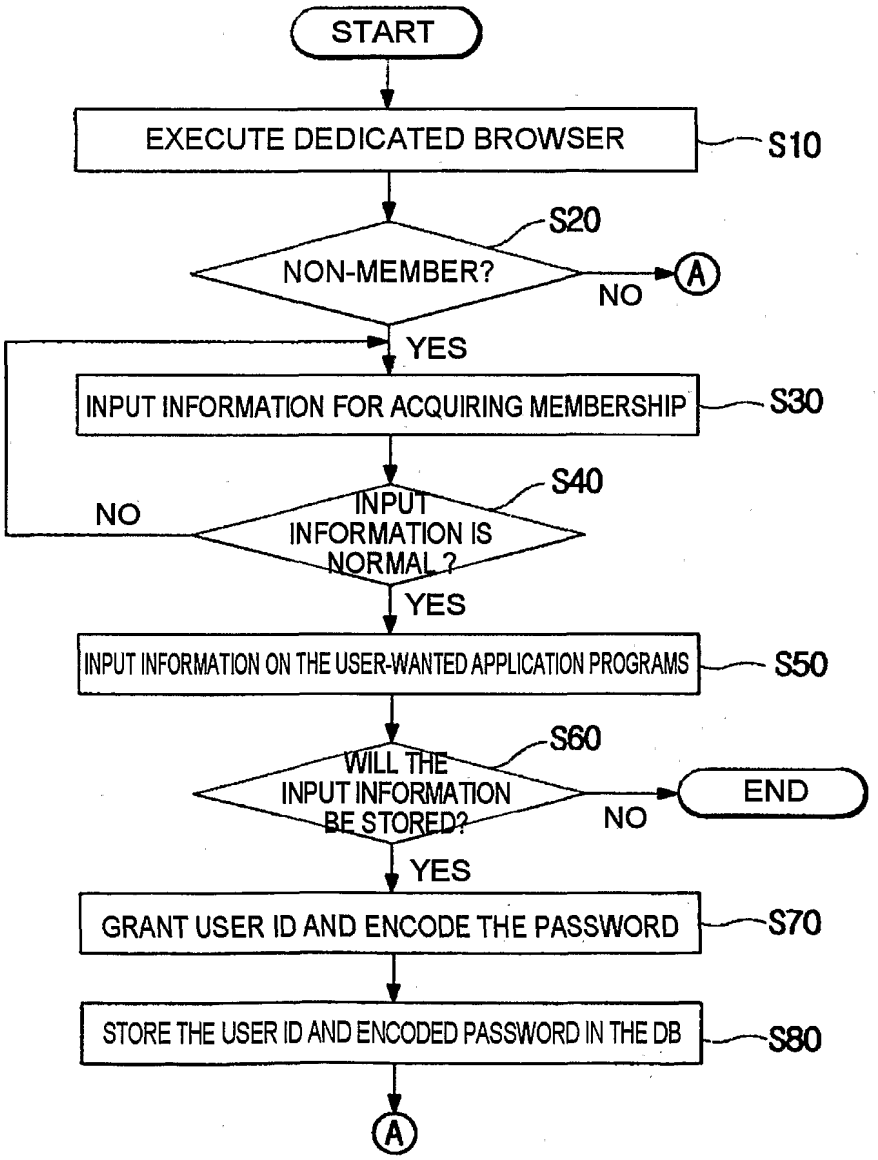

START
EXECUTE DEDICATED BROWSER
S10
S20
NON-MEMBER?
NO
A
YES
INPUT INFORMATION FOR ACQUIRING MEMBERSHIP
S30
S40
INPUT INFORMATION IS NORMAL ?
NO
YES
INPUT INFORMATION ON THE USER-WANTED APPLICATION PROGRAMS
S50
S60
WILL THE INPUT INFORMATION BE STORED?
END
NO
YES
GRANT USER ID AND ENCODE THE PASSWORD
S70
STORE THE USER ID AND ENCODED PASSWORD IN THE DB
S80
A

