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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 April 2001 (19.04.2001)

PCT

(10) International Publication Number
WO 01/28211 A1

(51) International Patent Classification[7]: H04M 3/50, 3/00, 11/00, G06F 9/46, 15/68, 15/00

(21) International Application Number: PCT/US00/28071

(22) International Filing Date: 11 October 2000 (11.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/415,617 11 October 1999 (11.10.1999) US

(71) Applicant: **DICTAPHONE CORPORATION** [US/US]; 3191 Broadbridge Avenue, Stratford, CT 065614 (US).

(72) Inventors: **CYR, James**; 8 Christopher Lane, Clinton, CT 06413 (US). **GREENE, Channell**; 69 Plankton Street, Bridgeport, CT 06606 (US). **KUHNEN, Regina, J.**; 66 Woodhaven Drive, Trumbull, CT 06611 (US).

(74) Agent: **NEFF, Gregor, N.**; Kramer Levin Naftalis & Frankel LLP, 919 Third Avenue, New York, NY 10022 (US).

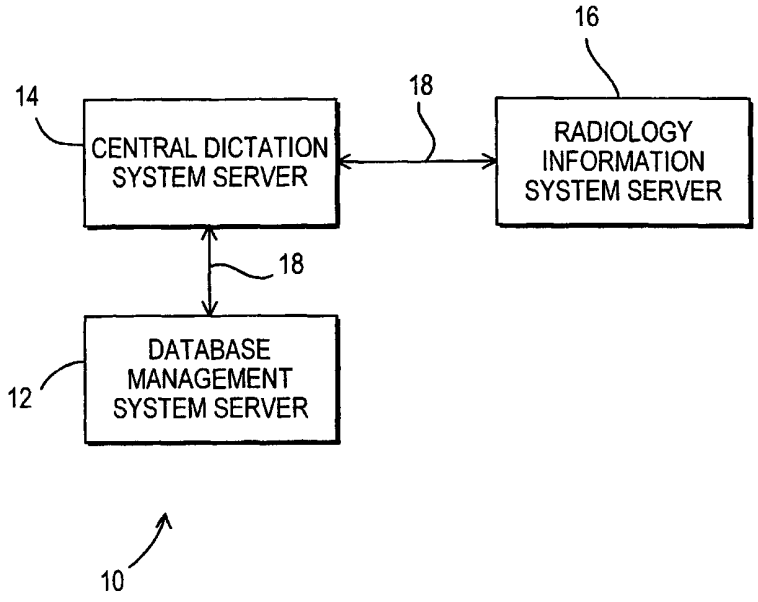
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE,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



(57) Abstract: A software system provides an interface between a radiology information system (16) and a central dictation system (14). The software includes first and second application modules which are both in communication with a database management system (12). Other components of the software system include protocole DLL's (dynamic linked libraries) (28) and a communication DLL (26). Each of the software components maintains a separate trace buffer. An event occurring in any one of the components triggers a dump of all of the trace buffers. A server socket object is maintained in existence and placed in an accept mode when a client socket terminates communication with the server socket object.

WO 01/28211 A1

SYSTEM AND METHOD FOR INTERFACING A RADIOLOGY INFORMATION SYSTEM TO A CENTRAL DICTATION SYSTEM

FIELD OF THE INVENTION

The present invention relates to information systems and, more particularly, to an interface between information systems found within one organization and the storage capacities of such systems.

BACKGROUND OF THE INVENTION

Central dictation systems are well known and are a convenient and efficient way of arranging for the transcription of dictated information and producing corresponding documents in organizations that have numerous users who have need of transcription services. Central dictation systems are particularly common in hospitals and other large medical facilities, because the great value of the doctors' time requires that documentation be produced as efficiently as possible.

With central dictation systems, a user dictates data into an input station, whereby the voice message is transmitted to a central server, which assigns a job number and other identifying characteristics to the dictated information. Usually the user enters identifying information such as a user i.d. number, a file number, a patient number and so forth. This may be done via a keypad at the input station and/or by scanning a bar code with a bar code reading wand attached to the input station. The raw data is then transmitted to a dictation queue, where it awaits transcription. After the document is transcribed, a message is returned to the user reporting that the document has been completed and the number by which the document can be accessed.

Many hospitals also have a radiology information system. The function of a radiology information system is to keep track of x-ray examinations, CAT scans and other radiological procedures prescribed by attending physicians. After each procedure is complete, the resulting films or other images are reviewed by a radiologist, who dictates a report into the central dictation system. When the report is transcribed, the transcribed document is stored in the radiology information system and made available to the attending physician. The radiology information system

assigns a transaction or "accession" number to each examination. The accession number is used to keep track of the x-rays themselves and the status of the examination and of the corresponding radiologist's report.

It has been known to provide a software interface between radiology information systems and central dictation systems to aid in transferring accession numbers and other information between the two systems. Other forms of cooperation between the two systems are also desirable. While the dictated report is in the queue awaiting transcription, the radiologist should be able to obtain information regarding its status, and should be able to dictate additional data regarding a particular patient under a new but related accession number. The radiologist should also be able to obtain information regarding the status of a document and related accession numbers.

Although the central dictation and radiology information systems function well independently, the current state of interfaces between the two systems leaves something to be desired. This is in part due to the limited storage capacity of the combined system.

It would be advantageous if the storage capacities of the central dictation system and radiology information system were increased.

It would also be desirable to improve error diagnosis features of software provided as an interface between the radiology information system and the dictation system.

Also, some radiology information systems disconnect from the interface at the end of each transaction. With known client-server socket protocols, this creates a large processing overhead for frequently instantiating and destroying server socket objects.

OBJECTS AND SUMMARY OF INVENTION

It is an object of the invention to provide a software interface between a radiology information system and the central dictation system so as to allow for an expanded data storage capacity.

It is a further object of the invention to provide a software interface between such systems in which more detailed diagnostic information is made available.

It is still a further object of the invention to increase the processing efficiency of such a software interface.

According to an aspect of the invention, a software system provides an interface between a radiology information system and a central dictation system, the software system including a first application module for exchanging data messages with the radiology information system, a second application module for exchanging data messages with the central dictation system and with the first application module; and a database management system, wherein the first application module includes procedures for accessing the database management system; and the second application module includes procedures for accessing the database management system.

According to further aspects of the invention, at least one of the first application module and the second application module programs a first server computer and the database management system programs a second server computer which is separate from, but interfaced to, the first server computer, wherein the first server computer includes a mass storage component in which the central dictation system stores voice files corresponding to dictation jobs. This permits storage of records related to many more accession numbers than has been possible in previous systems wherein central dictation systems had limited storage capabilities for accession numbers. The radiology information system may be maintained in a third server computer which is separate from the first and second servers but which is interfaced to the first server.

According to other aspects of the invention, the software system may include a first protocol DLL for transmitting data messages between the first application module and the central dictation system; a second protocol DLL; and a communication DLL for transmitting data messages between the second protocol DLL and the radiology information system; the second protocol DLL transmitting messages between the second application module and the communication DLL.

According to another aspect of the invention, a central dictation system is interfaced to a radiology information system by a method which includes the steps of providing a first application module for exchanging data messages with the radiology

information system; providing a second application module for exchanging data messages with the central dictation system and with the first application module; providing a database management system; transmitting queries from said first application module to said database management system; and transmitting queries from said second application module to said database management system.

According to yet another aspect of the invention, a software system includes multiple software modules, wherein a respective trace buffer is provided for each of said software modules, each trace buffer for storing a plurality of records and maintained in RAM and configured as a ring buffer, each of said records indicative of an instruction executed by the respective software module, wherein the respective contents of each trace buffer are dumped to a nonvolatile storage device in response to occurrence of an event in any one of the software modules. This aspect of the invention provides for more detailed and extensive information to be made available for diagnostic analysis in the event of a system failure.

According to still another aspect of the invention, there is provided a method of managing a software system in which messages are passed between a first process and a second process by means of a server socket in the first process and a client socket in the second process, the client socket for sending said messages and the server socket for receiving said messages, the method including the steps of: registering the server socket in the first process; instantiating the server socket in the first process; placing the server socket into an accept mode in which the server socket awaits initiation of communication by the client socket; and placing the server socket into a receive mode upon initiation of communication by the client socket, the server socket receiving at least one message from the client socket in the receive mode; wherein the server socket is maintained in existence and placed in the accept mode upon the client socket terminating communication with the server socket. This aspect of the invention increases processing efficiency in certain circumstances, particularly when the process which maintains the client socket frequently interrupts communication with the process which maintains the server socket.

Other objects, features and advantages of the present invention will become apparent from the subsequent more detailed description of preferred embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram representation of an information system in which the present invention is applied.

Fig. 2 is a functional block diagram showing software which provides an interface in accordance with the invention between the radiology information system and central dictation system shown in Fig. 1.

Figs. 3A and 3B illustrate in the form of a flow-chart a process according to which a radiology information system manager application module shown in Fig. 2 identifies the accession numbers for documents.

Figs. 4A and 4B illustrate in the form of a flow-chart a method by which a radiology information system protocol DLL module shown in Fig. 2 handles data objects.

Fig. 5 is a flow-chart illustration of how a dictation information system manager application module shown in Fig. 2 reconciles data received from the radiology information system.

Figs. 6A and 6B are a flow-chart illustration showing details of a reconciliation procedure.

Figs. 7A and 7B illustrate in the form of a flow-chart a process according to which a dictation information system manager application module shown in Fig. 2 identifies the accession numbers for documents.

Fig. 8 is a block diagram showing trace buffers maintained by the software modules of Fig. 2 in accordance with the invention.

Fig. 9 is a block diagram representation of a typical one of the trace buffers of Fig. 8.

Fig. 10 illustrates in the form of a flow chart a trace buffer life cycle.

Fig. 11 illustrates in the form of a flow chart dumping of all information from all trace buffers into a permanent storage device upon the occurrence of a trigger event.

Fig. 12 illustrates in the form of a flow chart a process for managing a socket object in accordance with the invention.

DESCRIPTION OF PREFERRED EMBODIMENTS

System Overview

A preferred embodiment of the invention will now be described, initially with reference to Fig. 1. In Fig. 1, reference numeral 10 generally indicates the hardware aspects of an information system in which the invention may be applied. The major components of the central dictation system 10 are a database management system server computer 12, a central dictation system server computer 14, a radiology information system server computer 16, and a communications network 18 that connects the central dictation system server 14 and the radiology information system server 16, and the central dictation system server 14 and the database management system server 12.

All of the components shown in Fig. 1 may be constituted by standard hardware. Preferably the communication network 18 is a conventional local area network (LAN) and the servers 12, 14, 16 are conventional computing devices.

