EXHIBIT F

US006308171B1

(12) United States Patent
De La Huerga

(10) Patent No.: US 6,308,171 B1
(45) Date of Patent: *Oct. 23, 2001

(54) METHOD AND SYSTEM FOR AUTOMATED DATA STORAGE AND RETRIEVAL

(76) Inventor: Carlos De La Huerga, 9190 N. Upper River Rd., Milwaukee, WI (US) 53217

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: 09/512,125

(22) Filed: Feb. 24, 2000

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/247,349, filed on Feb. 10, 1999, which is a continuation-in-part of application No. 08/727,293, filed on Oct. 9, 1996, now Pat. No. 5,895,461, application No. 09/247,349, which is a continuation-in-part of application No. 08/871,818, filed on Jun. 9, 1997, now Pat. No. 5,903,889.

(60) Provisional application No. 60/023,126, filed on Jul. 30, 1996.

(51) Int. Cl.[7] G06F 17/30; G06F 17/60

(52) U.S. Cl. 707/3; 707/1; 705/2; 705/3

(58) Field of Search 707/1, 3; 705/2, 705/3

(56) References Cited

U.S. PATENT DOCUMENTS

5,291,399	*	3/1994	Chaco	705/3
5,558,638	*	9/1996	Evers et al.	705/3
5,867,821	*	2/1999	Ballantyne et al.	705/2
5,895,461	*	4/1999	De La Huerga et al.	707/1
5,903,889	*	5/1999	De La Huerga et al.	707/3
5,942,986	*	8/1999	Shabot et al.	340/825.44
6,157,914	*	12/2000	Seto et al.	705/3

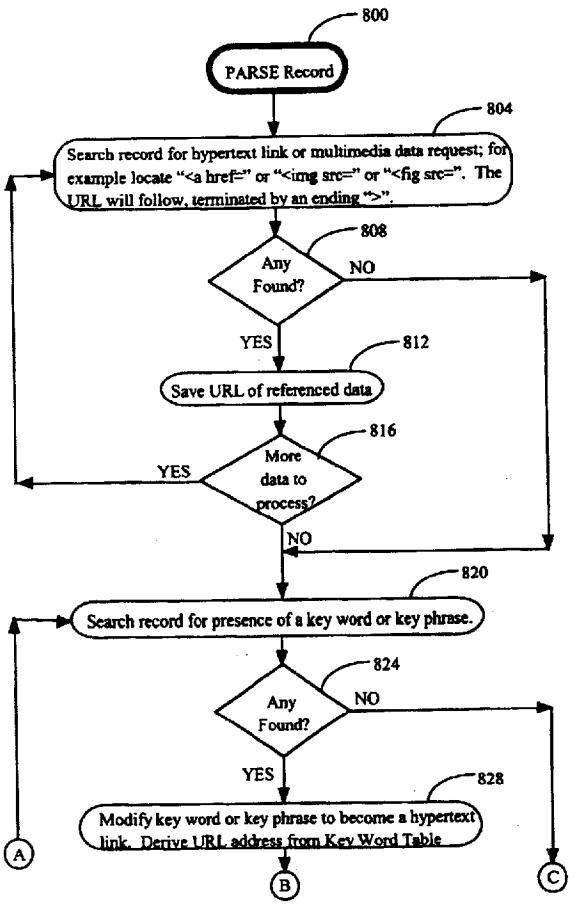
* cited by examiner

Primary Examiner—Thomas Black
Assistant Examiner—Frantz Coby

(57) ABSTRACT

An apparatus and method for linking a keyword phrase in a first record to a second record referenced by the keyword phrase including receiving the keyword phrase, identifying the second record and rendering the second record accessible and an apparatus and method for identifying a reference in a first record to a second record and indicating existence of the second record in the first record.