FIG. 3B

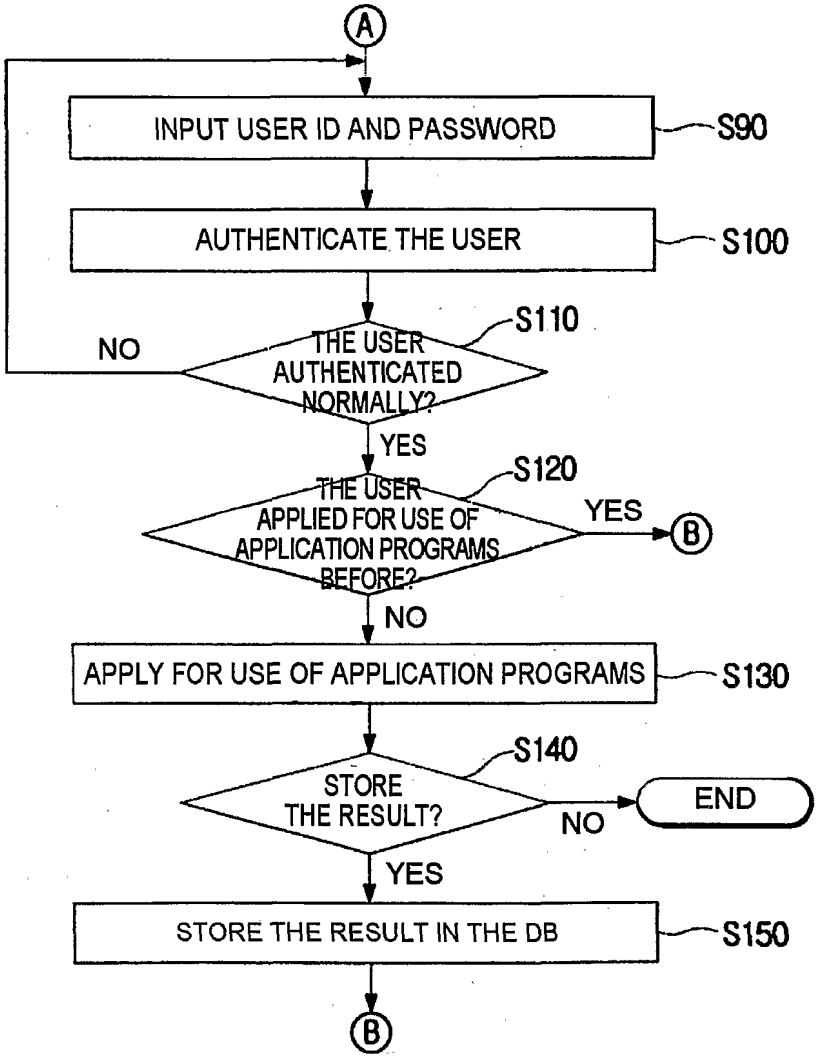

A
INPUT USER ID AND PASSWORD
S90
AUTHENTICATE THE USER
S100
S110
THE USER AUTHENTICATED NORMALLY?
NO
YES
S120
THE USER APPLIED FOR USE OF APPLICATION PROGRAMS BEFORE?
YES
B
NO
APPLY FOR USE OF APPLICATION PROGRAMS
S130
S140
STORE THE RESULT?
NO
END
YES
STORE THE RESULT IN THE DB
S150
B