Fig. 2 is a block diagram showing interface software modules provided in accordance with the invention. This software may be resident in the central dictation system server 14 shown in Fig. 1. Alternatively, the interface software may be resident on a separate computer (not shown) connected for data communication with each of the servers 12, 14, 16. As other alternatives, the interface software may be resident on the DBMS server 12 or the RIS server 16 (in which cases a data communication path should be provided between DBMS server 12 and RIS server 16). It is also contemplated to divide the interface software between two or more computers. Furthermore, all of the functions of the system shown in Fig. 1 may be consolidated in a single server, or the functions of any two of the three servers may be combined in one server.

Interface Software Modules

The software illustrated in Fig. 2 is generally indicated by reference numeral 24, and provides an interface between the application software which controls the

central dictation system and the application software which controls the radiology information system. Both of these applications may be conventional. The interface software 24 is made up of a radiology to information system interface portion 20 and a central dictation system interface portion 22. The radiology information system interface portion 20 contains three components, namely a communications dynamic linked library (DLL) 26, a protocol DLL 28, and a manager application 30. The comm. DLL 26 unpacks from the communication system format messages received from the radiology information system (RIS). The comm. DLL also translates into the communication systems format messages to be sent to the RIS. The protocol DLL 28 converts messages between a raw data format and a format prescribed by a messaging protocol used by the RIS.

The manager application 30 handles messages and exchanges information with the central dictation system interface portion 22. Functions performed by the manager application 30 will be described in greater detail below.

The central dictation system interface portion 22 includes two components: a protocol DLL 34 and a manager application 32. No comm. DLL is required in the interface portion 22 so long as the interface software 24 is resident in the same computer with the central dictation system.

The protocol DLL 34 of the dictation system interface portion 22 converts messages between a raw data format and a format prescribed by a messaging protocol used by the central dictation system. The manager application 32 of the dictation system interface portion 22 handles messages and exchanges information with the RIS interface portion 20. Functions performed by the application 32 will be described in more detail below.

Both the radiology information system interface portion 20 (particularly application 30) and the central dictation system interface portion 22 (particularly application 32) are in communication with the database management system 36. The DBMS 36 stores information relating to radiological examinations and related reports processed through the RIS and the central dictation system. The database management system 36 is preferably resident on its own server 12 (Fig. 1), so that it

has a storage capacity that is theoretically unlimited. In practice, of course, the storage capacity of the DBMS 36 is limited to that of the server 12 and any associated memory units.

Figs. 3A and 3B illustrate in the form of a flowchart processes carried out by the radiology information system manager application module 30. In an initial step 40 of Fig. 3A, the radiology information system manager application module 30 receives patient information in the form of a message passed from the radiology information system. The radiology information system manager application module 30 then sends a query (indicated at step 42) to the database management system 36 regarding whether corresponding patient information, including an accession number, exists within the system. The database 36 is examined for a matching accession number (step 44). If a matching accession number is not found, the information from the message is stored to data tables, as indicated at step 46, and the application resets itself to a ready condition (step 48).

If a matching accession number is found at step 44, the application determines the type of the accession number, as indicated by step 52, and takes appropriate action.

Fig. 3B illustrates an event handling routine carried out by the manager application 30. Initially, as indicated at step 54, the application determines what kind of event has occurred. If the accession number has been previously entered (step 56), the data table is updated with the patient's new information (step 58) and the event is written to the trace log (step 60).

If an accession number was not previously entered and has not been queried for, the radiology information system manager application queries for the accession number (step 62), the system records that an accession number was entered during dictation (step 64), and the event is written to the trace log (step 66).

If an accession number has been queried for and dictated against (step 68), the RIS manager application records that an accession number was entered during dictation (step 70), and writes the event to the trace log (step 72). The application also sends a reconciliation message to the dictation information system manager

application 32 (step 74), and finally sends a request for dictation to the radiology information system protocol DLL 28 (step 76).

In the event a message is obsolete and waiting to be purged, as indicated at step 78, the RIS manager application 30 updates the data table entry with all relevant patient information (step 80), writes the event to the trace log (step 82), and sends a new message indicating that an accession number had been previously entered (step 84).

Finally, if the application cannot locate a matching accession number (step 86), it writes the event to the trace log (step 88) and then issues an error message (step 90).

Figs. 4A and 4B illustrate in the form of a flowchart a process by which the radiology information system protocol DLL 28 handles data objects. Once the radiology information system DLL receives a message (step 92), it determines the message type (step 94). If it cannot identify the message (step 112), it issues an error message (step 114). If the message indicates an idle condition (step 108), the DLL simply acknowledges the message (step 110).

With regard to three types of messages, the radiology information system protocol DLL 28 must verify the format of the data object, as indicated at step 116. This occurs in the event of the type of message indicating that an examination has been completed (step 96), in which case an "examination complete" entry is made in the log with patient and exam information (step 98); or in the event of a "report finalized" message (step 100), the event is logged (step 102), or if the message requires a response (step 104), a "response required" entry is made in the log (step 106).

Once the format of the data object has been verified at step 116, the radiology information system protocol DLL 28 validates the message field (step 118). If the field is found not to be valid at step 120, the DLL returns an error message (step 114). If the field is valid, the radiology information system DLL 28 proceeds to handle the data object as illustrated in Fig. 4B.

Initially, the DLL unpacks the data object, as indicated at step 122. It then converts the data to a raw data string (step 124), and sends the raw data to the radiology information system manager application 30 (step 126). Thereafter, the DLL logs the activity (step 128) and sends an acknowledgement message to the RIS (step 130).

Fig. 5 is a flow-chart illustration of how the dictation information system manager application module 32 as shown in Fig. 2 reconciles data received from the radiology information system interface portion 20. As step 132, the dictation information system manager application module 32 receives raw patient data from the radiology information system manager application 30. A reconciliation procedure is then invoked at step 134 and determined to be complete at step 136.

Figs. 6A and 6B together form a flow-chart illustration of some details of the reconciliation procedure. Step 138 indicates invocation of the reconciliation procedure. The manager application then sends an inquiry to the database 36 to determine the accession number which corresponds to the patient information (step 140). If the accession number is found (step 142), the reconciliation procedure is complete (step 144).

If the accession number is not found at step 142, a recovery routine is invoked at step 148. After a pause 150, the application retries the reconciliation request (step 152). If such request is found to be unsuccessful at step 156, the routine loops back to step 150. If the reconciliation request is found to be successful at step 156, the system runs a recover program (step 158, Fig. 6B). The database 36 is then queried for the accession number (step 160). Next the table entry is updated with new patient information (step 162) and the event is written to the trace log (step 164).

Figs. 7A and 7B illustrate in the form of a flow-chart a process according to which the dictation information system manager application module 32 identifies relevant existing information, such as accession numbers, for dictated documents. In an initial step 166 of Fig. 7A, the dictation information system manager application module 32 receives patient information in the form of a message from the dictation system. The dictation information system manager application module 32 then sends

a query (step 168) to the database management system 36 regarding whether a corresponding accession number exists within the system. If a matching accession number is not found at step 170, the raw information is stored to data tables (step 174) and the application resets itself to a ready condition (step 176).

If a matching accession number is found at step 170, the application determines the type of the accession number (step 178) and takes appropriate action. Fig. 7B illustrates an event handling routine carried out by the manager application 32. Initially as indicated at step 180, the application determines what kind of event has occurred. If the accession number has been previously entered (step 182), the data table is updated with the patient's new information (step 184) and the event is written to the trace log (step 186).

If an accession number was not previously entered and has not been queried for, the dictation system manager application 32 queries for a matching accession number (step 188), the system records that an accession number was entered during dictation (step 190), the event is written to the trace log (step 192), and a reconciliation message is sent to the radiology information system (step 194).

If an accession number has been queried for and dictated against (step 196), the manager application 32 records that an accession number was entered during dictation (step 198), and writes the event to the trace log (step 200). The manager application 32 also sends a reconciliation message to the dictation information system application (step 202), and finally sends a request for dictation to the dictation information system protocol DLL 34 (step 204).

In the event a message is obsolete and waiting to be purged, as indicated at step 206, manager application 32 updates the data table entry with all relevant patient information (step 208), writes the event to the trace log (step 210), and sends a message indicating that an accession number had been previously entered (step 212).

Finally, if the application cannot locate a matching accession number (step 214), it writes the event to the trace log (step 216) and then issues an error message (step 218).

Trace Buffers

Fig. 8 schematically illustrates another aspect of the invention. According to this aspect, respective trace buffers 222a - 222e are maintained by each of the software components shown in Fig. 8 (and Fig. 2), namely the radiology information system communications DLL 26, the radiology information system protocol DLL 28, the radiology information system manager application 30, the dictation information system manager application 32, and the dictation information system protocol DLL 34, respectively. It is contemplated that the trace buffers be maintained in RAM.

A typical one of the trace buffers is schematically illustrated in Fig. 9. As shown in Fig. 9, each trace buffer 222 is preferably configured as a ring buffer which can store a plurality of records 224-0 to 224-n in respective storage locations 225-0 to 225-n. In a preferred embodiment the number of storage locations is 100 in each buffer, but of course the number of storage locations provided may be more or less than 100. Since the trace buffer is a ring buffer, when all the storage locations are full, the next record to be stored replaces the oldest record in the buffer.

Fig. 10 illustrates the life cycle of one of the trace buffers 222. A trace buffer is initialized (step 226) to a condition in which it is ready to store data. A message to be stored in the trace buffer is received (step 228) and it is then determined, at step 230, whether the trace buffer is available for access. If so, the message is written into the next storage location in the buffer (step 240). Otherwise, the routine idles until access to the trace buffer is allowed.

After step 240, it is determined, at step 242, whether the message represents an event which should trigger permanent storage of the trace buffer. If not, the buffer is returned to the ready condition.

However, if it was found at step 242 that the message was indicative of a trigger event, then step 246 follows. (Examples of trigger events are a system failure, a fatal error in the software component which maintains the trace buffer, or a fatal error in another one of the software components.) At step 246 the trigger event is logged to an index line. Then, the 5 contents of the entire log of the trace buffer are

dumped (step 248) to a nonvolatile storage device 220 (Fig. 8). The nonvolatile storage device 220 may be, for example, a hard disk.

The trace buffer is then cleared (step 250), and the trace buffer is reinitialized.

Fig. 11 portrays, in flow chart form, further events that occur after a trigger event is logged to any one of the trace buffers 222a-222e. As seen from Fig. 11, it is determined at step 244 whether a trigger event has occurred. If so the trigger event is logged in an index line of each of the trace buffers of the respective software components (steps 246, 254, 256) and then the contents of the trace buffers are dumped to nonvolatile store 220 (steps 248, 260, 262).

In accordance with this aspect of the invention, a respective trace buffer is maintained by each one of several individual software components, and a trigger event which originates in any one of the software components, triggers a buffer dump from all of the components. Consequently, enhanced and more comprehensive diagnostic procedures are possible, since the activities of each component are stored for later analysis.