64 Claims, 30 Drawing Sheets



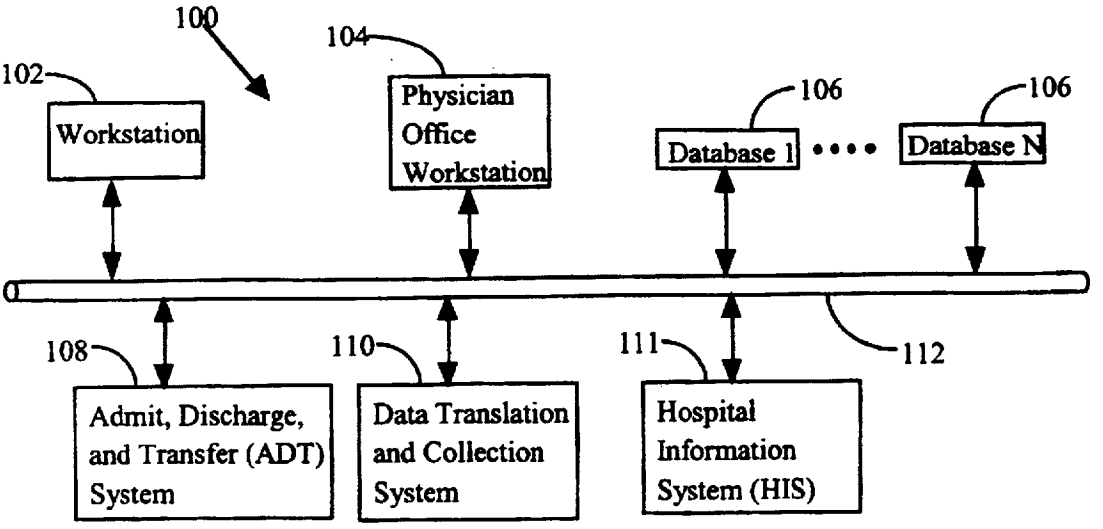


Figure 1

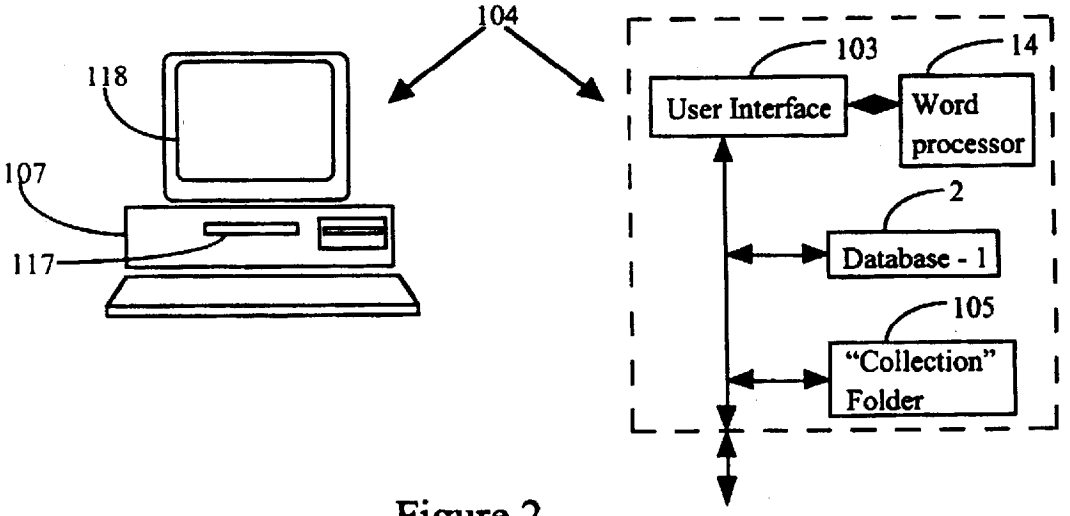


Figure 2

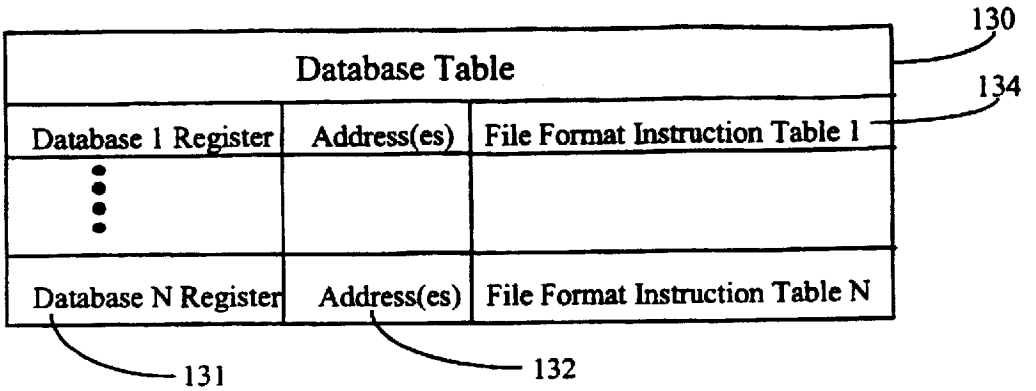


Figure 3A

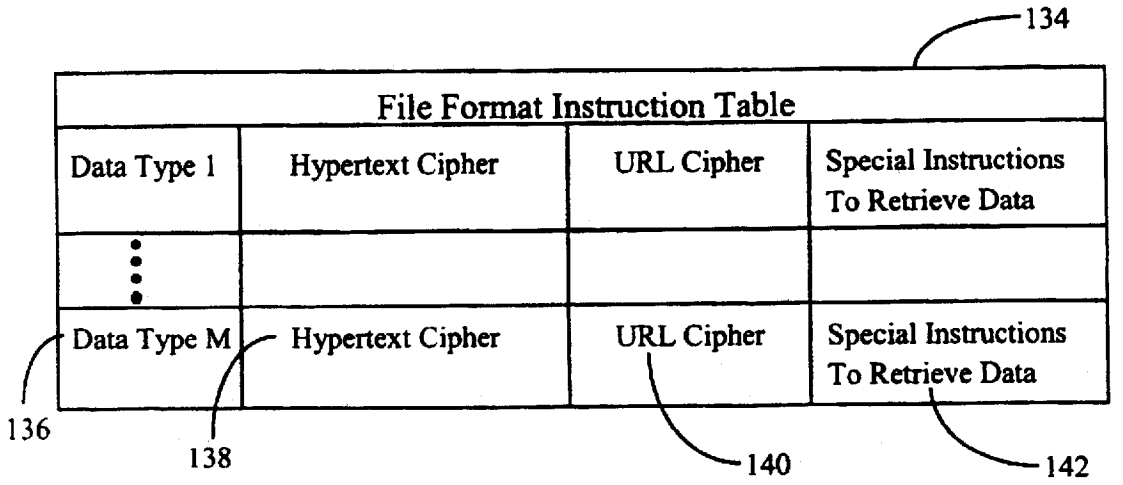


Figure 3B

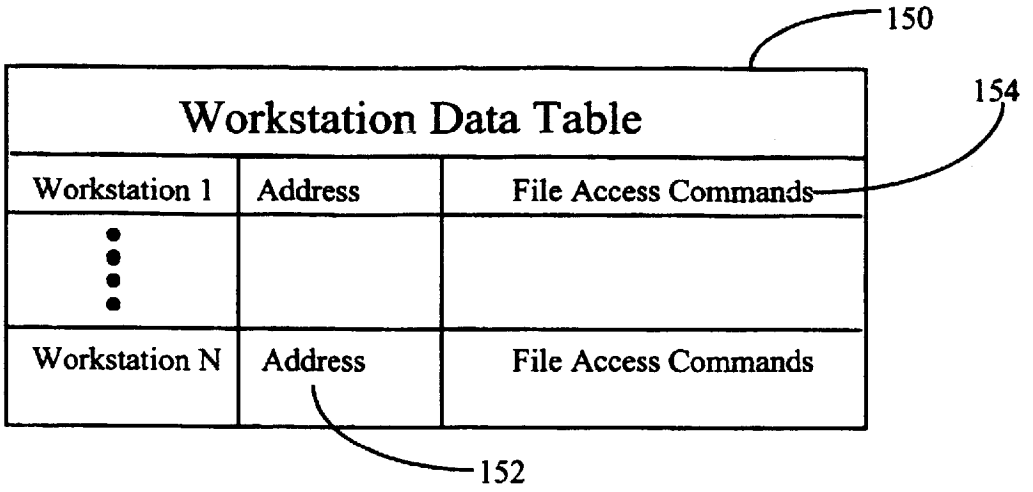


Figure 4A