FIG. 3C

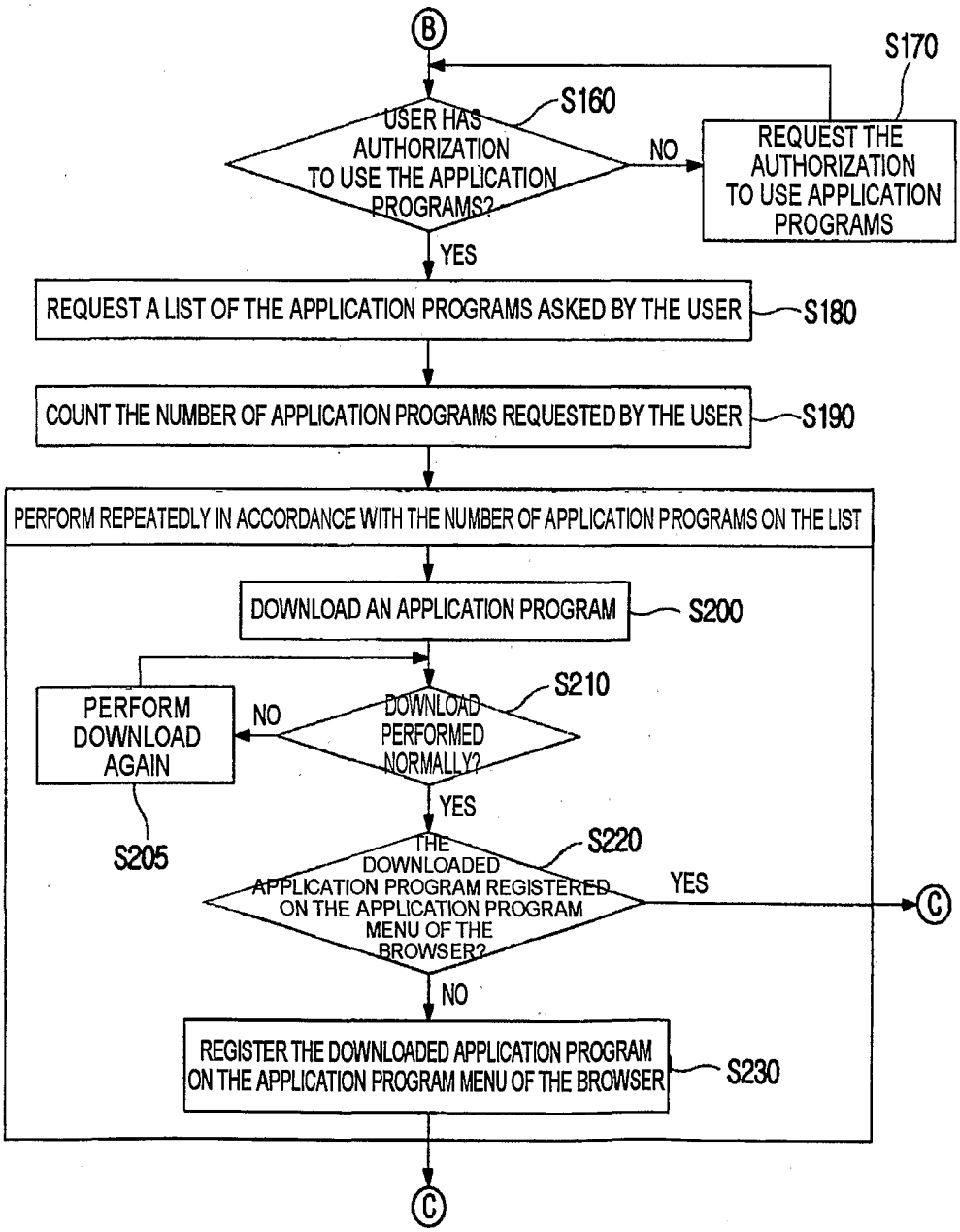

B
S170
S160
USER HAS AUTHORIZATION TO USE THE APPLICATION PROGRAMS?
NO
REQUEST THE AUTHORIZATION TO USE APPLICATION PROGRAMS
YES
REQUEST A LIST OF THE APPLICATION PROGRAMS ASKED BY THE USER
S180
COUNT THE NUMBER OF APPLICATION PROGRAMS REQUESTED BY THE USER
S190
PERFORM REPEATEDLY IN ACCORDANCE WITH THE NUMBER OF APPLICATION PROGRAMS ON THE LIST
DOWNLOAD AN APPLICATION PROGRAM
S200
S210
PERFORM DOWNLOAD AGAIN
NO
DOWNLOAD PERFORMED NORMALLY?
YES
S205
S220
THE DOWNLOADED APPLICATION PROGRAM REGISTERED ON THE APPLICATION PROGRAM MENU OF THE BROWSER?
YES
C
NO
REGISTER THE DOWNLOADED APPLICATION PROGRAM ON THE APPLICATION PROGRAM MENU OF THE BROWSER
S230
C