Socket Management

Fig. 12 illustrates another aspect of the invention. According to this aspect, the overhead processing required for implementing a client-server socket model is reduced for certain environments in which the interface software of the present invention may be used.

In Fig. 12, block 264 as a whole represents an initialization phase for a server socket object. In accordance with conventional practice (as embodied, for example, in the Windows NT operating system), initialization includes the steps of registering a server socket object (step 266), creating or instantiating the server socket object (step 268) and binding the server socket object (step 270). Then, also according to known procedures, the server socket is placed in an "accept" mode (step 272), indicating that the socket is awaiting connection from a client socket.

The procedure of Fig. 12 idles, as indicated at block 274, until the client socket initiates communication. Once communication is initiated, the server socket is placed in a "receive" mode (step 278), wherein the server socket is enabled to receive

information from the client. If the client disconnects (branch 280), the server socket is returned to the "accept" mode of step 272. If the client does not disconnect (branch 282), but there is no message from the client, the server socket idles in a loop made up of blocks 278, 283, 284.

If at block 284 a message is received from the client, the server socket receives and processes the message string (step 288). Once the string has been received and processed at block 288, it is determined once again whether the client has disconnected (step 290) or if the communication is broken (step 292). If either one of these occurs, the server socket is returned to the accept mode of block 272. Otherwise the socket continues in the receive mode of block 278.

The procedure described above in connection with Fig. 12 departs from the conventional practice for client/server sockets in that the server socket is maintained in existence and returned to an accept mode when the client socket disconnects or the physical communication channel is shut down or disrupted. By contrast, conventional practice calls for the server socket to be eliminated (de-instantiated) upon the occurrence of either of these two events. The present inventors have found that the conventional practice results in an undesirable amount of processing overhead, including frequent destruction and instantiation of server socket objects, in cases where the RIS to which the interface is made frequently disconnects from the interface software or the communication channel frequently shuts down. For example, some RIS software is arranged to disconnect from the server socket in the interface after each transaction. Since the number of transactions per day may be in the thousands, the improved procedure of the present invention may substantially reduce the processing burden by maintaining the server socket in existence even after the client has disconnected.

It is to be understood that the above description may indicate to those skilled in the art additional ways in which the principles of this invention may be used without departing from the spirit of the invention. The particularly preferred embodiments of the apparatus are thus intended in an illustrative and not limiting sense. The true spirit and scope of the invention are set forth in the following claims.

WHAT IS CLAIMED IS:

1. A software system for providing an interface between a radiology information system and a central dictation system, said software system including:

a first application module for exchanging data messages with the radiology information system;

a second application module for exchanging data messages with the central dictation system and with the first application module; and

a database management system; wherein:

the first application module includes a procedure for accessing the database management system; and

the second application module includes a procedure for accessing the database management system.

2. A software system according to claim 1, wherein at least one of the first application module and the second application module programs a first computer device and the database management system programs a second computer device which is separate from, but interfaced to, the first computer device.

3. A software system according to claim 2, wherein the first computer device includes a mass storage component in which the central dictation system stores voice files corresponding to dictation jobs.

4. A software system according to claim 3, wherein the radiology information system is maintained in a third computer device which is separate from the first and second computer devices but which is interfaced to the first computer device.

5. A software system according to claim 1, further comprising:

a first protocol DLL for transmitting data messages between the first application module and the central dictation system;

a second protocol DLL; and

a communication DLL for transmitting data messages between the second protocol DLL and the radiology information system; the second protocol DLL transmitting messages between the second application module and the communication DLL.

6. A method of interfacing a central dictation system to a radiology information system, the method comprising the steps of:

providing a first application module for exchanging data messages with the radiology information system;

providing a second application module for exchanging data messages with the central dictation system and with the first application module;

providing a database management system;

transmitting queries from said first application module to said database management system; and

transmitting queries from said second application module to said database management system.

7. A software system including multiple software modules, wherein a respective trace buffer is provided for each of said software modules, each trace buffer for storing a plurality of records, each of said records indicative of an instruction executed by the respective software module.

8. A software system according to claim 7, wherein each of said trace buffers is maintained in RAM.

9. A software system according to claim 8, wherein each of said trace buffers is configured as a ring buffer.

10. A software system according to claim 8, wherein the respective contents of each trace buffer are dumped to a nonvolatile storage device in response to occurrence of an event in any one of the software modules.

11. A method of managing a software system in which messages are passed between a first process and a second process by means of a server socket in the first process and a client socket in the second process, the client socket for sending said messages and the server socket for receiving said messages, the method comprising the steps of:

registering the server socket in the first process;

instantiating the server socket in the first process;

placing the server socket into an accept mode in which the server socket awaits initiation of communication by the client socket; and

placing the server socket into a receive mode upon initiation of communication by the client socket, the server socket receiving at least one message from the client socket in the receive mode;

wherein the server socket is maintained in existence and placed in the accept mode upon the client socket terminating communication with the server socket.

12. In a method of managing a software system in which messages are passed between a first process and a second process by means of a server socket in the first process and a client socket in the second process, the client socket for sending said messages and the server socket for receiving said messages, wherein the method comprises registering the server socket in the first process, instantiating the server socket in the first process, placing the server socket into an accept mode in which the server socket awaits initiation of communication by the client socket, and placing the server socket into a receive mode upon initiation of communication by the client socket, the server socket receiving at least one message from the client socket in the receive mode; the improvement wherein the server socket is maintained in existence and placed in the accept mode upon the client socket terminating communication with the server socket.

