(11) Application No. **AU 2003274867 C1**

**(19) AUSTRALIAN PATENT OFFICE**

(54) Title

Method and arrangement for use of shared resources in a network

(51)[6] International Patent Classification(s)

**G06F** 3/12 (2006.01) 20060101AFI2005100

G06F 3/12 8BMEP

PCT/SE2003/001663

(21) Application No: 2003274867 (22) Application Date: 2003.10.28

(87) WIPO No: WO04/038580

(30) Priority Data

| (31) Number | (32) Date | (33) Country |
|---|---|---|
| 0203165-6 | 2002.10.28 | SE |

(43) Publication Date : 2004.05.13

(43) Publication Journal Date : 2004.06.17

(71) Applicant(s)

Devlabs AB

(72) Inventor(s)

Berglin, Patrik

(74) Agent/Attorney

Ellis Verboeket Terry, c/ Spruson & Ferguson Level 35, St Martins Tower 31 Market Street, Sydney, NSW, 2000

(56) Related Art

WO 1999/038068 A1

GB 2331387 A

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 May 2004 (06.05.2004)

PCT

(10) International Publication Number
WO 2004/038580 A1

(51) International Patent Classification[7]: G06F 3/12

(21) International Application Number:
PCT/SE2003/001663

(22) International Filing Date: 28 October 2003 (28.10.2003)

(25) Filing Language: Swedish

(26) Publication Language: English

(30) Priority Data:
0203165-6 28 October 2002 (28.10.2002) SE

(71) Applicant *(for all designated States except US)*: **DEVLABS AB** [SE/SE]; Kista Science Tower, S-163 80 Kista (SE).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **BERGLIN, Patrik** [SE/SE]; Borgargatan 6, S-117 34 Stockholm (SE).

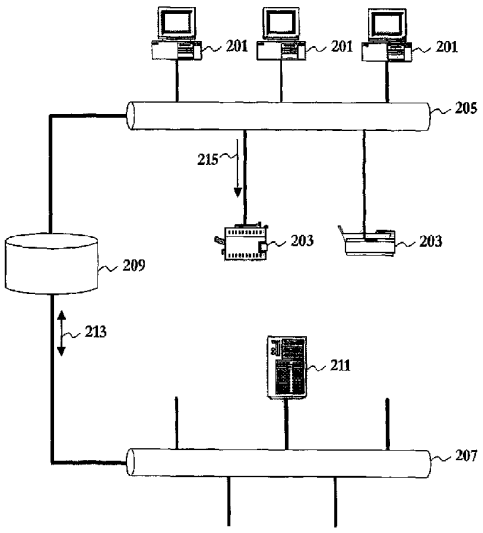
(74) Agents: **FRITZON, Rolf** et al.; Kransell & Wennborg AB, Box 27834, S-115 93 Stockholm (SE).

(81) Designated States *(national)*: AE, AG, AL, AM, AT (utility model), AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (utility model), CZ, DE (utility model), DE, DK (utility model), DK, DM, DZ, EC, EE (utility model), EE, EG, ES, FI (utility model), 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, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK (utility model), SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO utility model (GH), ARIPO patent (GH), ARIPO utility model (GM), ARIPO patent (GM), ARIPO utility model (KE), ARIPO patent (KE), ARIPO utility model (LS), ARIPO patent (LS), ARIPO utility model (MW), ARIPO patent (MW), ARIPO utility model (MZ), ARIPO patent (MZ), ARIPO utility model (SD), ARIPO patent (SD), ARIPO utility model (SL), ARIPO patent (SL), ARIPO utility model (SZ), ARIPO patent (SZ), ARIPO utility model (TZ), ARIPO patent (TZ), ARIPO utility model (UG), ARIPO patent (UG), ARIPO utility model (ZM), ARIPO patent (ZM), ARIPO utility model (ZW), ARIPO patent (ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI,

*[Continued on next page]*

(54) Title: METHOD AND ARRANGEMENT FOR USE OF SHARED RESOURCES IN A NETWORK




(57) Abstract: A method for, from a server, controlling transfers of jobs from clients to shared resources in the network comprises that a request for sending a job to a shared resource selected by a client is received from the client and it is checked continuously whether the resource is available and has capacity by the moment for reception of jobs. A go-ahead is sent immediately to the client that the client can send the job to the resource if the resource is available and has capacity by the moment. The request is placed in a queue for the resource if the resource is available, but for the moment lacks capacity, the queue is updated continuously and a go-ahead is sent to the client that the client can send the job to the resource when the request has advanced to the first position in the queue and the resource has capacity. The step of receiving a request is followed by assigning an identity to the job, which is sent to the client, and the identity is comprised in the go-ahead sent to the client so that the client from said identity can identify the job. A confirmation that the job has been completed successfully by the resource or an indication that the job has not been completed successfully by the resource is received from the client, after which the request is removed from the queue.

WO 2004/038580 A1

SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, 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.*

# METHOD AND ARRANGEMENT FOR USE OF SHARED RESOURCES IN A NETWORK

## TECHNICAL FIELD OF THE INVENTION

The present invention relates to a technique for use of shared resources in a network by a client connected in the network. The technique is particularly, but not exclusively, designed for direct transfer of print jobs from the client to a printer connected in the network.

Specifically, the invention relates to a method for controlling and monitoring from a server use of a shared resource by a client connected in the network, a computer program product for performing said method of controlling and monitoring use, a server having said computer program product for performing said method of controlling and monitoring use of shared resources installed, a method for use of a shared resource in a network by a client, a computer program product for performing said method of use of shared resource, a client terminal having said computer program product for performing said method of using a shared resource installed, and a network comprising a plurality of said client terminals and at least one of said server.

## RELATED ART

A conventional printing system for networks is based on the principle of storage of print jobs on a server, which involves that a client in a network sends each print job to a print server, which receives a print job, processes the print job and sends it further to a printer for printing.

In Fig. 1 an example of such a network is shown. The network comprises a number of clients or user computers 101, which together with a number of printers 103 are connected in a client network 105. The client network 105 is in turn connected to a server network 107, optionally via a router 109. The server network 107 comprises, inter alia, a number of central print servers 111, which monitor and control print jobs from the clients 101, and send them to the printers 103.

2003274867 18 Feb 2009

The client sends a print job to one of the print servers 111, which is schematically indicated by arrow 113. The amount of data, which is sent, is typically three times as big as a corresponding file is on the local hard disk drive of the client. When the print job reaches the print server 111 it is locally spooled onto the hard disk drive of the print server, which requires storage space on the server. When the print server 111 has spooled the print job onto the correct print queue (i.e. in the queue for the selected printer) and when it is time for the current print job to be printed, the print server 111 sends the print job to the selected printer for printing, which is schematically indicated by arrow 115.

As being illustrated in Fig. 1, the print job is not sent the shortest distance to the printer, but makes a detour via a print server 111.

WO 99/38068 describes a job token printer assignment system for print jobs, wherein a client sends a token to a print server, which comprises the profile of the print job, which is to be printed. The print server parses the token, determines an appropriate printer for the print job, and returns a selected printer token to the client, which includes the network address and name of the selected printer. The client then sends the print job, accompanied by the selected printer token, to the appropriate printer. In this way, the print job is assigned to a proper printer for the print job by the print server.

Reference to any prior art in this specification does not constitute an admission that such prior art forms part of the common general knowledge.