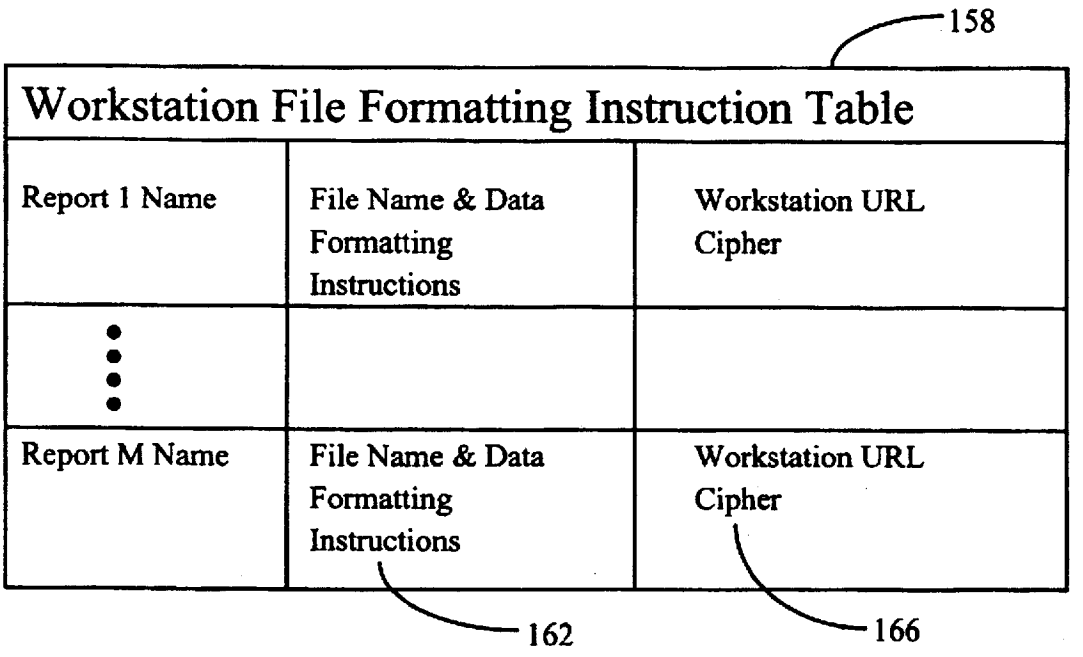


Figure 4B

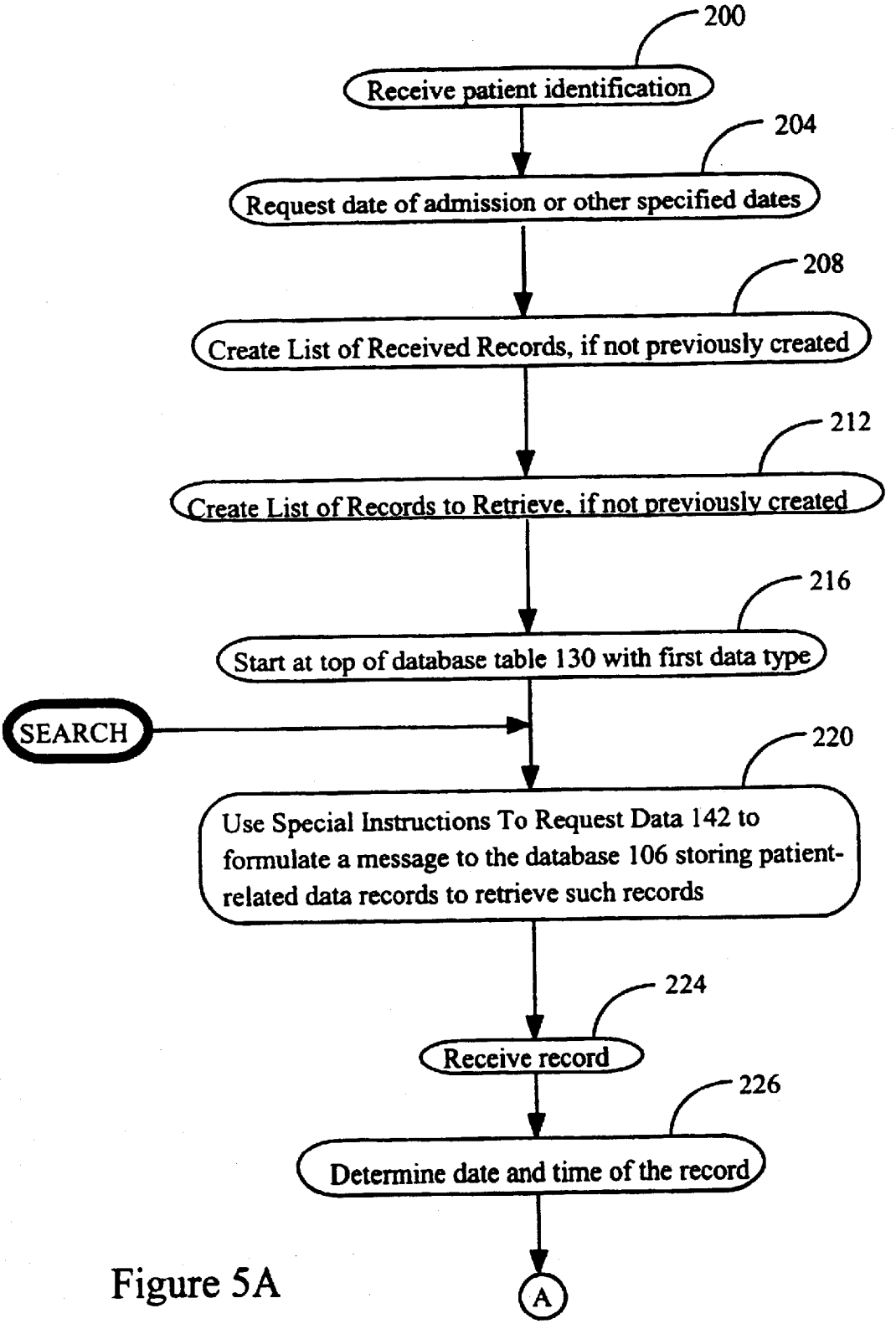


Figure 5A

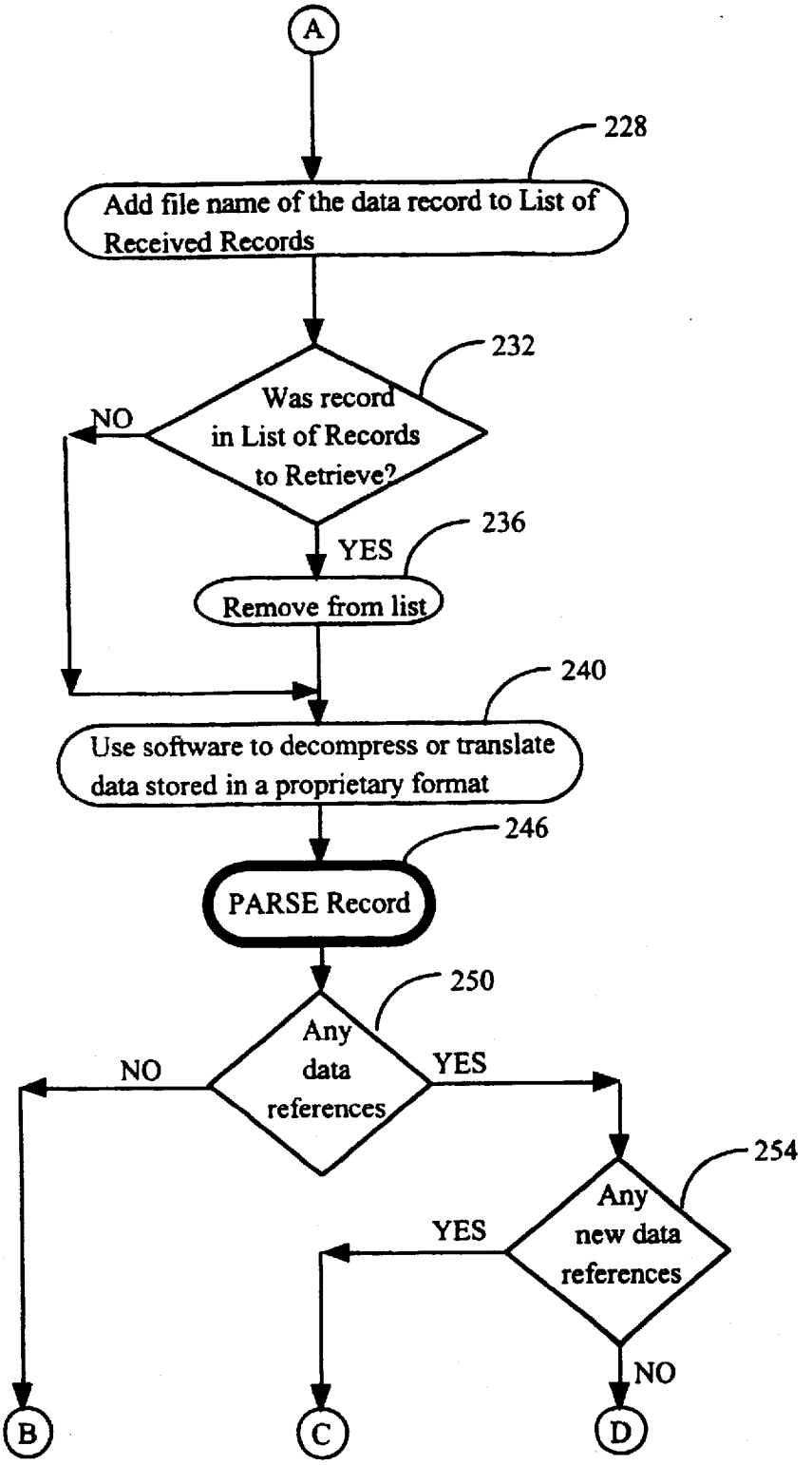


Figure 5B

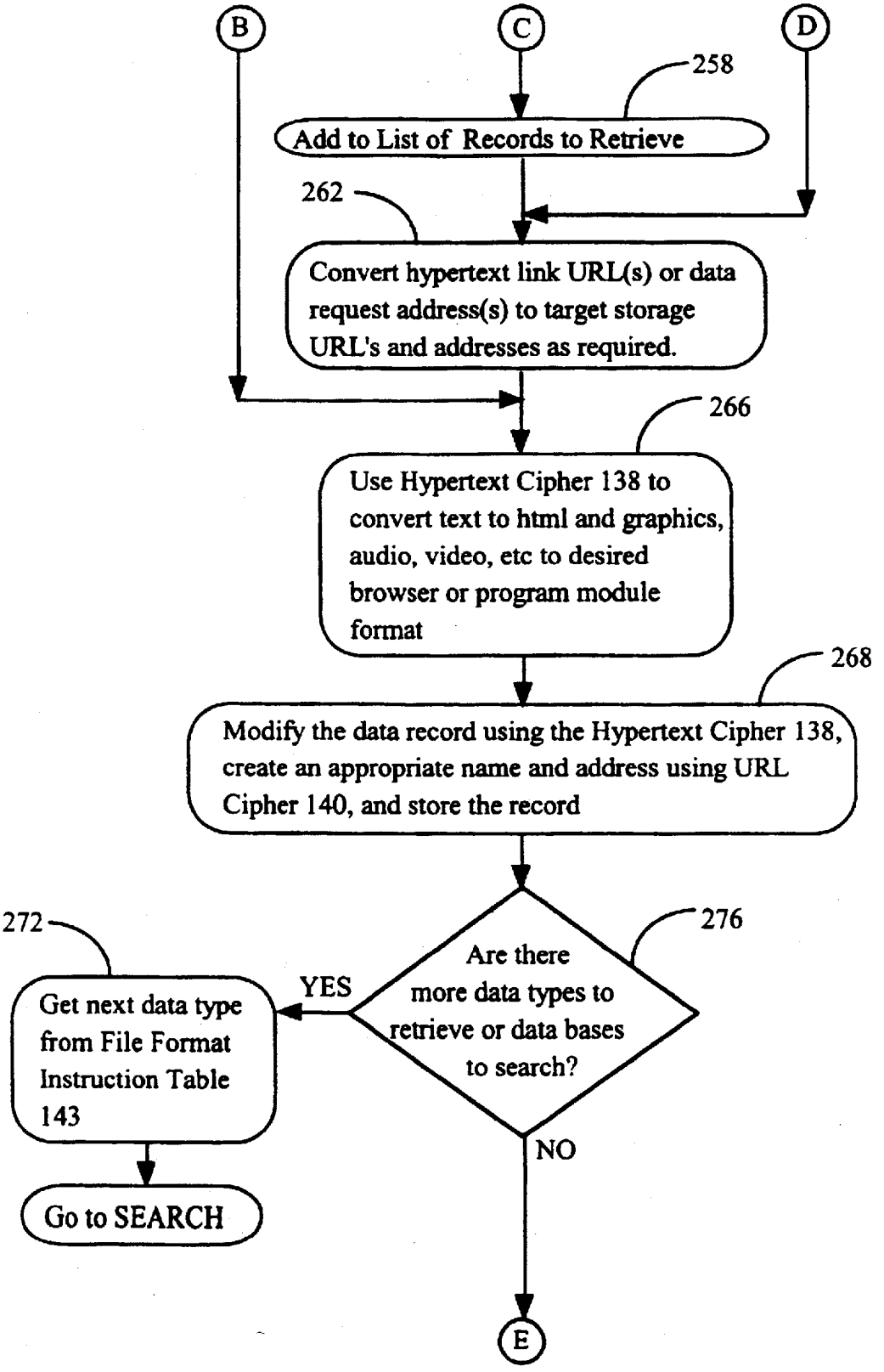


Figure 5C

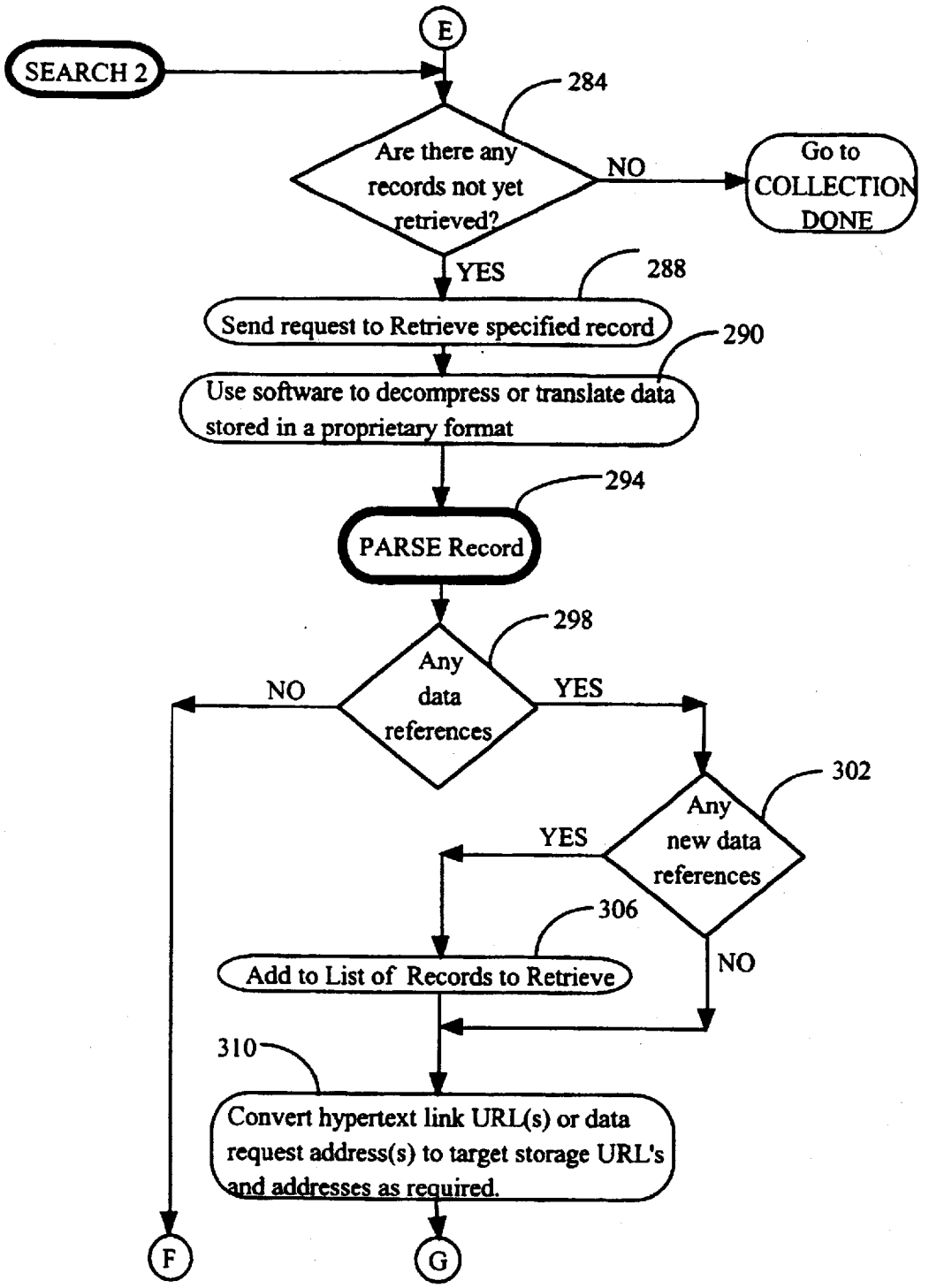


Figure 5D

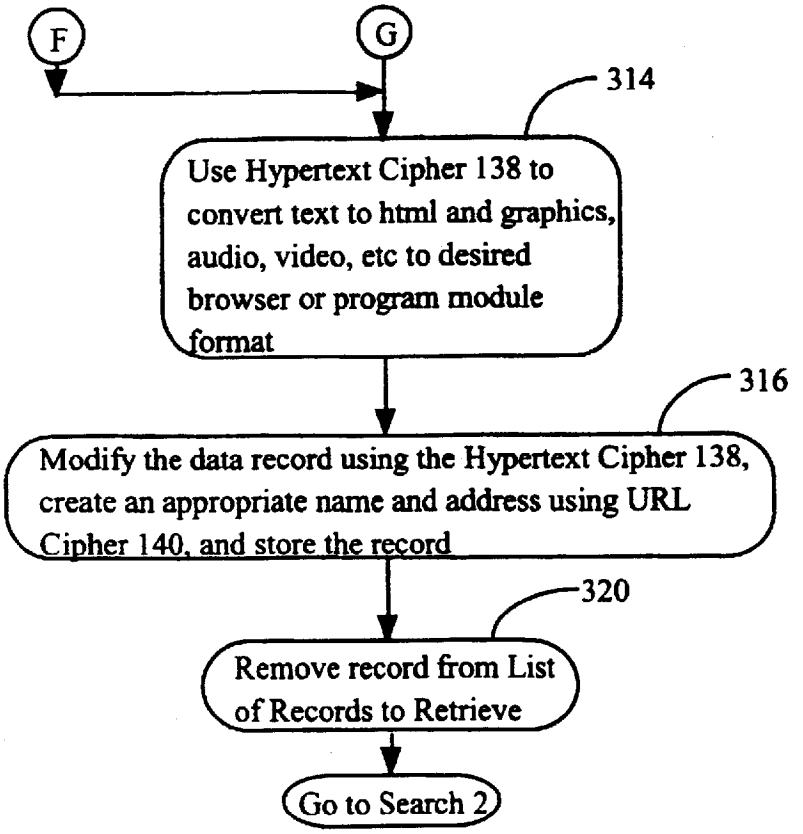


Figure 5E