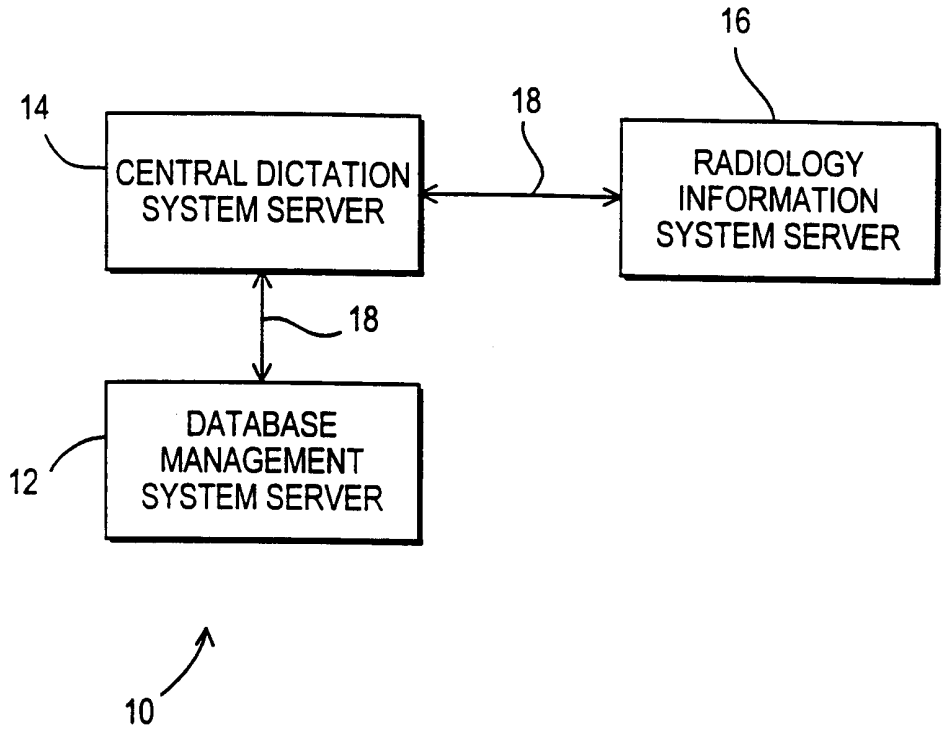

16
14
18
CENTRAL DICTATION SYSTEM SERVER
RADIOLOGY INFORMATION SYSTEM SERVER
18
DATABASE MANAGEMENT SYSTEM SERVER
12
10

FIG. 1

FIG. 2

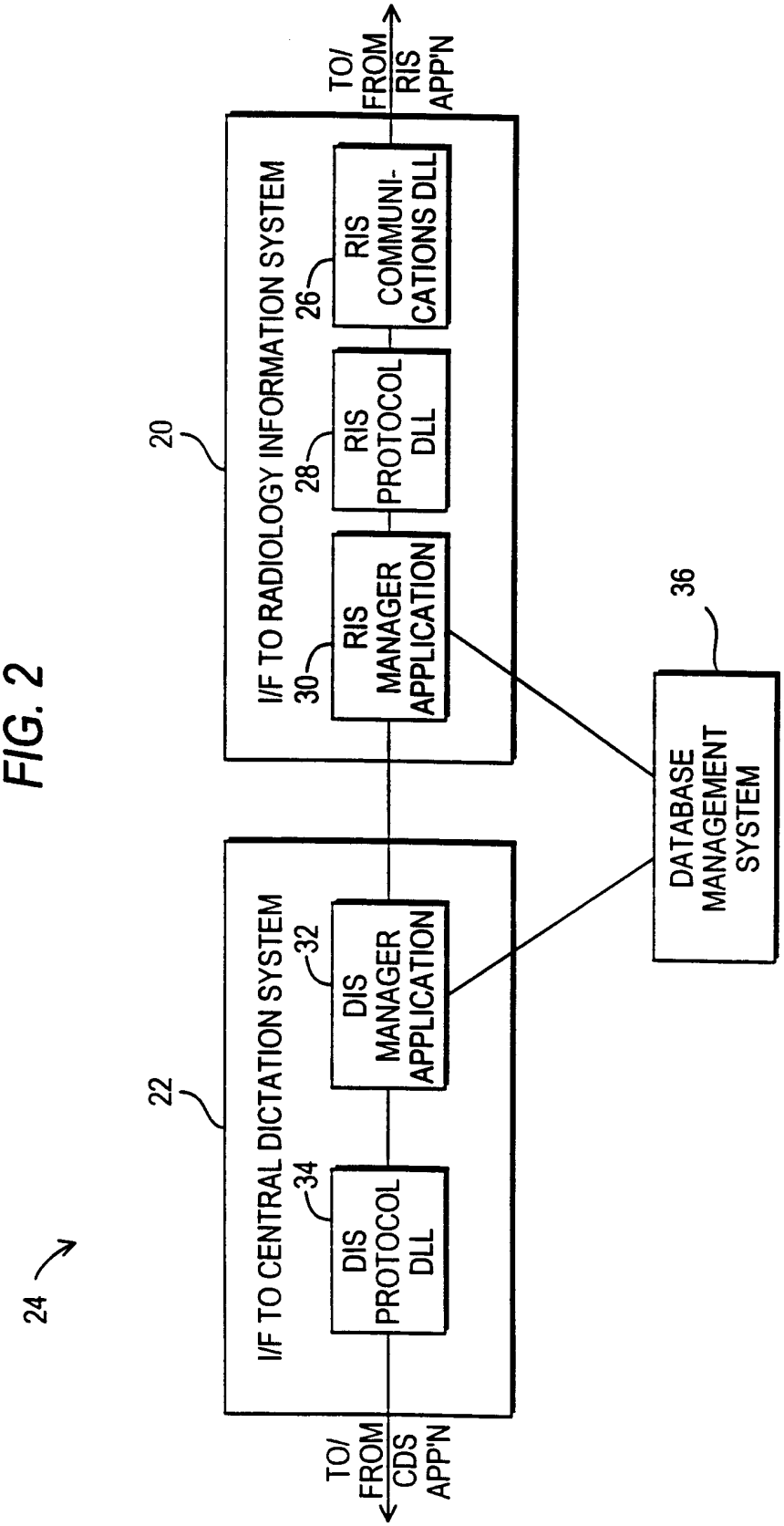
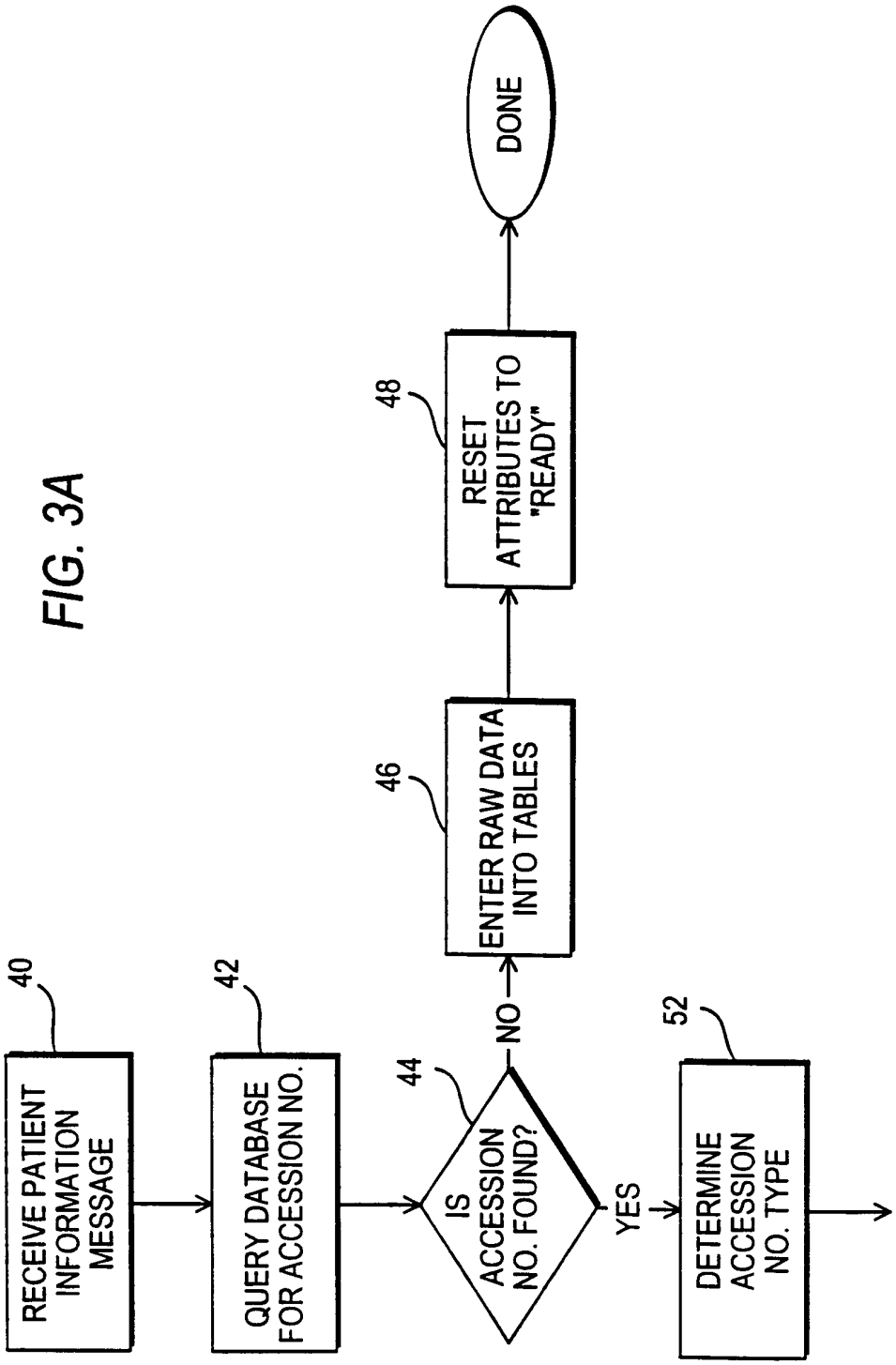

24
I/F TO CENTRAL DICTATION SYSTEM
22
DIS PROTOCOL DLL
34
DIS MANAGER APPLICATION
32
TO/ FROM CDS APP'N
I/F TO RADIOLOGY INFORMATION SYSTEM
20
RIS MANAGER APPLICATION
30
RIS PROTOCOL DLL
28
RIS COMMUNI-CATIONS DLL
26
TO/ FROM RIS APP'N
DATABASE MANAGEMENT SYSTEM
36

FIG. 3A


RECEIVE PATIENT INFORMATION MESSAGE
40
QUERY DATABASE FOR ACCESSION NO.
42
IS ACCESSION NO. FOUND?
44
NO
YES
ENTER RAW DATA INTO TABLES
46
RESET ATTRIBUTES TO "READY"
48
DONE
DETERMINE ACCESSION NO. TYPE
52

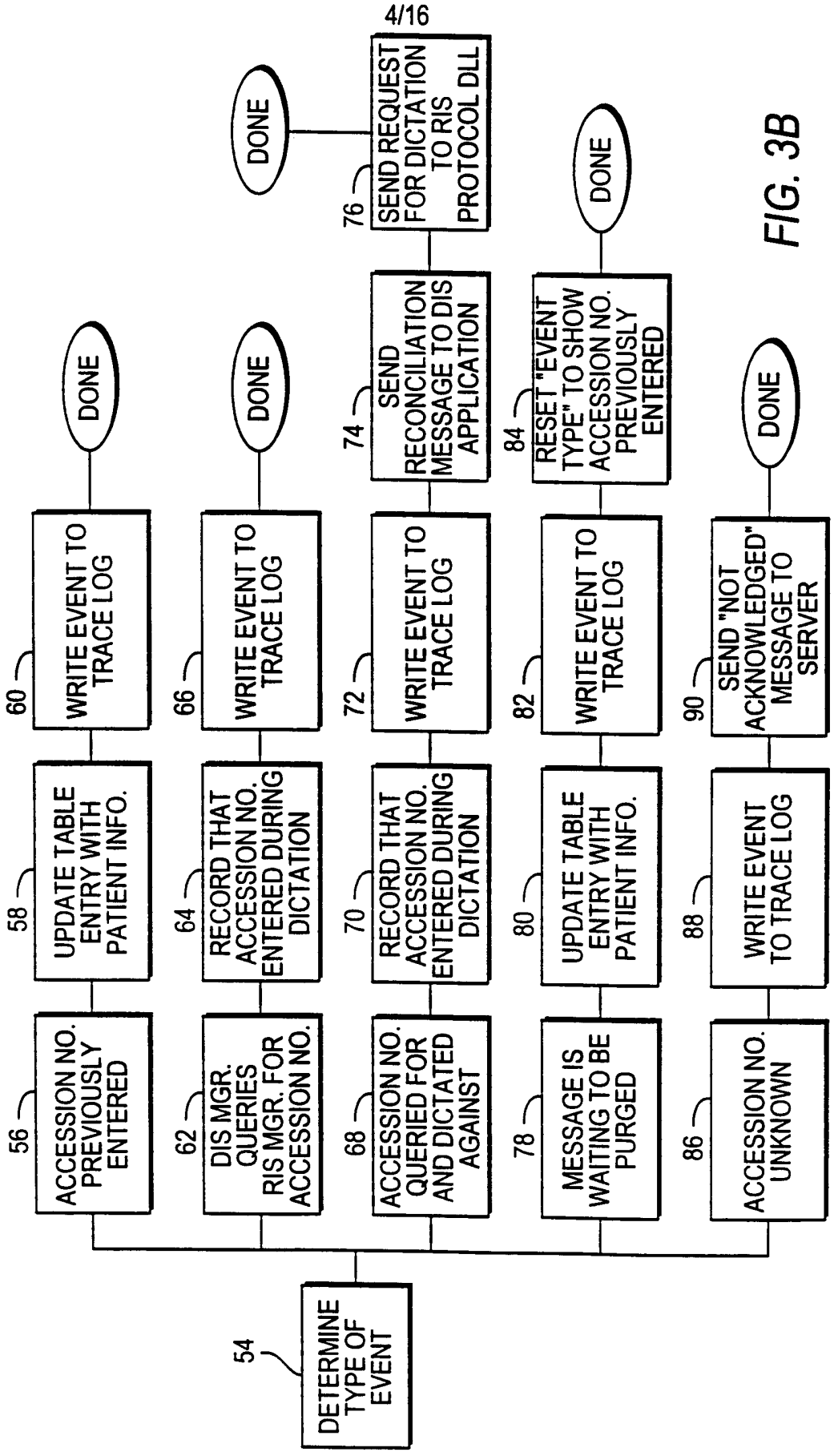

54
DETERMINE TYPE OF EVENT
56
ACCESSION NO. PREVIOUSLY ENTERED
58
UPDATE TABLE ENTRY WITH PATIENT INFO.
60
WRITE EVENT TO TRACE LOG
DONE
62
DIS MGR. QUERIES RIS MGR. FOR ACCESSION NO.
64
RECORD THAT ACCESSION NO. ENTERED DURING DICTATION
66
WRITE EVENT TO TRACE LOG
DONE
68
ACCESSION NO. QUERIED FOR AND DICTATED AGAINST
70
RECORD THAT ACCESSION NO. ENTERED DURING DICTATION
72
WRITE EVENT TO TRACE LOG
74
SEND RECONCILIATION MESSAGE TO DIS APPLICATION
76
SEND REQUEST FOR DICTATION TO RIS PROTOCOL DLL
DONE
78
MESSAGE IS WAITING TO BE PURGED
80
UPDATE TABLE ENTRY WITH PATIENT INFO.
82
WRITE EVENT TO TRACE LOG
84
RESET "EVENT TYPE" TO SHOW ACCESSION NO. PREVIOUSLY ENTERED
DONE
86
ACCESSION NO. UNKNOWN
88
WRITE EVENT TO TRACE LOG
90
SEND "NOT ACKNOWLEDGED" MESSAGE TO SERVER
DONE