**SUMMARY OF THE INVENTION**

The inventors of the present invention have realized that a problem with the above said concept using print servers is that the number of print queues and the number of print jobs, which can be handled by each print server, is limited to the amount of memory, the processing capacity and the storage space of the server. When the print server is heavily loaded the print jobs are handled slower.

COMS ID No: ARCS-224075 Received by IP Australia: Time (H:m) 15:06 Date (Y-M-d) 2009-02-18

Large networks require a large number of print servers, which involves increased costs in the form of hardware, software and operation and maintenance.

Another problem with the known print server concept is that the print job is sent over a network twice: one time from the computer to the print server and then another time from the print server to the printer to be printed. This implies an increased need of bandwidth, which can be demanding and thus costly, particularly in large networks or in networks where the printing frequency is high.

The job token printer assignment system described in WO 99/38068 solves indeed the problem of sending the print job twice over the network. The print server has not to spool a large print job, but needs only to identify the features comprised in the token from the client and assign the print job to an appropriate printer. It seems though like that the system described in WO 99/38068 is related to a system for editorials and the like, where printing of books or other is performed in a larger number and where no flexibility regarding the printings is necessitated. The system described in WO 99/38068 does for instance not allow the clients to select a printer among many and cannot parse the selected printer to see if it has printing capacity by the moment. The system neither manages to handle print queues nor controls the printings in dependence on how the print jobs advance in the different queues. Further, it has no capability to handle several print jobs from the client simultaneously, nor capacity to monitor how the print jobs are actually performed, particularly if/when they are completed.

Furthermore, the inventors of the present invention have realized that similar problems generally occur when shared resources are used on the network. For instance, in the case where a device to store data on a data storage medium in the network, for instance a CD or DVD recorder (writer), where a job shall be sent to the shared resource for recording, a control and supervision may be performed from a server similar

2003274867 18 Feb 2009



to the above described print server, which leads to the same kind of problems. Alternatively, the shared resource is connected in the network in a manner, which only allows a single client at a time to use the resource, and wherein no queue at all exists.

It would be desirable to provide a technique for transfer of jobs from a client connected in a network to a shared resource connected in the network and selected by the client, particular a printer, a telefax apparatus, a projector, or a device for storing data on a storage medium, which lacks the above said problems and limitations.

It would also be desirable to provide such a technique for direct and flexible transfer of jobs from the client connected in the network without there being a risk of losing jobs.

It would also be desirable to provide such a technique, which is simple, robust and reliable.

It would also be desirable to provide such a technique which makes use of a minimum of hardware and software and which simultaneously is capable of handling jobs in large networks and in networks where use of said shared resources is high.

It would also be desirable to provide such a technique by using methods for, from a server, controlling and monitoring transfers of jobs from a client connected in a network to a shared resource connected in the network and selected by the client, computer program products for performing the methods, servers and client terminals having the program products installed, and networks comprising the servers and the client terminals.

COMS ID No: ARCS-224075 Received by IP Australia: Time (H:m) 15:06 Date (Y-M-d) 2009-02-18

2003274867 18 Feb 2009



The present invention, provides methods, computer program products, servers, client terminals, and networks in accordance with the appended patent claims.

A ticket-based centralised server system is used to control the client and his/her transfers of jobs to the shared resource. The technique is unique since the programs, which control the flow of jobs, exist both on the client and on a centralised ticket server. The programs start and stop the use of the shared resource by the client, and the flow of jobs is controlled from the ticket server.

A unique identity is assigned to the jobs by the server, which informs the client of the unique identity. Hereby, the client may initiate several jobs, to be performed by the shared resource, closely after each other, and these jobs are handled individually by the ticket server.

A confirmation that the respective job has been completed successfully by the shared resource, or an indication that the job has not been successfully by the shared resource, is received by the ticket server from the client, after which the corresponding request is removed from the queue.

Preferably, updated status information regarding the completion of the job by the shared resource repetitively during use of the shared resource by the client, where absence of such updated status information at the ticket server indicates an operation error at the client or a communication error in the communication between the client and the ticket server. This will trigger the ticket server to examine the situation and to attend to the problem, or to change status for the job and/or for the shared resource. The job may or may not be removed from the queue. The server may have a timer installed, wherein different actions may be triggered by different delays.

COMS ID No: ARCS-224075 Received by IP Australia: Time (H:m) 15:06 Date (Y-M-d) 2009-02-18

2003274867 18 Feb 2009



By using a ticket-based system the network utilization is limited heavily in contrast to the case where the network is used twice.

The inventive technique is also lessening the load on the centralised server by means of the fact that the client processes the use and sends any data directly to the shared resource instead of sending it to the server, which in turn sends it further to the shared resource.

Further, the ticket-based concept involves an increased capability of changing priorities of jobs. If for instance a job is interrupted before it has been given a green light, the job needs not to be sent on the network at all - in contrary to the known technique where transfers of print jobs are always sent from a client to a print server.

The proposed concept also makes use of the capacity surplus and the parallel processing capacity that exist on the clients of today, which typically are personal computers. In many of the larger office environments powerful personal computers are used for simpler processing such as e-mail handling, word processing and Internet browsing. These computers are very suitable for background communication with a ticket server and at a green light from the ticket server, initiate transfer of the job to the shared resource.

Further advantages of the invention, and characteristics thereof will be evident from the detailed description of preferred embodiments of the present invention given hereinafter and the accompanying Figures, which are given by way of illustration only, and are thus not limitative of the present invention.

In a first aspect the invention provides a method for controlling and monitoring, from a server, transfers of jobs

COMS ID No: ARCS-224075 Received by IP Australia: Time (H:m) 15:06 Date (Y-M-d) 2009-02-18

2003274867 18 Feb 2009



from clients connected in the network to shared resources connected in the network, **characterized by** the steps of:

- receiving from a client connected in the network a request to be allowed to send a job to a selected shared resource connected in the network; and
- checking continuously whether said shared resource connected in the network is accessible and has capacity for receiving a job at present, wherein
- a go-ahead is sent immediately to the client connected in the network that the client can send the job directly to the selected shared resource if the selected shared resource is accessible and has capacity to receive a job at present;
- the request is placed in a queue for the selected shared resource if the shared resource connected in the network is accessible but at present lacks capacity for receiving a job, the queue is updated continuously, and a go-ahead is sent to the client connected in the network that the client can send the job directly to the selected shared resource when the request has advanced to the first position in said queue and the selected shared resource has capacity to receive a job; and
- the client connected in the network is notified if the selected shared resource is not accessible, wherein
- said step of receiving a request from a client connected in the network is followed by assigning an identity to the job, which is notified the client connected in the network, and said identity is comprised in the go-ahead, which is sent to the client connected in the network that the client can send the job directly to the selected shared resource, so that the client connected in the network based on said identity is capable of identifying said job; and
- a confirmation of the fact that the job has been completed successfully by the shared resource or an indication of the fact that the job has not been completed successfully by the shared resource is received from the client, after which said request is removed from said queue.