FIG. 3D

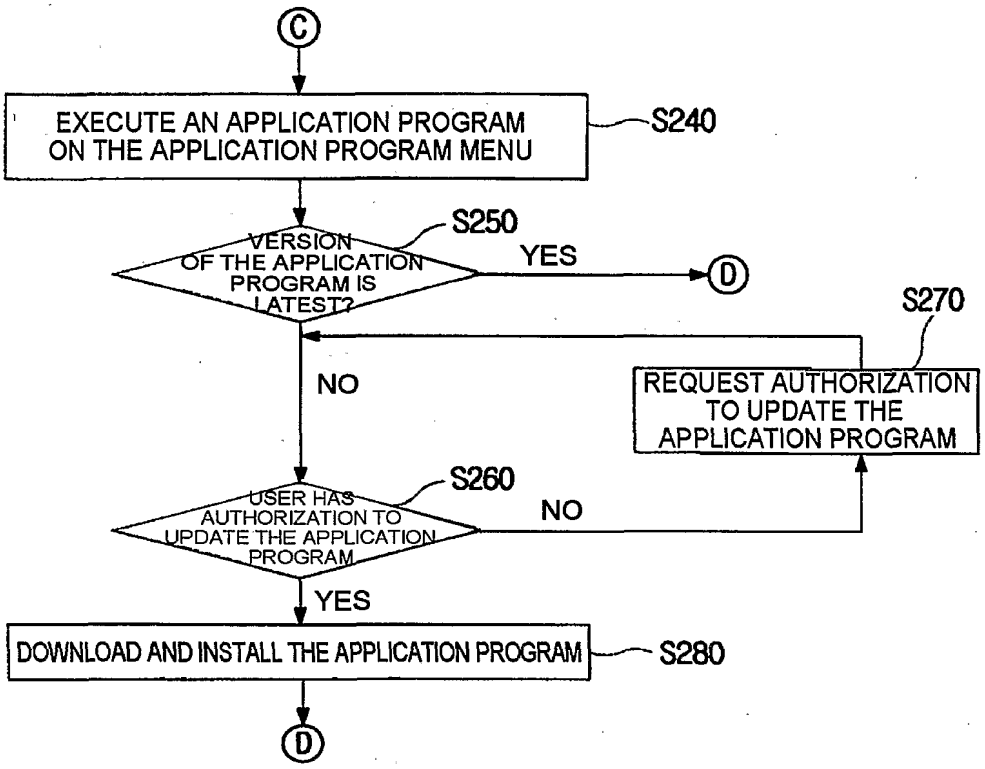

C
EXECUTE AN APPLICATION PROGRAM ON THE APPLICATION PROGRAM MENU
S240
VERSION OF THE APPLICATION PROGRAM IS LATEST?
S250
YES
D
S270
REQUEST AUTHORIZATION TO UPDATE THE APPLICATION PROGRAM
NO
USER HAS AUTHORIZATION TO UPDATE THE APPLICATION PROGRAM
S260
NO
YES
DOWNLOAD AND INSTALL THE APPLICATION PROGRAM
S280
D