FIG. 3B

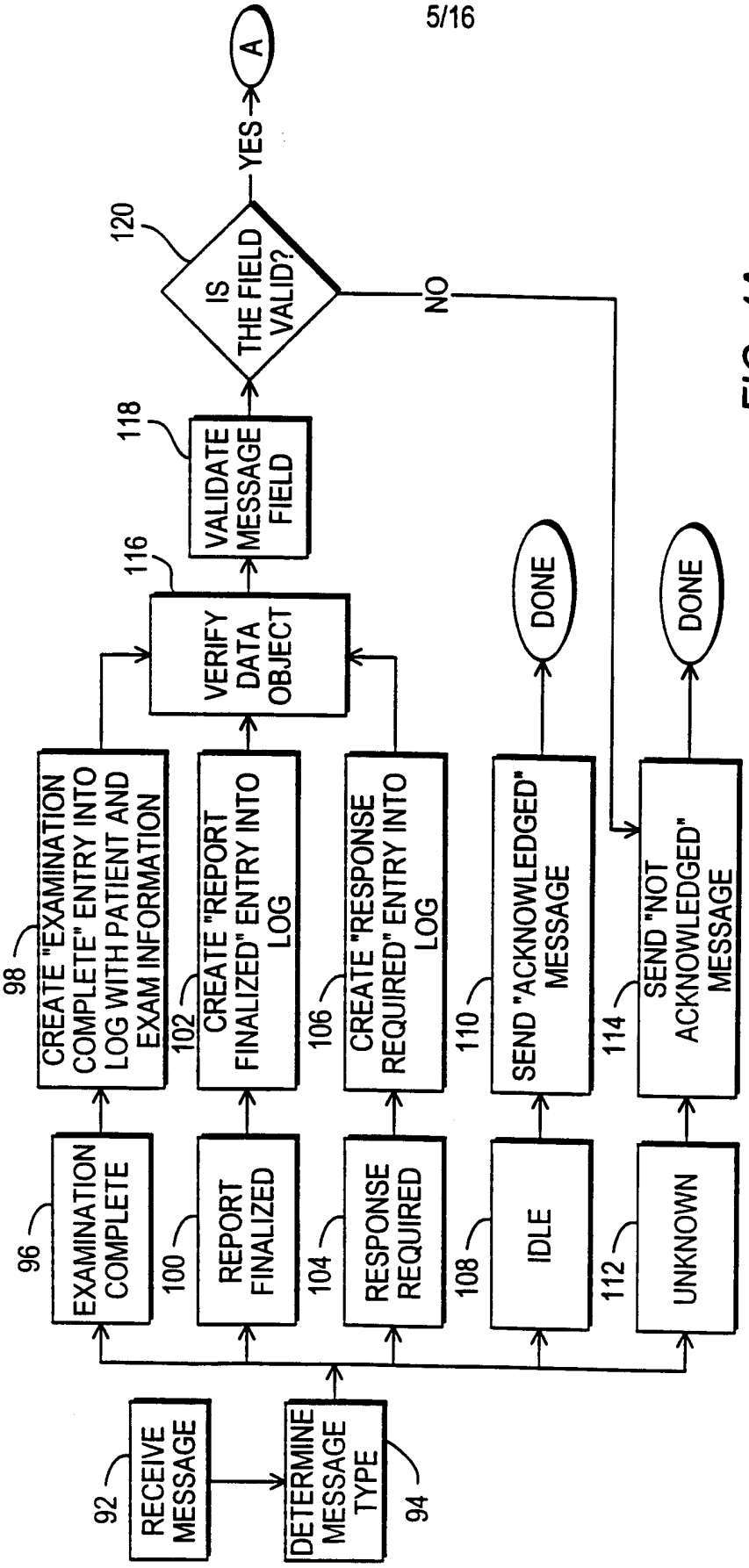

RECEIVE MESSAGE
92
DETERMINE MESSAGE TYPE
94
EXAMINATION COMPLETE
96
REPORT FINALIZED
100
RESPONSE REQUIRED
104
IDLE
108
UNKNOWN
112
CREATE "EXAMINATION COMPLETE" ENTRY INTO LOG WITH PATIENT AND EXAM INFORMATION
98
CREATE "REPORT FINALIZED" ENTRY INTO LOG
102
CREATE "RESPONSE REQUIRED" ENTRY INTO LOG
106
SEND "ACKNOWLEDGED" MESSAGE
110
SEND "NOT ACKNOWLEDGED" MESSAGE
114
DONE
DONE
VERIFY DATA OBJECT
116
VALIDATE MESSAGE FIELD
118
IS THE FIELD VALID?
120
YES
A
NO

FIG. 4A

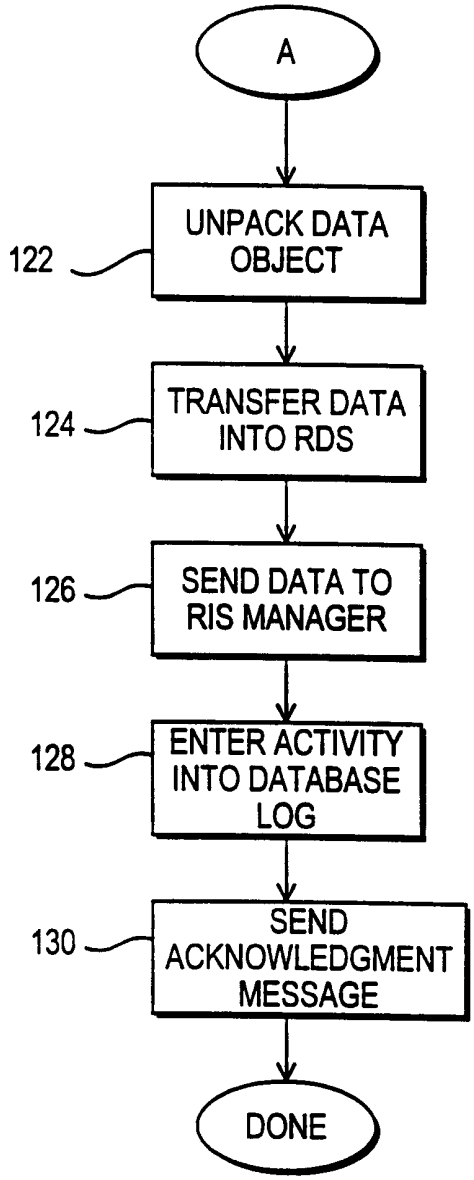

A
UNPACK DATA OBJECT
122
TRANSFER DATA INTO RDS
124
SEND DATA TO RIS MANAGER
126
ENTER ACTIVITY INTO DATABASE LOG
128
SEND ACKNOWLEDGMENT MESSAGE
130
DONE

FIG. 4B

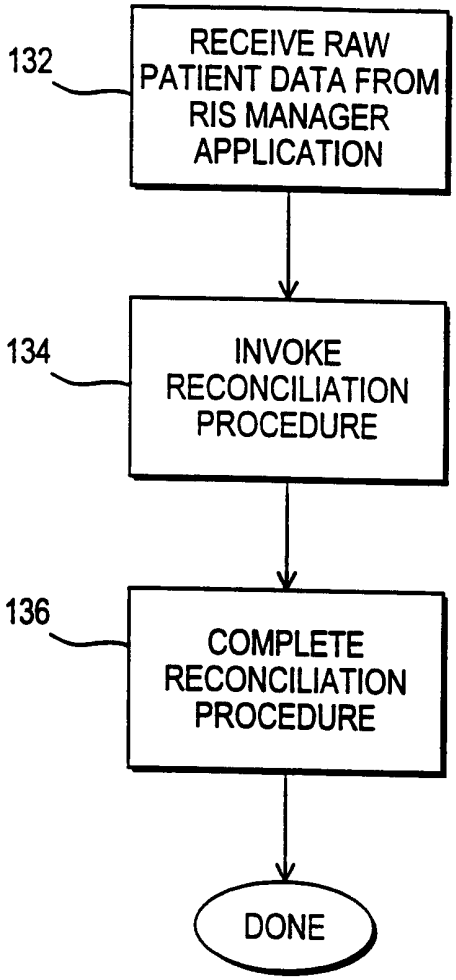

132
RECEIVE RAW PATIENT DATA FROM RIS MANAGER APPLICATION
134
INVOKE RECONCILIATION PROCEDURE
136
COMPLETE RECONCILIATION PROCEDURE
DONE