COMS ID No: ARCS-224075 Received by IP Australia: Time (H:m) 15:06 Date (Y-M-d) 2009-02-18

2003274867 18 Feb 2009



In a second aspect the invention provides a method for transfer a job from a client connected in the network to a shared resource selected by the client among several shared resources connected in the network, **characterized by** the steps of:

- sending to a server for controlling and monitoring transfers of jobs to shared resources connected in the network, a request to be allowed to send a job directly to the selected shared resource connected in the network, where said server is adapted to assign an identity to the job, which is communicated to the client connected in the network, to place the request in a queue for the selected shared resource, to update the queue continuously, and to give go-ahead to the client connected in the network when he/she is next to send the job to the selected shared resource; and
- receiving a message of said identity from said server;
- preparing and storing said job;
- receiving a go-ahead from said server that the job may be sent to the selected shared resource, where the go-ahead comprises said identity so that said job can be identified based on said identity;
- sending the job directly to the selected shared resource; and
- receiving from the selected shared resource a confirmation that the job has been completed successfully by the shared resource or an indication of the fact that the job has not been completed successfully by the shared resource, and forwarding this confirmation or indication to the server so that removal of the request from said queue is made possible.

**BRIEF DESCRIPTION OF THE DRAWINGS**

Fig. 1 illustrates schematically a network, wherein a printing technique according to the prior art is implemented.

COMS ID No: ARCS-224075 Received by IP Australia: Time (H:m) 15:06 Date (Y-M-d) 2009-02-18

Fig. 2 illustrates schematically a network, wherein an embodiment of the present invention is implemented.

Short description of fig. 2:

2-1. The client sends a request to the print server for a job ID in order to print on a selected printer.

2-2. The ticket server has all information regarding the status of the printers in a real time database and if the selected printer is active on the network the ticket server sends a job ID to the client.

2-3. The client receives its job ID and performs the printing on the printer.

2-4. At most 20 kb data is sent between the client and the ticket server.

Figs 3-7 are flow diagrams showing different aspects of a preferred embodiment of the present invention.

Short description of Fig. 3:

3-1. Send sequence, DLL version, print driver version, printer name, user name and user domain.

3-2. Check the information received in step 1 to see if something is too old or if the user has not access to print, go to step 4.

3-3. Put the data received in step 1 in the database, fetch the unique value generated by the SQL server and send it back to the client.

3-4. Read the error code and check what has to be attended to, if a version of something is old, fetch a newer version. After having performed the updating, request the user to print again.

If the user has not access to print on a selected printer the user may have to perform an interactive selection, where the user can select what has to be done with the printing: cancel or move to another printer.

3-5. Receive job ID.

3-6. Send sequence, IP address, number of pages, size of print job, job identification, front end paper version and configuration file version.
3-7. Process data.
3-8. Update information for the print job in the database, send confirmation to client.
3-9. Does not the user have access?
3-10. Give the user possibility to move the print job to another printer or to cancel the job completely. If the user selects to move the print job to another printer, the client has to start over again from step 1.
3-11. Was client too old?
3-12. Fetch new version of the client, and then the client has to be restarted if the client is run as a service.
3-13. Print driver version was too old, update print driver and start the client.
3-14. Shut down TCP/IP connection if not NAT.

Short description of Fig. 4:

4-1. Send sequence, job ID, error code for request to cancel print job.
4-2. Process data.
4-3. Cancel print job from database and tell client to cancel job.
4-4. Cancel job from disk/memory.

Short description of Fig. 5:

5-1. The server requests to interrupt print job.
5-2. Change status of print job in database to "interrupt".
5-3. Send sequence to client to request cancellation of the print job.
5-4. Cancel print job and tell printer that the job is cancelled from the client.
5-5. Process data.
5-6. Cancel print job from database.

Short description of Fig. 6:

6-1. Fetch first job from print queue (sorted after the oldest job and then priorities).

6-2. Highlight job for printing.

6-3. Tell client to start printing, information which is to be sent is sequence number, job ID, printer IP address and which function shall be used for printing.

6-4. Could the printing be started?

6-5. Process data.

6-6. Update status of job to "printing".

6-7. Is the number of attempts larger than the maximum number of allowed attempts?

6-8. Wait a predetermined number of seconds, go then to step 4.

6-9. Error code is sent to the server. For instance the client could not connect to the printer.

6-10. Process data.

6-11. Update information in database according to information received from client, check printer by daemon controlling printers.

6-12. Could the printing be performed without any errors from the printer?

6-13. Process data.

6-14. Move information of print job to separate table in the database so that it later can be used to generate statistics and possibly invoicing information.

6-15. Send error code to server.

6-16. Process data.

6-17. Update information in database and highlight job as failed, check printer by daemon controlling printers.

Short description of Fig. 7:

7-1. Has the queue stopped to operate?

7-2. Inform support organization.

7-3. Inform all clients having print jobs awaiting on the printer that the printer is not operating.

7-4. Show a dialogue window for the client, where the client receives a possibility to cancel the job, to move the job to another printer, or to wait until the printer is operating again.
7-5. The client wishes to remove the job?
7-6. Process data.
7-7. Cancel job from database.
7-8. The client wants to move the job to another printer?
7-9. Has the client access to print on the selected printer?
7-10. Is the selected printer operating?
7-11. The client is waiting for the printer to start operate again.
7-12. Move the print job to selected queue.
7-13. Tell user that the user has not access to print on the selected printer, start over again from step 4.

**PREFERRED EMBODIMENTS**

In the following description, for depicting and not limiting purposes, specific details are set forth, such as particular applications, techniques, methods etcetera in order to provide a thorough understanding of the present invention. It shall, however, be readily apparent to the man skilled in the art that the invention can be practised in other embodiments deviating from these specific details. In other instances detailed descriptions of well-known methods, protocols, devices or circuits are omitted in order not to obscure the present description with unnecessary details.

The description is divided into five different sections, the first of which generally describes the main properties of a technique for direct printing on networks according to a preferred embodiment of the present invention. Then, detailed descriptions of preferred embodiments of the ticket server and the client, respectively, follow. Thereafter, different alternatives to a printer for use as a shared resource accord-

ing to the present invention are described. Finally, a number of advantages of the technique is listed.

1. General description of a technique for direct printing in networks according to a preferred embodiment of the present invention

In fig. 2, an example of a network is shown, wherein the printing technique according to the present invention is implemented. The network comprises a number of clients or user computers 201, which together with a number of printers 203 are connected in a client network 205. The client network 205 is in turn connected to a server network 207, optionally via a router 209. The server network 207 comprises i.a. a ticket server 211, which monitors and controls the printings from the clients 201 on the printers 203.

The printings are performed according to the following. The client, who desires to print on a selected printer, sends a request to the ticket server 211 to obtain a green light in order to print a print job on the printer. If the printer is available and active, i.e. it can receive a print job, the ticket server 211 sends a go-ahead to the client and the client, thus receiving the go-ahead, sends the print job directly to the selected printer for printing. For this purpose the ticket server 211 monitors the printer and has thus information whether the printer is active or inactive, and if it is occupied with print jobs from the same or another client on the network.

If the printer is occupied, the request is placed in a queue by the ticket server 211. This queue is updated continuously, and when the above print job is next to be printed the ticket server 211 sends a go-ahead to the client, which may send the print job directly to the printer for printing.

In fig. 2 the bidirectional arrow 213 indicates the signalling between the ticket server 211 and one of the clients 201, while

arrow 215 indicates the transferring of the print job from the client to the selected printer 201.

The print job is assigned an identity, which is sent to the client, so that the client can associate an identity with the print job. Then, when the ticket server 211 sends the go-ahead to the client, the identity is comprised therein so that the client can send the correct print job to the printer (if the client has several print jobs active). A confirmation that the assigned identity has been received by the client can be sent back to the ticket server 211.