FIG. 3E

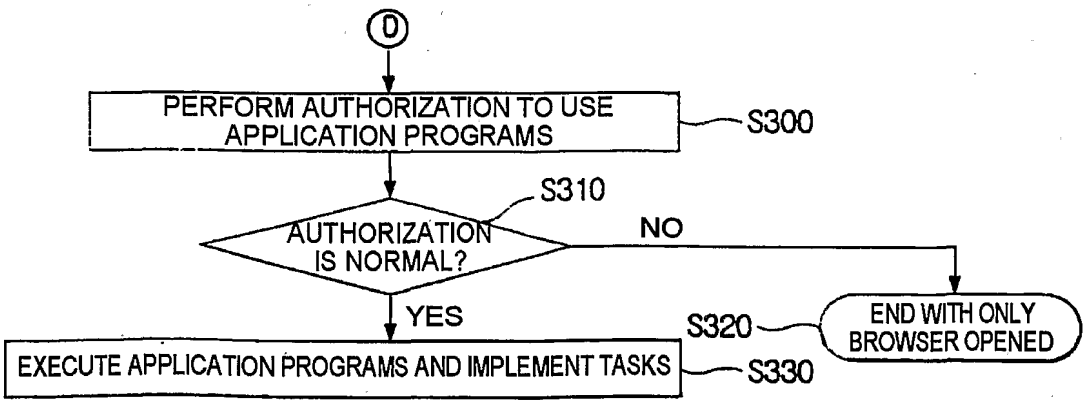
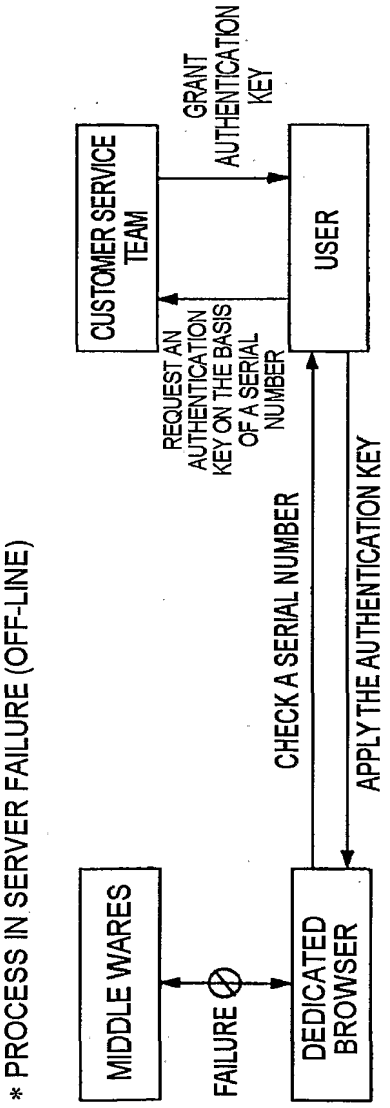

D
PERFORM AUTHORIZATION TO USE APPLICATION PROGRAMS
S300
AUTHORIZATION IS NORMAL?
S310
NO
YES
END WITH ONLY BROWSER OPENED
S320
EXECUTE APPLICATION PROGRAMS AND IMPLEMENT TASKS
S330

FIG. 4


* PROCESS IN SERVER FAILURE (OFF-LINE)
MIDDLE WARES
FAILURE
DEDICATED BROWSER
CHECK A SERIAL NUMBER
APPLY THE AUTHENTICATION KEY
REQUEST AN AUTHENTICATION KEY ON THE BASIS OF A SERIAL NUMBER
CUSTOMER SERVICE TEAM
GRANT AUTHENTICATION KEY
USER

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR01/01058

A. CLASSIFICATION OF SUBJECT MATTER

IPC7 G06F 15/00

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

B. FIELDS SEARCHED

Minimun documentation searched (classification system followed by classification symbols)

G06F 15/00

Documentation searched other than minimun documentation to the extent that such documents are included in the fileds searched

KOREAN PATENTS AND APPLICATIONS FOR INVENTIONS SINCE 1975
KOREAN UTILITY MODELS AND APPLICATIONS FOR UTILITY MODELS SINCE 1975

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

HTTP://WWW.USPTO.GOV/
WPI, PAJ, IEEE/IEE ELECTRONIC LIBRARY(1998) 'GAME AND (STOCK OR FINANCE) AND (STUDY OR TEACH)'

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US-5892900 (InterTrust Technologies Corp.) April 6, 1999 *ABSTRACT*	1 ~ 24
A	US-5765153 (International Business Machines Corporation) June 9, 1998 *ABSTRACT*	1 ~ 24
A	US-5958009 (Hewlett-Packard Company) September 28, 1999 *ABSRACT*	1 ~ 24
A	US-6011918 (International Business Machines Corporation) January 4, 2000 *ABSTRACT*	1 ~ 24
P, A	US-6088796 (Cianfrocca) July 11, 2000 *ABSTRACT*	1 ~ 24
P, A	US-6094655 (International Business Machines Corporation) July 25, 2000 *ABSTRACT*	1 ~ 24
		1 ~ 24

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevence
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)
"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
"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
"X" document of particular relevence; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevence; 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
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 JULY 2001 (30.07.2001)	31 JULY 2001 (31.07.2001)

Name and mailing address of the ISA/KR
Korean Intellectual Property Office
Government Complex-Daejeon, Dunsan-dong, Seo-gu, Daejeon Metropolitan City 302-701, Republic of Korea
Facsimile No. 82-42-472-7140

Authorized officer
YANG, In Soo
Telephone No. 82-42-481-5782