FIG. 5

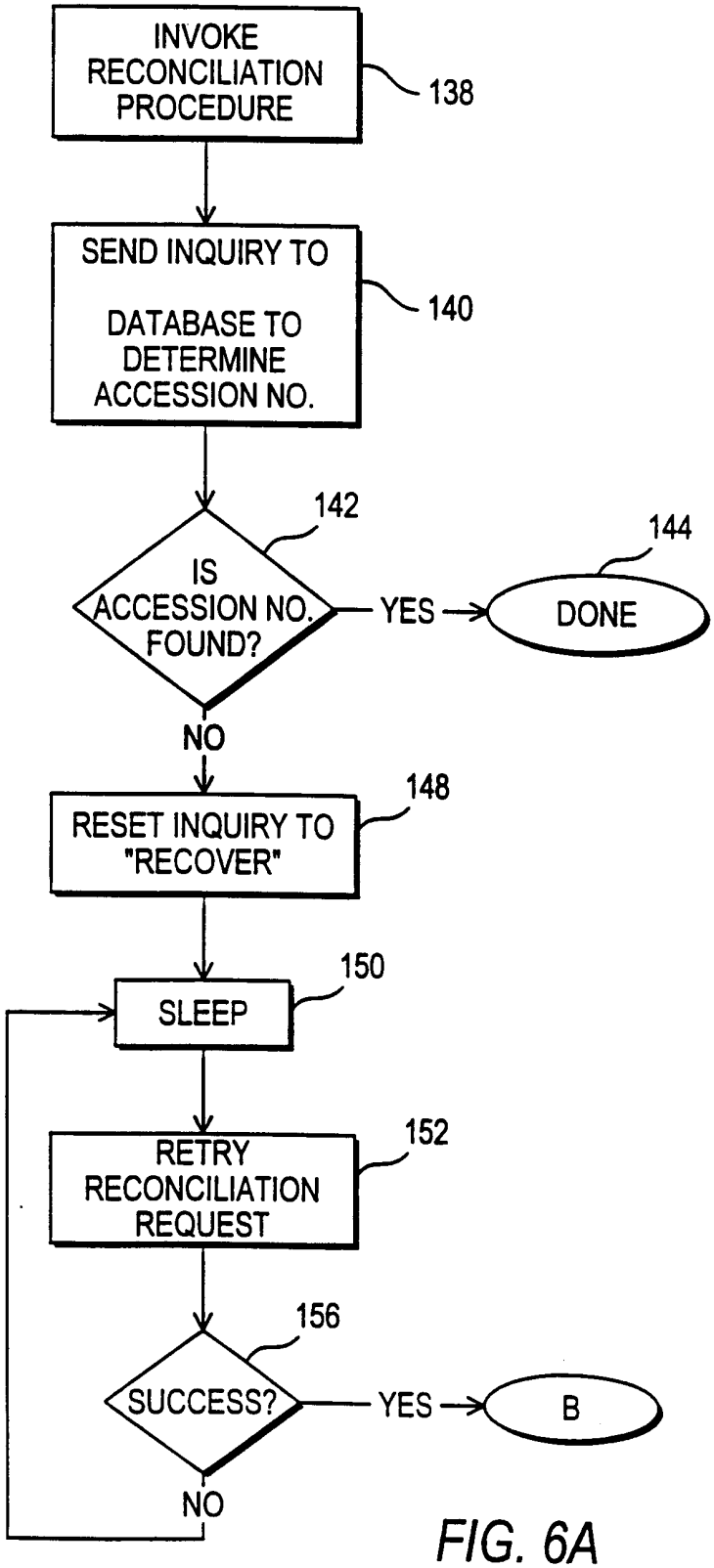

INVOKE RECONCILIATION PROCEDURE
138
SEND INQUIRY TO DATABASE TO DETERMINE ACCESSION NO.
140
142
IS ACCESSION NO. FOUND?
YES
144
DONE
NO
148
RESET INQUIRY TO "RECOVER"
150
SLEEP
152
RETRY RECONCILIATION REQUEST
156
SUCCESS?
YES
B
NO

FIG. 6A

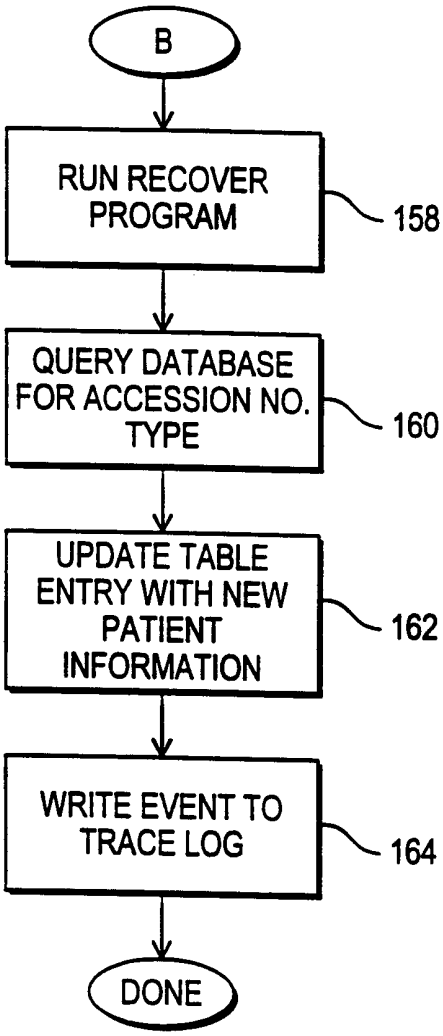
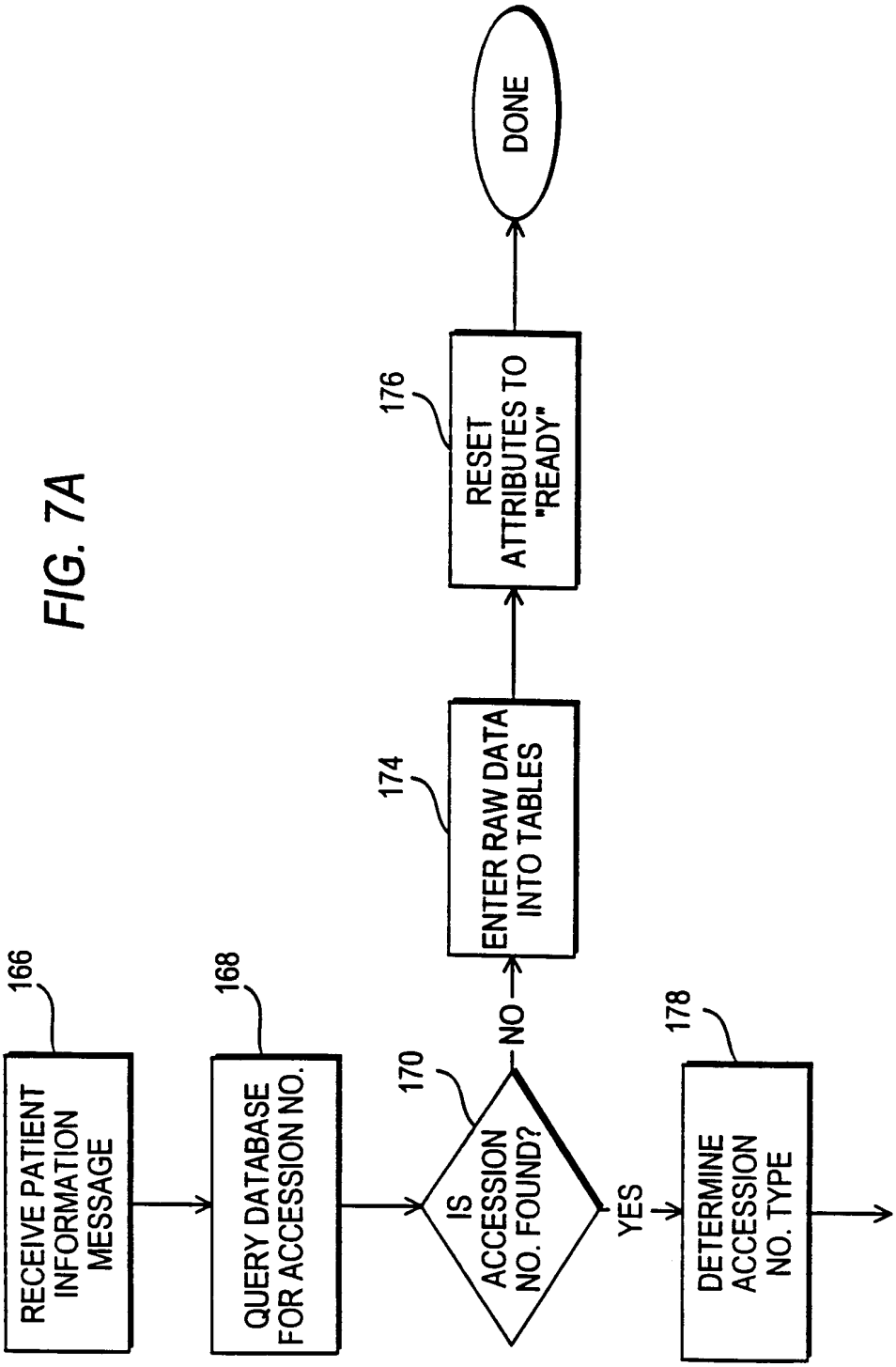

B
RUN RECOVER PROGRAM
158
QUERY DATABASE FOR ACCESSION NO. TYPE
160
UPDATE TABLE ENTRY WITH NEW PATIENT INFORMATION
162
WRITE EVENT TO TRACE LOG
164
DONE

FIG. 6B

FIG. 7A


166
RECEIVE PATIENT INFORMATION MESSAGE
168
QUERY DATABASE FOR ACCESSION NO.
170
IS ACCESSION NO. FOUND?
NO
YES
174
ENTER RAW DATA INTO TABLES
176
RESET ATTRIBUTES TO "READY"
178
DETERMINE ACCESSION NO. TYPE
DONE