Furthermore, the client may send a confirmation to the ticket server 211, when it has received the go-ahead to print. This confirmation can also comprise an indication that the print job has been sent or will be sent by the client. A further confirmation when the print job has been completed may be sent from the client to the ticket server 211, or in the case the print job has not been completed successfully, an indication of this may be sent from the client to the server, after which the server can remove the request from the queue.

Preferably, updated status information regarding the completion of the print job is sent from the client to the ticket server 211 repetitively, particularly on a regular basis, while the printing is active, wherein absence of such updated status information at the ticket server 211 (i.e. if the time between two status updates has been exceeded) indicates that an operation error of the client, or that a communication error in the combination between the client and the ticket server 211, has occurred (program error or network error). This triggers the ticket server to examine the situation (e.g. examine the printer), and attend to the problem or change status of the job and/or the printer. The job may or may not be moved from the queue. The server may have a timer installed, wherein different actions can be trigged by the different delays. If errors are indicated the administrator of the system is informed.

Further, information regarding status of the printer may be sent from the ticket server 211 to the client to keep the client updated. More about this follows below.

The amount of data, which is sent between the client and the ticket server 211, is typically at most 20 kb (in the version developed by applicant and which is described in detail below) and this creates possibilities to save much bandwidth and simultaneously centralize the ticket server 211 to one site in the network only, even if the network is large.

2. Detailed description of a preferred embodiment of a ticket server

A detailed description of the ticket server for controlling and monitoring transfers of print jobs from a client to a printer according to the present invention follows below. Note that in both this and the following section, the function of the invention and the communication between the server and the client is described. References to the flow diagrams in figs 3-7 are indicated in parenthesis where appropriate (the steps in the flow diagrams are given by a figure corresponding to the number of the figure followed by a dash and a serial number).

The ticket server comprises the following parts:

- Computer
- Database
- Program receiving requests of print jobs from client
- Program handling print queues
- Program or server daemon for checking printer status
- Software making file and printer sharing possible for Windows, OS/2 and DOS clients
- Program for replication of data between a number of ticket servers.

In the description below, it is assumed that the network is a TCP/IP-based network. After modifications, the invention is obviously applicable for any kind of network.

Printing: first stage (fig. 3)

Initialization of the printing is performed by the client. When the ticket server receives a message from a client with a request for a print job, the ticket server receives the following data from the client (step 3-1):

- ☐ Sequence number
- ☐ Client version number
- ☐ Operative system
- ☐ Printer driver version
- ☐ Printer name
- ☐ User domain
- ☐ User name

When the server has read the information that the client has sent, a sanitary check of the client data is performed, i.e. the information is checked according to the following rules (step 3-2), and if something is erroneous, an error code is generated (step 3-4, and step 3-9 - step 3-13).

It is checked whether the data received is complete. If this is not the case an error code is sent to the client requesting the information. If the client software is old, the client can be forced to upgrade its client software. In such a case an error code for this is sent. Further, it is checked whether the printer driver, which the client is using, is unsuitable in some manner, e.g. if a bug or similar has been found in the driver so that it should be changed immediately. If so, an error code for this is sent. Finally, it is checked whether the user has access to print on the printer (step 3-9). If this is not the case an error code is sent to the client, which is invited to select another printer.

If an error code is to be sent, it is sent according to the following:

- ☐ Sequence number (corresponding to an error code being sent)

- Error code

The error is logged and the print attempt results normally in no further actions from the part of the server, but a new print attempt has to be generated by the client.

If all information was correct, the server is entering information in the database and receives a unique identifier back from the database (step 3-3). This is used in later communications between the client and the server. Also the print job is "marked" with this identifier in order for the client to know which print job is regarded (since the client has to be capable of handling several print jobs simultaneously, the client has to have an identifier for each print job to prevent sending an incorrect print job to an incorrect printer).

The identifier is sent back to the client in a sequence according to the following:

- Sequence number
- Job ID (the above said identifier).

When the client has received the above sequence (step 3-5), it sends back data describing the print job more in detail (step 3-6). The information, which is received from the client, is:

- Sequence number
- The IP-address of the client
- Number of pages
- Number of bytes (size of the print job)

The received data is processed by the server (step 3-7), and the information of the print job is updated in the database (step 3-8).

The IP-address of the client is sent in order for the server to decide whether the client is on a so-called NAT (Network Address Translation) network. This is done since it is often

very difficult to re-establish a connection in a different direction, i.e. from the server to the client if the client is on a NAT network. If so is the case, the connection will be hold open until all communication regarding the print job is completed (i.e. until all steps in this description are performed). In normal instances, i.e. if the client is not on a NAT network, the connection is closed between the different sessions in order to make an optimum printing technique without any unnecessary signalling overhead possible.

The number of pages and the number of bytes are collected in order partly to see how large print jobs are being sent, partly for registration for statistical reasons.

The server responds to the client with a sequence, wherein it tells the client that the server is ready and that the client should wait for further information from the server (step 3-8).

The sequence is:

- Sequence number

The connection between client and server is then closed, unless the client is on a NAT network, see above.

Printing: second stage (fig. 6)

The server interrogates continuously the database of new print jobs. This is performed by first performing a search to find the printers on the network, which have print jobs in queue, and thereafter each queue is parsed to find all print jobs in the queue for the respective printer. This information is thereafter sorted by time stamp, i.e. when the print job came to the server. The FIFO-principle (FIFO, first in first out) is preferably employed, i.e. the oldest print job is processed first. The print jobs can also be sorted using other kinds of priority, i.e. if a print job has a higher priority than others, which are before in the queue, the print job with

higher priority is moved before the print jobs with lower priority in the queue.

The server daemon, which monitors status of the printers in the network, is described in the end of this section.

When the server has found a print job, which is to be printed, the print job is fetched from the print queue (step 6-1) and marked for printing (step 6-2). The server then fetches information of the printer, on which the print job is to be printed. The server fetches information from different parts of the database. The IP-address for the printer and the kind of mechanism that shall be used for sending the print job from the client to the printer is collected. It also fetches information from another part of the database where status of the printers is stored. Status is collected by the daemon that supervises printers.

If status of the printer is something else than OK, a sequence is sent to the client telling that the printer is not operating and makes it possible for the client to either cancel the print job, select another printer, or wait for the printer to start operate again.

The sequence, which then is sent, is accordingly:

- Sequence number
- Error code.

If the printer is active, i.e. in operation, and also accessible, the server sends information to the client according to the following (step 6-3):

- Sequence number
- Job ID (the unique identifier)
- The IP-address of the printer
- Mechanism of printing.

The client then confirms start of printing by responding with a sequence, wherein it informs that the printing has been started (step 6-4).

If it was not possible to connect to the printer, another sequence (step 6-9) is sent, possibly after a number of failed attempts (step 6-7 - step 6-8).

When the printing has successfully been started a sequence is sent according to:

- Sequence number.

If on the other hand it was not possible to start the printing, the following sequence is sent:

- Sequence number
- Error code (cause to why).

When the server receives the response from the client, it checks which sequence it is (step 6-5 and step 6-10). When the printing has been started successfully, the server updates the status of the print job in the database to "being printed" (step 6-6). If, on the other hand, it was not possible to start the print job, the information for the printer is updated so that the daemon monitoring printers knows that it shall check the printer immediately (step 6-11). The connection between the server and the client is closed.

During the printing, the method of receiving, from the client, updated status information, which has been described above with reference to fig. 2, is used (not illustrated here).

After a successful printing the following sequence is received from the client (step 6-12):

- Sequence number
- Job ID.