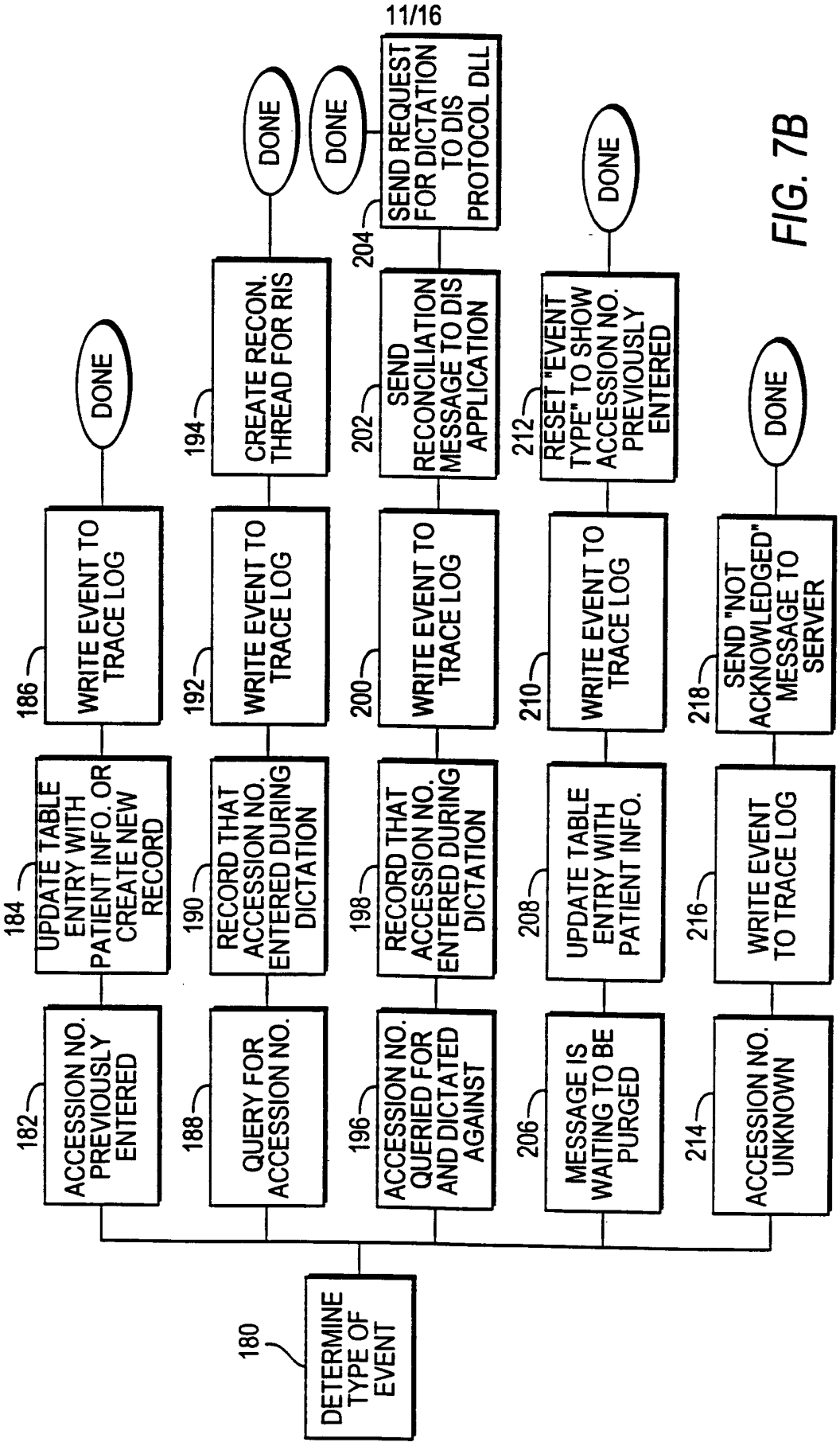
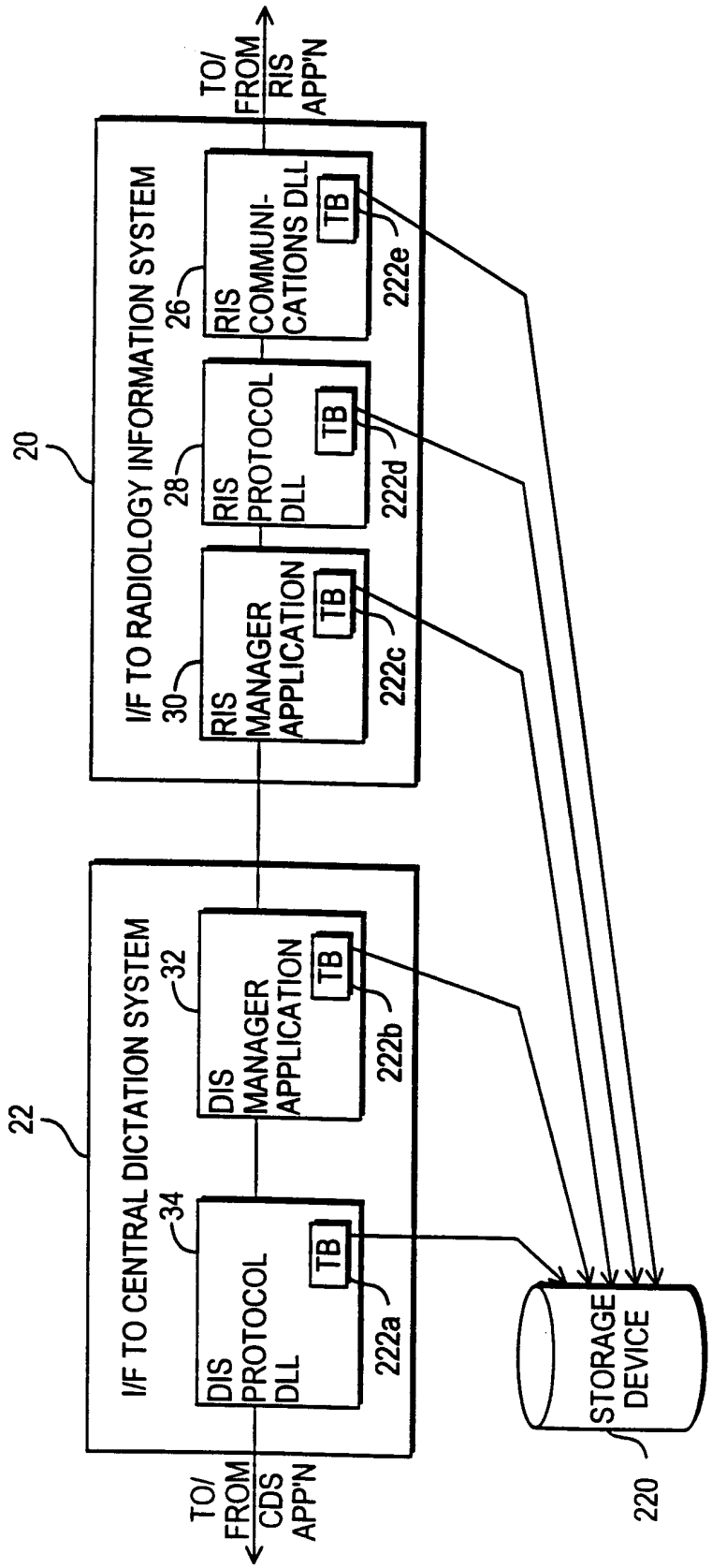

DETERMINE TYPE OF EVENT
180
ACCESSION NO. PREVIOUSLY ENTERED
182
UPDATE TABLE ENTRY WITH PATIENT INFO. OR CREATE NEW RECORD
184
WRITE EVENT TO TRACE LOG
186
DONE
QUERY FOR ACCESSION NO.
188
RECORD THAT ACCESSION NO. ENTERED DURING DICTATION
190
WRITE EVENT TO TRACE LOG
192
CREATE RECON. THREAD FOR RIS
194
DONE
ACCESSION NO. QUERIED FOR AND DICTATED AGAINST
196
RECORD THAT ACCESSION NO. ENTERED DURING DICTATION
198
WRITE EVENT TO TRACE LOG
200
SEND RECONCILIATION MESSAGE TO DIS APPLICATION
202
SEND REQUEST FOR DICTATION TO DIS PROTOCOL DLL
204
DONE
MESSAGE IS WAITING TO BE PURGED
206
UPDATE TABLE ENTRY WITH PATIENT INFO.
208
WRITE EVENT TO TRACE LOG
210
RESET "EVENT TYPE" TO SHOW ACCESSION NO. PREVIOUSLY ENTERED
212
DONE
ACCESSION NO. UNKNOWN
214
WRITE EVENT TO TRACE LOG
216
SEND "NOT ACKNOWLEDGED" MESSAGE TO SERVER
218
DONE

FIG. 7B

FIG. 8


TO/ FROM RIS APP'N
I/F TO RADIOLOGY INFORMATION SYSTEM
20
RIS MANAGER APPLICATION
30
RIS PROTOCOL DLL
28
RIS COMMUNI-CATIONS DLL
26
TB
222c
222d
222e
I/F TO CENTRAL DICTATION SYSTEM
22
DIS MANAGER APPLICATION
32
DIS PROTOCOL DLL
34
222b
222a
STORAGE DEVICE
220
TO/ FROM CDS APP'N

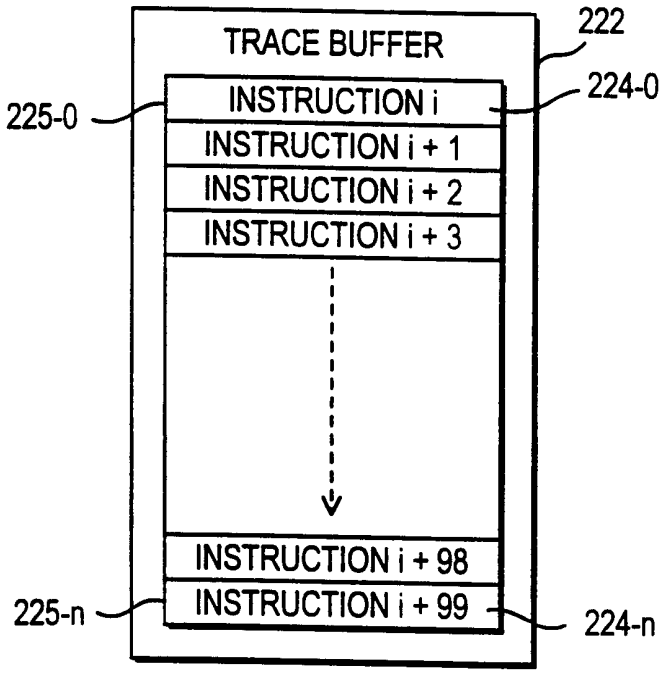

TRACE BUFFER
222
INSTRUCTION i
INSTRUCTION i + 1
INSTRUCTION i + 2
INSTRUCTION i + 3
INSTRUCTION i + 98
INSTRUCTION i + 99
225-0
224-0
225-n
224-n

FIG. 9

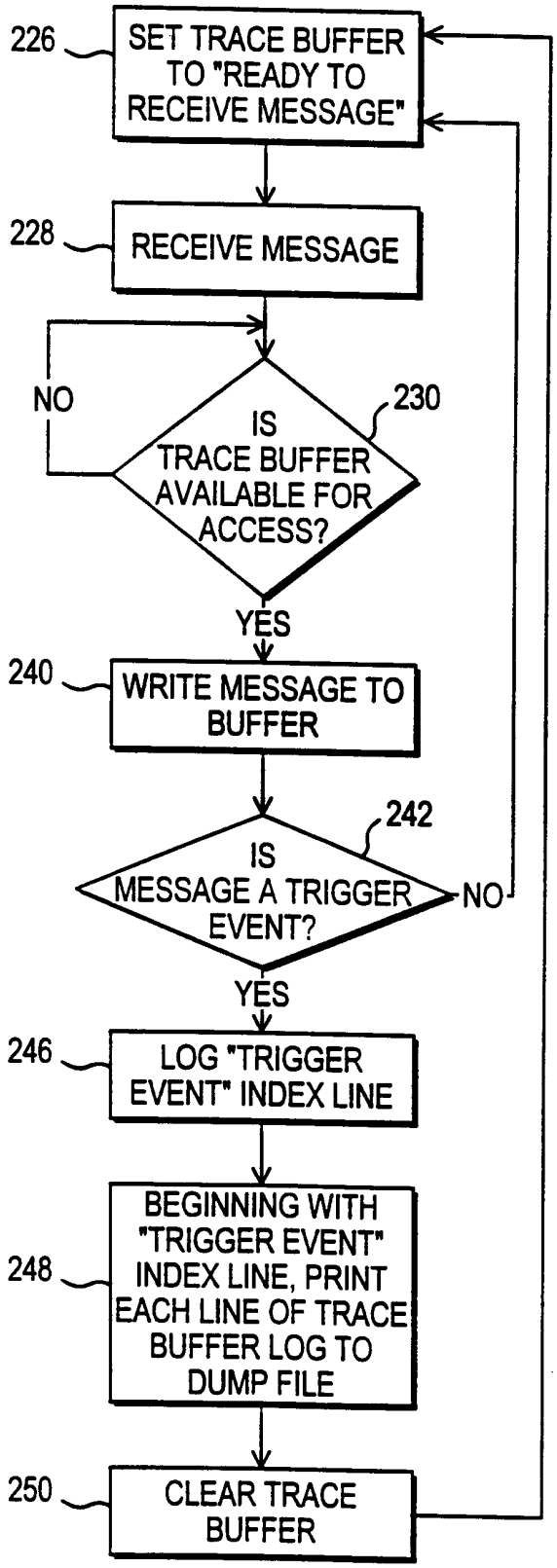

226
SET TRACE BUFFER TO "READY TO RECEIVE MESSAGE"
228
RECEIVE MESSAGE
NO
230
IS TRACE BUFFER AVAILABLE FOR ACCESS?
YES
240
WRITE MESSAGE TO BUFFER
242
IS MESSAGE A TRIGGER EVENT?
NO
YES
246
LOG "TRIGGER EVENT" INDEX LINE
248
BEGINNING WITH "TRIGGER EVENT" INDEX LINE, PRINT EACH LINE OF TRACE BUFFER LOG TO DUMP FILE
250
CLEAR TRACE BUFFER