The received data is processed in the server (step 6-13), and the information of the successfully completed print job is moved to a separate table in the database, so that it can later be used for generating statistics and possibly invoicing information (step 6-14).

If the printing has not been completed, a sequence is received from the client according to (step 6-15):

- Sequence number
- Job ID
- Error code.

The server processes received data (step 6-16), updates the information in the database, and marks the print jobs as failed. Further, information for the printer is updated so that the daemon monitoring printers knows that it should check the printer immediately (step 6-17).

The server may allow a plurality of print jobs to be printed simultaneously on a printer depending on the sizes of the print jobs and the memory or storage capacity of the printer in question. The server may monitor the number of print jobs and/or the amount of byte print jobs, which are active on a particular printer.

There is a possibility for the server to interrupt the print job (step 5-1 in fig. 5) at whatever time. Then, the status of the print job is altered in the database to "being removed" (step 5-2), and the server contacts the client and tells it to cancel the print job, and thereafter to confirm it. The following sequence is sent from the server to the client (step 5-3):

- Sequence number
- Job ID.

When the client has cancelled the print job, it responds by sending back a sequence according to (step 5-4):

- Sequence number.

The server processes the data (step 5-5), and cancels the print job from the database (step 5-6).

Note that the information of the clients and their print jobs, information of handling and updating of print queues, and information of status of printers normally are received from the clients according to what has been described above. Nevertheless, the server communicates directly with the printers to obtain further information of these, and possibly for confirming information received from the clients. More about this is described below.

Status checking of printers

The server daemon checks continuously status of printers. This is done by forwarding an SNMP-question (SNMP, Simple Network Management Protocol) to the respective printers. The information, which is received, is:

- Status of the printer
- Toner information (laser printers)
- Paper information.

The information, which is checked, depends on what the printer model supports. Often, more information may be fetched, e.g. if the printer drum of a laser printer has to be changed, paper is stuck, etcetera.

If a printer status is inactive, i.e. "off-line", the server daemon intends to check the cause for this by checking whether paper is stuck, if toner is empty, if a drum has to be changed, etcetera.

The status of the printer in the database is updated to be "off-line", and the clients, which have print jobs, are informed by the server daemon that the printer is not operating and the client has the possibility to move the print job, to keep it in the queue, or to cancel it.

The information may also be sent to a helpdesk or similar department, which can attend to the problem of the printer. If the server daemon can confirm that the paper is empty the information can e.g. be sent to a specific department/person, whose assignments comprise to put some more paper into the printer etcetera.

The system is also continuously checking the database if a direct check of printer is needed. If it finds a printer, which has to be checked immediately, it does not wait until next time the printer normally shall be checked, but performs it immediately instead. This is done e.g. if a client has reported that it cannot connect to the printer.

In fig. 7 is shown a flow diagram of the communication between server and client in case the server daemon detects that a queue is not operating as it should.

Redundancy

In order to handle a shut down of the ticket server the network is preferably equipped with at least two ticket servers. In order to obtain a transparent transfer from one server to another, when the former is shut down by some reason, it is required that all data is replicated between the ticket servers. This can be performed by means of redundancy technique.

The clients may possibly be given possibility to choose either one of the ticket servers to cooperate with.

3. Detailed description of a client

In the system a main program and a number of subprograms are comprised, which together form the client depending on the operative system of the client. The main program is adapted to handle several print jobs simultaneously, and to process correct information for correct printing. Further, the programs have functionality in order to show dialogue windows with important information if something has happened with one or several print jobs, and information of the status of the respective print job.

When a client initiates a print job, i.e. requests a program to send a print job to a preinstalled printer for printing the main program takes over. Hereafter, a dialogue with the ticket server follows.

The client program starts with a request to obtain permission from the ticket server. This is done via RPC, blocking sockets or non-blocking sockets. The data is sent by hexadecimal notation to minimize the size of message, and is distorted by an XOR-algoritm.

In the first dialogue with the server the following information is sent (step 3-1 i fig. 3)

- ☐ Sequence number
- ☐ Client version number
- ☐ Operative system
- ☐ Printer driver version
- ☐ Printer name
- ☐ User domain
- ☐ User name.

A response from the server is received (step 3-5), which comprises

- ☐ Sequence number
- ☐ Job ID.

If the printer driver and/or the client version of the client program software is too old, or is not certified, an error code sequence will exchange the above said sequence (step 3-4). The sequence has the following information:

- Sequence number
- Error code.

If this occurs the client software will download correct version of either the driver or the client software from the server.

The client spools the print job, depending on size, either directly to the memory or to a catalogue on the computer's local hard disk drive. When the print job has been spooled to the memory or the hard disk drive further information of the print job is sent to the server. The print job is associated with the job ID, which has been received by the server, so that the print job can be identified when it is time for printing.

The following information is sent from the client to the server (step 3-6):

- Sequence number
- IP-address
- Number of pages
- Number of bytes.

Thereafter, a confirmation of the reception of information from the server is received, and a first dialogue part is completed (step 3-8).

Then, no further data is sent between client and server until the print job of the client is ready to be printed or that the printer no longer is accessible.

If the client is not on a NAT network it closes the TCP/IP-connection to the server (step 3-14).

When the printing is to be started, the server will start a new dialogue with the client. This dialogue is started with receiving the following information from the server (step 6-3 in fig. 6):

- Sequence number
- Job ID
- IP-address of printer
- Printer connection - the kind of connection, over which the print job is going to be sent.

The client sends back a confirmation to the server that it has received the information and starts a direct printing towards the printer (step 6-4). Only a sequence number is sent:

- Sequence number.

If the client cannot send the print job, or if something has occurred, so that the print job cannot be completed, a new sequence is sent with correct error code (step 6-9). This information is as follows:

- Sequence number
- Error code.

During the printing, the method of sending, from the client, updated status information as has been described above with reference to fig. 2 may be employed (not illustrated here).

When the printing is ready (if it will be ready) the client sends a new sequence to the ticket server indicating that the printing is completed (step 6-12). This sequence is as follows:

- Sequence number
- Job ID.

If a client interrupts the print job, the following sequence is sent from the client to the server (step 4-1 in fig. 4):

- Sequence number
- Job ID
- Error code.

The data is processed by the server (step 4-2), and the print job is cancelled from the database. The client receives from the server instructions to cancel the print job (step 4-3), after which the client cancels the print job from the memory or hard disk drive (step 4-4).

4. Alternative to a printer for use as a shared resource in accordance with the present invention

It shall be understood that while the preferred embodiments above have been described in connection with use of printers connected in the network, the present invention is not limited in this respect. More generally, *printer* can be exchanged for *shared resource* connected in the network. In this case the *print queue* is exchanged for *queue for use of shared resource*, *print job* is exchanged for *job to be sent to shared resource*, and *printing* is exchanged for *sending the job*. The shared resource can for instance be a transmitter device shared on the network, such as a telefax apparatus, a display unit shared on the network such as a digital projection apparatus, or a shared resource for storing data on a data storage medium on a network such as a CD- or a DVD-recorder. After suitable amendments the description above is applicable for such shared resource devices.

5. Advantages of the technique

The advantages of the present invention comprises inter alia the following:

- Reduced network load. The invention reduces the need of network capacity since the processed job is always stored on the client before it is sent to the shared resource. The job

is always sent directly to the shared resource - no intermediate storage on a server is needed.

- Scalability. The invention is scalable and is adapted for medium size to large enterprises. Since the product has low demands on internal bandwidth, the product can advantageously be used in multinational organizations. For instance, each server has capacity to handle 20000 queues and 100000 jobs simultaneously.

- Low load on the server gives faster use of the shared resource.

- Small demands of hard disk drive space on the server.

- Centralised administration and organization. A centralised solution gives large possibilities to achieve cost efficiency and active monitoring, which gives a maximum use of shared resources.

- Flexibility. The invention is flexible and can be used by most operative systems existing on the market today, which is supported by the manufacturers of shared resources. The solution can be adapted to new situations without any disturbances. Additional modules exist and can be implemented.

- Automated distribution and updating of inter alia drivers for the shared resources. When a new kind of shared resource becomes available on the market, or new drivers are released, these will be tested to then be distributed over the complete network. Patches and updates of components in the system follow the same procedure.

- Certification and testing. All components and drivers are certified and tested before being implemented in the environment of the client.

- Accessibility. The use of redundancy gives high accessibi-

lity.

- Remote monitoring and support. Monitoring of the shared resource, particularly paper and toner supply in case the shared resource is a printer, is available in the administration interface. This facilitates planning of maintenance of the shared resource, particularly replenishing toner and paper in case the shared resource is a printer, in order to minimize the shut down time on the shared resources of the enterprise.

- Economy. The invention is simple to implement and has low demands in respect of hardware, software and network capacity.

- Costs and use of the shared resources can be monitored on all levels, for instance on clients, departments, groups or shared resource.

The invention is obviously not limited to the above-described preferred embodiments, which are also shown on the drawings, but can be modified within the scope of the appended patent claims.

## PATENT CLAIMS

1. A method for controlling and monitoring, from a server, transfers of jobs from clients connected in the network to shared resources connected in the network, **c h a r a c t e - r i z e d b y** the steps of:

- receiving from a client connected in the network a request to be allowed to send a job to a selected shared resource connected in the network; and
- checking continuously whether said shared resource connected in the network is accessible and has capacity for receiving a job at present, wherein
- a go-ahead is sent immediately to the client connected in the network that the client can send the job directly to the selected shared resource if the selected shared resource is accessible and has capacity to receive a job at present;
- the request is placed in a queue for the selected shared resource if the shared resource connected in the network is accessible but at present lacks capacity for receiving a job, the queue is updated continuously, and a go-ahead is sent to the client connected in the network that the client can send the job directly to the selected shared resource when the request has advanced to the first position in said queue and the selected shared resource has capacity to receive a job; and
- the client connected in the network is notified if the selected shared resource is not accessible, wherein
- said step of receiving a request from a client connected in the network is followed by assigning an identity to the job, which is notified the client connected in the network, and said identity is comprised in the go-ahead, which is sent to the client connected in the network that the client can send the job directly to the selected shared resource, so that the client connected in the network based on said identity is capable of identifying said job; and

- a confirmation of the fact that the job has been completed successfully by the shared resource or an indication of the fact that the job has not been completed successfully by the shared resource is received from the client, after which said request is removed from said queue.

2. A method of claim 1, wherein updated status information regarding the completion of the job by the shared resource is received from the client connected in the network repetitively between said go-ahead is sent to the client connected in the network and said confirmation that the job has been completed successfully by the shared resource or said indication that the job has not been completed successfully by the shared resource is received, wherein absence of such updated status information indicates an operation error of the client connected in the network, or a communication error in the communication between the client and the server.

3. A method of claim 1 or 2, wherein a confirmation that said assigned identity has been received by the client connected in the network, is received.

4. A method of any of claims 1-3, wherein a confirmation that the go-ahead, which is sent to the client connected in the network that he/she may send the job to the selected shared resource directly has been received by the client connected in the network, is received.

5. A method of claim 4, wherein said confirmation that the go-ahead, which is sent to the client connected in the network that he/she may send the job to the shared resource directly has been received, is also confirmation that the job has been or will be sent to the shared resource directly.

6. A method of any of claims 1-5, wherein said shared resource is a printer and said job is a print job.

7. A method of claim 6, wherein said shared resource is a transmitter device, particularly a telefax apparatus, a displaying unit, particularly a projector, or a device for storing data, particularly a CD recorder, a DVD recorder or similar, and said job is a job to send, display or store data.

8. A method of any of claims 1-7, wherein information regarding status of the shared resource is sent to the client connected in the network.

9. A method of any of claims 1-8, wherein

- a version of client software for the client connected in the network, which relates to communication with the selected shared resource, is stored,
- information is received from the client connected in the network concerning the version of the client software the client connected in the network is using for communication with the selected shared resource,
- the versions of the client softwares are compared, and
- a copy of said stored client software is transferred to or installed on the client connected in the network if said comparison shows that the stored client software is younger than the client software used by the client connected in the network.

10. A method of claim 9, wherein said information regarding the version of the client software the client connected in the network is using, is received together with said request to be allowed to send a job to the selected shared resource.

11. A method of any of claims 1-10, wherein

- said request to be allowed to send a job to the selected shared resource comprises a user domain and a user identity for the client connected in the network,
- it is checked that the client connected in the network has authorization to send a job to the selected shared resource, and

- if the client connected in the network does not have authorization to send a job to the selected shared resource an error code is sent to the client connected in the network.

12. A method of claim 11, wherein

- if the client connected in the network has authorization to be allowed to send a job to the selected shared resource it is checked which user priority the client connected in the network has, and

- said request is put in said queue depending on the user priority of the client connected in the network.

13. A method of any of claims 1-12, wherein information of the size of said job is received from the client connected in the network.

14. A method of claim 13 when depending on claim 3, wherein said information of the size of the said job is received together with said confirmation that said assigned identity has been received.

15. A method of any of claims 1-14, wherein status of the shared resources in the network is checked continuously.

16. A method of any of claims 1-15, wherein information regarding status of shared resources in the network, information regarding queues and information regarding clients and jobs are continuously copied to a second server for controlling and monitoring of transfers of jobs.

17. A computer program product downloadable into the internal memory of a server and comprising software code portions for performing the method of any of claims 1-16 when said computer program product is executed on said server.

18. A server having the computer program product of claim 17 downloaded.

19. A method for transfer a job from a client connected in the network to a shared resource selected by the client among several shared resources connected in the network, **c h a r a - c t e r i z e d b y** the steps of:

- sending to a server for controlling and monitoring transfers of jobs to shared resources connected in the network, a request to be allowed to send a job directly to the selected shared resource connected in the network, where said server is adapted to assign an identity to the job, which is communicated to the client connected in the network, to place the request in a queue for the selected shared resource, to update the queue continuously, and to give go-ahead to the client connected in the network when he/she is next to send the job to the selected shared resource; and
- receiving a message of said identity from said server;
- preparing and storing said job;
- receiving a go-ahead from said server that the job may be sent to the selected shared resource, where the go-ahead comprises said identity so that said job can be identified based on said identity;
- sending the job directly to the selected shared resource; and
- receiving from the selected shared resource a confirmation that the job has been completed successfully by the shared resource or an indication of the fact that the job has not been completed successfully by the shared resource, and forwarding this confirmation or indication to the server so that removal of the request from said queue is made possible.

20. A method of claim 19, wherein updated status information regarding the completion of the job by the shared resource is sent to said server repetitively between said go-ahead is received from said server and said confirmation that the job has been completed successfully by the shared resource or said indication that the job has not been completed successfully by the shared resource is received from the shared resource and is forwarded to said server, wherein absence of

such updated status information on said server indicates an operation error of the client connected in the network, or a communication error in the communication between the client and the server.