FIG. 10

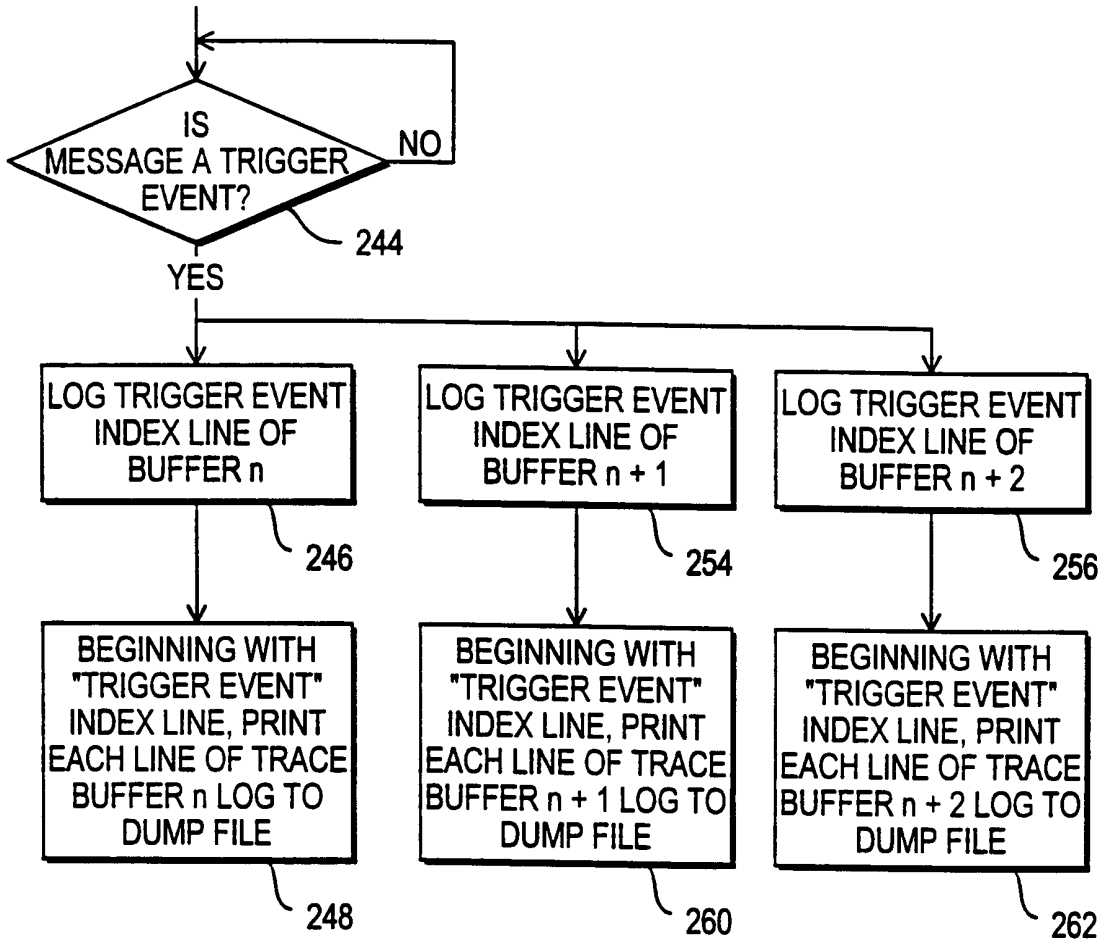

IS MESSAGE A TRIGGER EVENT?
NO
244
YES
LOG TRIGGER EVENT INDEX LINE OF BUFFER n
246
LOG TRIGGER EVENT INDEX LINE OF BUFFER n + 1
254
LOG TRIGGER EVENT INDEX LINE OF BUFFER n + 2
256
BEGINNING WITH "TRIGGER EVENT" INDEX LINE, PRINT EACH LINE OF TRACE BUFFER n LOG TO DUMP FILE
248
BEGINNING WITH "TRIGGER EVENT" INDEX LINE, PRINT EACH LINE OF TRACE BUFFER n + 1 LOG TO DUMP FILE
260
BEGINNING WITH "TRIGGER EVENT" INDEX LINE, PRINT EACH LINE OF TRACE BUFFER n + 2 LOG TO DUMP FILE
262

FIG. 11

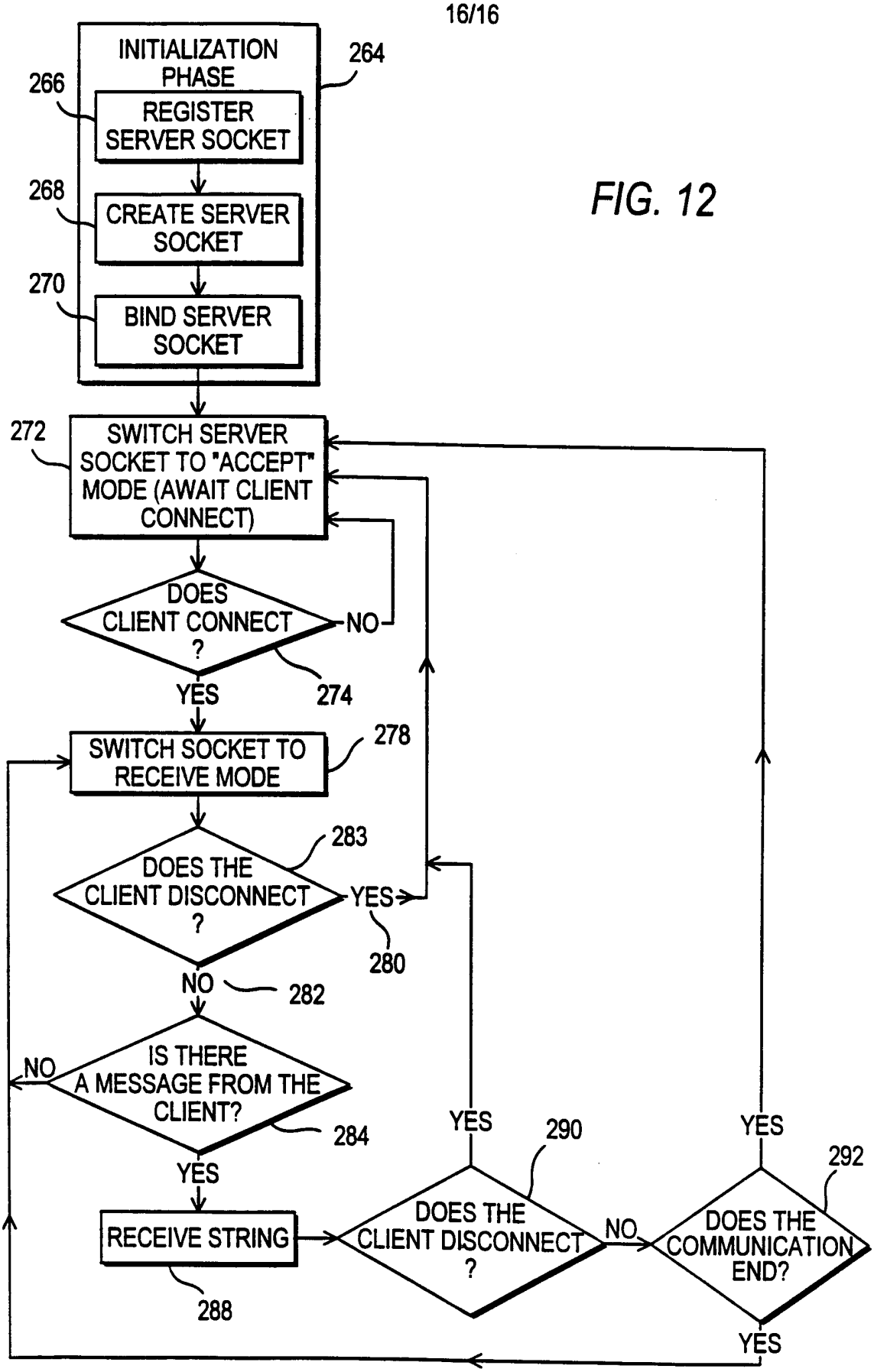

INITIALIZATION PHASE
264
266
REGISTER SERVER SOCKET
268
CREATE SERVER SOCKET
270
BIND SERVER SOCKET
272
SWITCH SERVER SOCKET TO "ACCEPT" MODE (AWAIT CLIENT CONNECT)
DOES CLIENT CONNECT ?
NO
YES
274
SWITCH SOCKET TO RECEIVE MODE
278
DOES THE CLIENT DISCONNECT ?
283
YES
280
NO
282
IS THERE A MESSAGE FROM THE CLIENT?
NO
284
YES
RECEIVE STRING
288
DOES THE CLIENT DISCONNECT ?
290
YES
NO
DOES THE COMMUNICATION END?
292
YES
YES

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/28071

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : H04M 3/50, 3/00, 11/00; G06F 9/46, 15/68, 15/00
US CL : 379/89, 269; 395/615, 709/328; 364/518

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 379/89, 269; 395/615, 709/328; 364/518

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y, P	US 6,094,688 A (MELLEN-GARNETT et al.) 25 July 2000 (25.07.2000), see columns 3-6.	1-12
Y	US 5,913,061 A (GUPTA et al.) 15 June 1999 (15.06.1999), see Summary of the Invention.	1-12
Y, P	US 6,014,440 A (MELKILD et al.) 11 January 2000 (11.01.2000), see columns 3-6	1-12
Y	US 5,163,085 A (SWEET et al.) 10 November 1992 (10.11.1992), see Summary of the Invention.	1-12
Y	US 4,833,625 A (FISHER et al.) 23 May 1989 (23.05.1989), see columns 5-10.	1-12
Y	US 5,625,678 A (BLOMFIELD-BROWN) 29 April 1997 (29.04.1997), see columns 4-7.	1-12
Y	US 5.615,112 A (LIU SHENG et al.) 25 March 1997 (25.03.1997), see Summary of the Invention.	1-12

☐ 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 relevance

"E" earlier application or patent 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 another 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 relevance; 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 relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
26 December 2000 (26.12.2000)

Date of mailing of the international search report
31 JAN 2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. 703 305 3230

Authorized officer
ALVIN OBERLEY James R. Matthews
Telephone No. 703 305 3665

Form PCT/ISA/210 (second sheet) (July 1998)