21. A method of claim 19 or 20, wherein a confirmation that said message of said assigned identity has been received, is sent to said server.

22. A method of any of claims 19-21, wherein a confirmation that said go-ahead that the job may be sent to the selected shared resource has been received, is sent to said server.

23. A method of claim 22, wherein said confirmation that said go-ahead has been received is also confirmation that the job has been or will be sent to the shared resource.

24. A method of any of claims 19-23, wherein said shared resource is a printer and said job is a print job.

25. A method of claim 24, wherein said shared resource is a transmitter device, particularly a telefax apparatus, a displaying device, particularly a projector, or a device for storing data, particularly a CD recorder, a DVD recorder or similar, and said job is a job to send, display or store data.

26. A method of any of claims 19-25, wherein

- information is sent to said server concerning the version of the client software the client connected in the network is using for communication with the selected shared resource,

- a copy of a client software stored by the server is receive from said server if said copy has a later version number than the version number of the client software the client connected in the network is using.

27. A method of claim 26, wherein said information regarding the version of the client software the client

connected in the network is using, is sent to said server together with said request to be allowed to send a job to the selected shared resource.

28. A method of any of claims 19-27, wherein

- said request to be allowed to send a job to the selected shared resource comprises a user domain and a user identity for the client connected in the network; and
- an error code is received from said server if the client connected in the network does not have authorization to send a job to the selected shared resource.

29. A method of any of claims 19-28, wherein information of the size of said job is sent to said server.

30. A method of claim 29 when depending on claim 21, wherein said information of the size of the said job is sent to said server together with said confirmation that said assigned identity has been received.

31. A computer program product downloadable into the internal memory of a server and comprising software code portions for performing the method of any of claims 19-30 when said computer program product is executed on said server.

32. A client terminal having the computer program product of claim 31 downloaded.

33. A network comprising at least one server according to claim 18 and a plurality of the client terminal of claim 32.

1/7

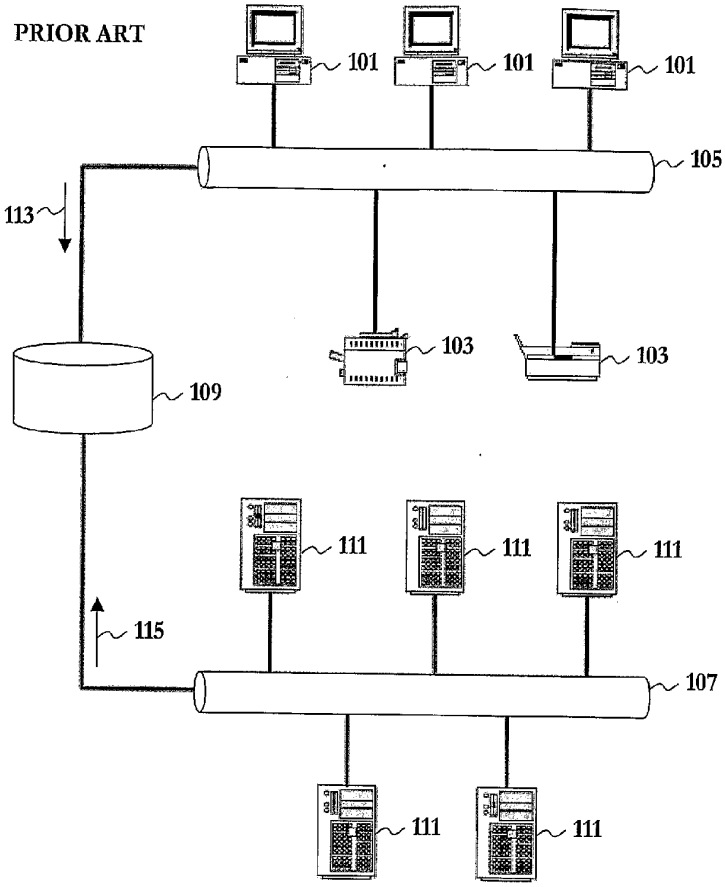

PRIOR ART
101
101
101
105
113
103
103
109
111
111
111
115
107
111
111

Fig. 1

2/7

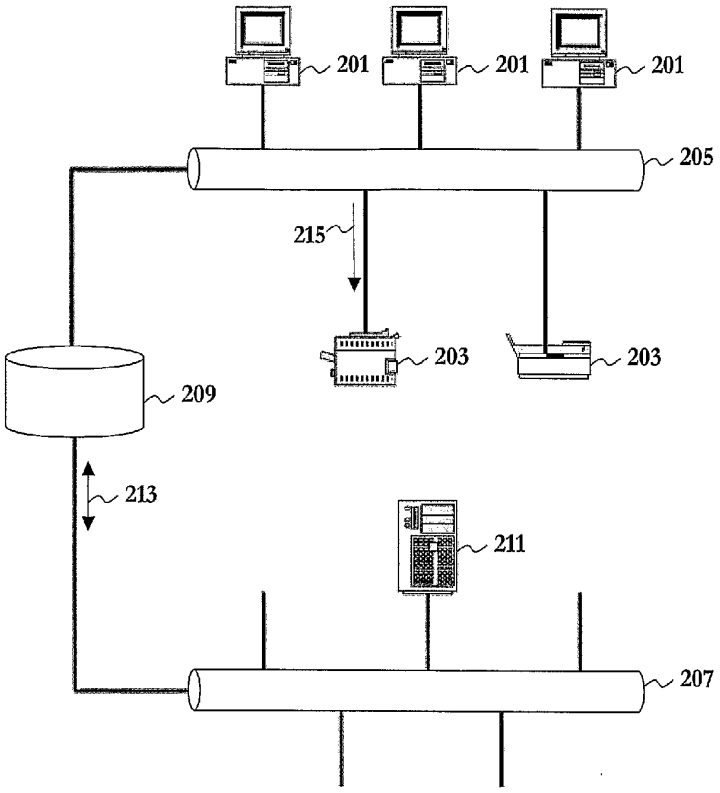



Fig. 2

3/7

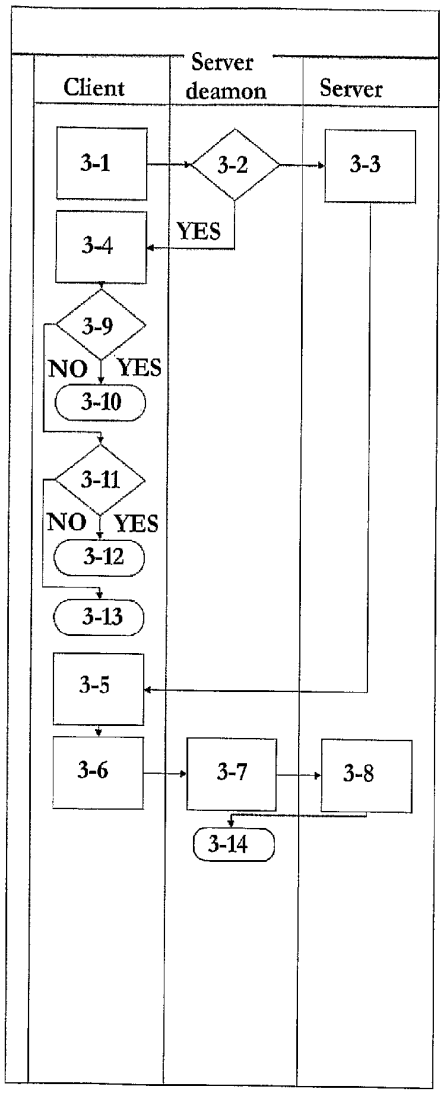



Fig. 3

SUBSTITUTE SHEET (RULE 26)

4/7

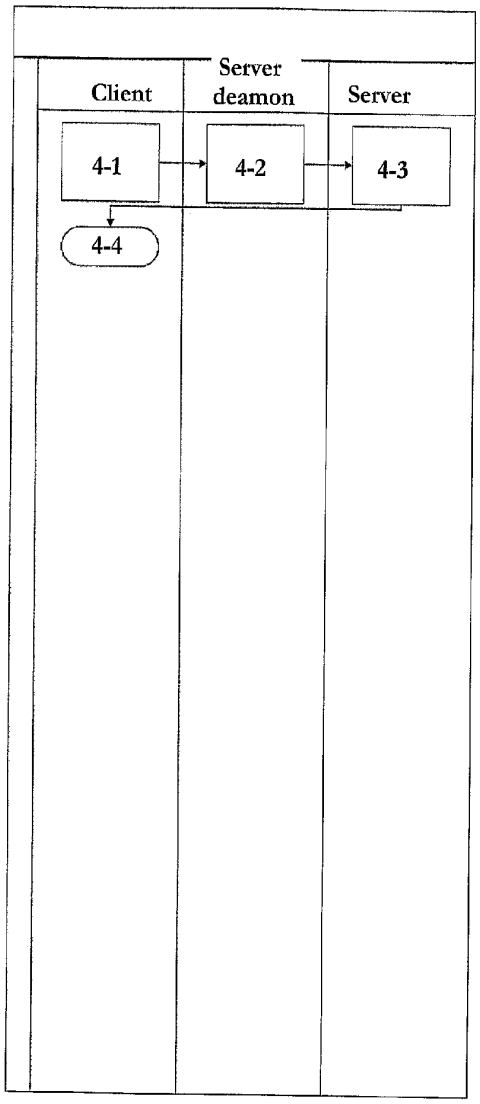



Fig. 4

5/7

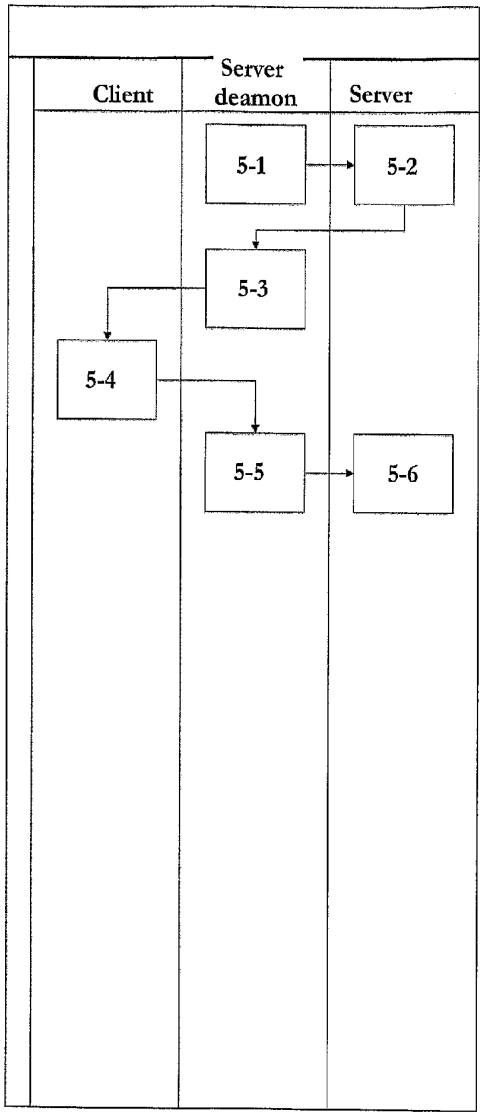



Fig. 5

SUBSTITUTE SHEET (RULE 26)

6/7

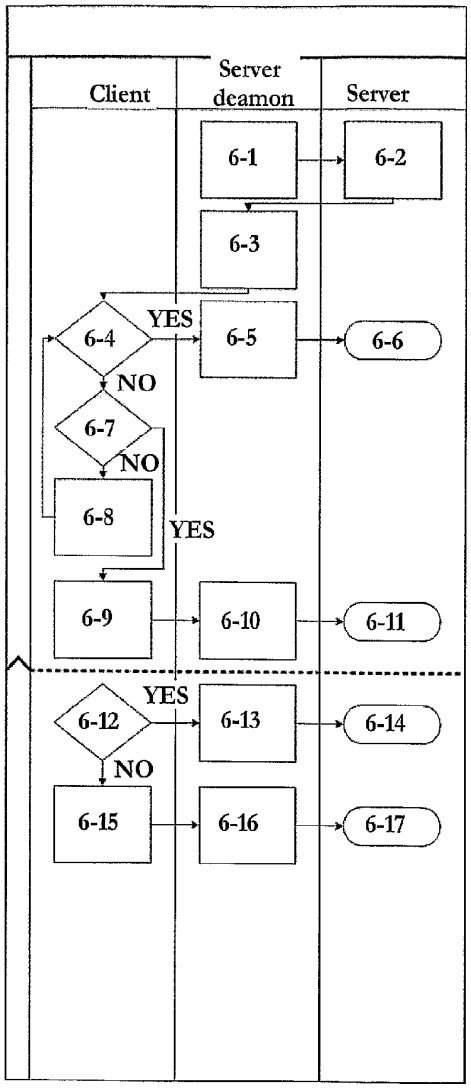



Fig. 6

SUBSTITUTE SHEET (RULE 26)

7/7

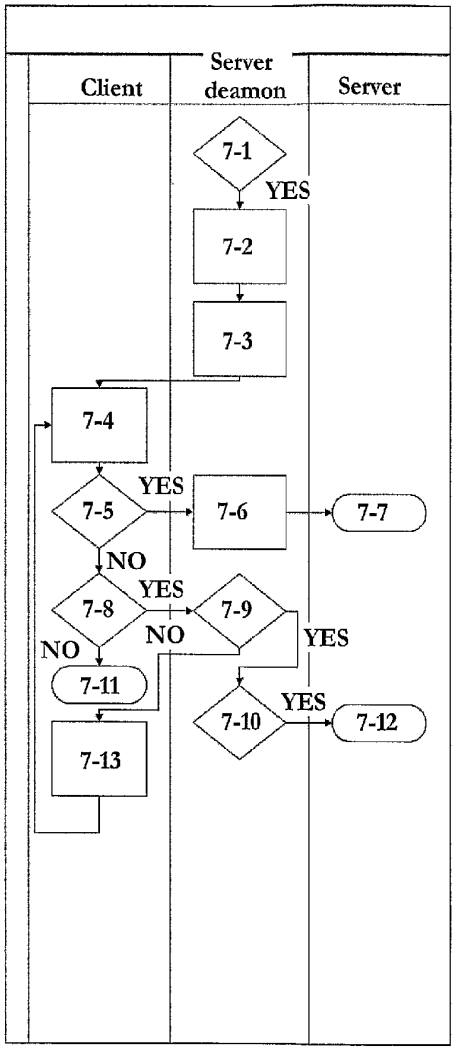



Fig. 7

SUBSTITUTE SHEET (RULE 26)