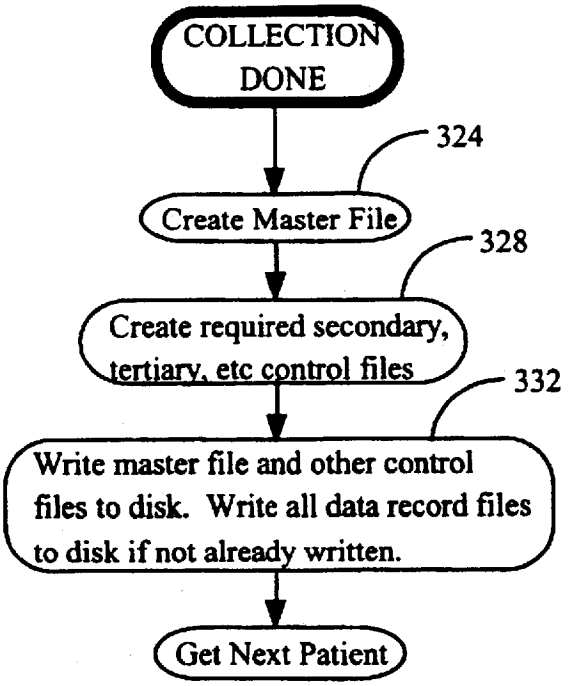


Figure 5F

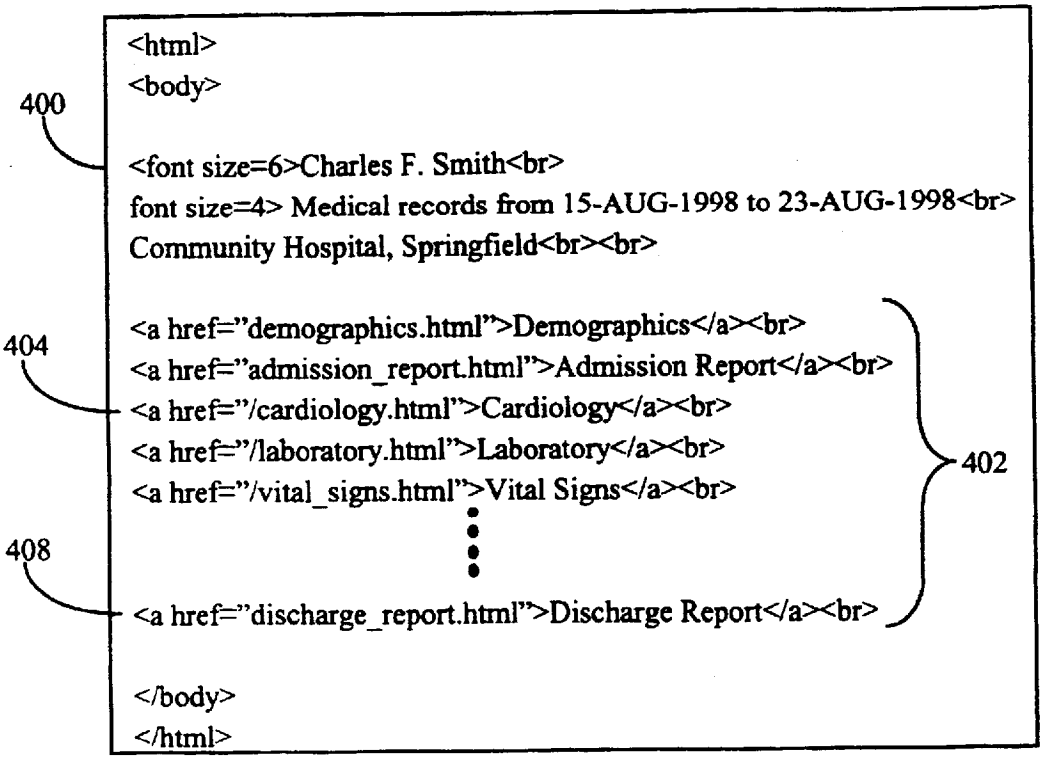


Figure 6A

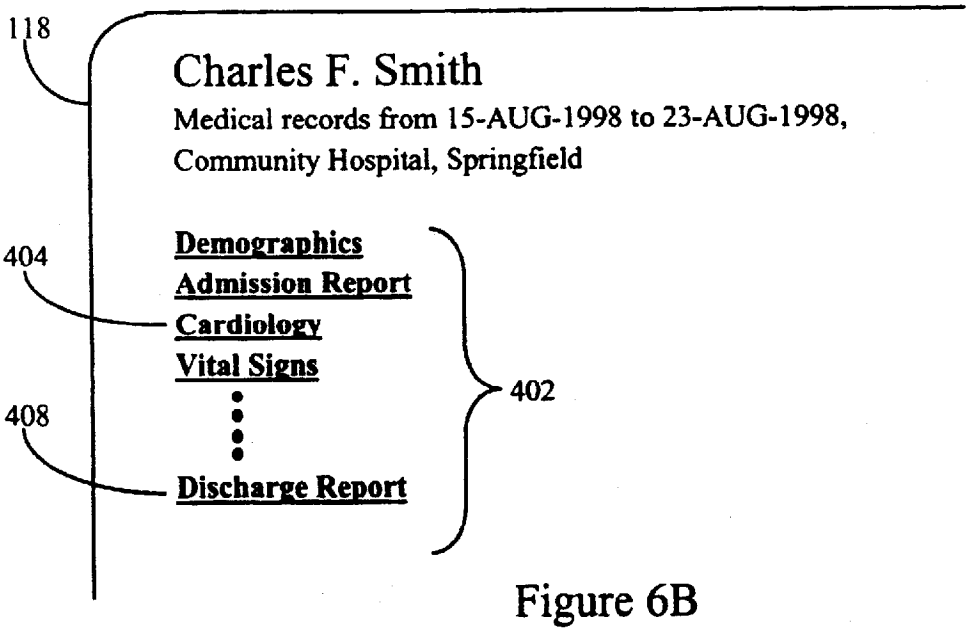


Figure 6B

412

```
<html>
<body>
DISCHARGE SUMMARY for Charles F. Smith<br><br>

Date of Admission: 15-AUG-98<br>
Date of Discharge: 17-AUG-98<br><br>

HISTORY OF PRESENT ILLNESS:<br>
Patient #1 is a 47-year-old male admitted for further evaluation of recent exertional
angina and abnormal exercise test.<br><br>

416
HOSPITAL COURSE:<br>
The patient was admitted to the Telemetry unit and underwent <a
href="charles_f_smith/cardiology/cath/1998-08-15/10:25/report.html">cardiac
catheterization on August 15, 1998</a>. Catheterization demonstrated normal
ventricular function without evidence for prior infarction. The coronary arteriogram
showed moderate stenosis throughout the mid and distal portions of the left anterior
descending artery and diagonal branch, as well as ••••
```

Figure 7A

118

416

DISCHARGE SUMMARY for Charles F. Smith

Date of Admission: 15-AUG-98
Date of Discharge: 17-AUG-98

HISTORY OF PRESENT ILLNESS:
Patient #1 is a 47-year-old male admitted for further evaluation of recent exertional angina and abnormal exercise test.

HOSPITAL COURSE:
The patient was admitted to the Telemetry unit and underwent **catheterization on August 15, 1998**. Catheterization demonstrated normal ventricular function without evidence for prior infarction. The coronary arteriogram showed moderate stenosis throughout the mid and distal portions of the left anterior descending artery and diagonal branch, as well as ••••

Figure 7B

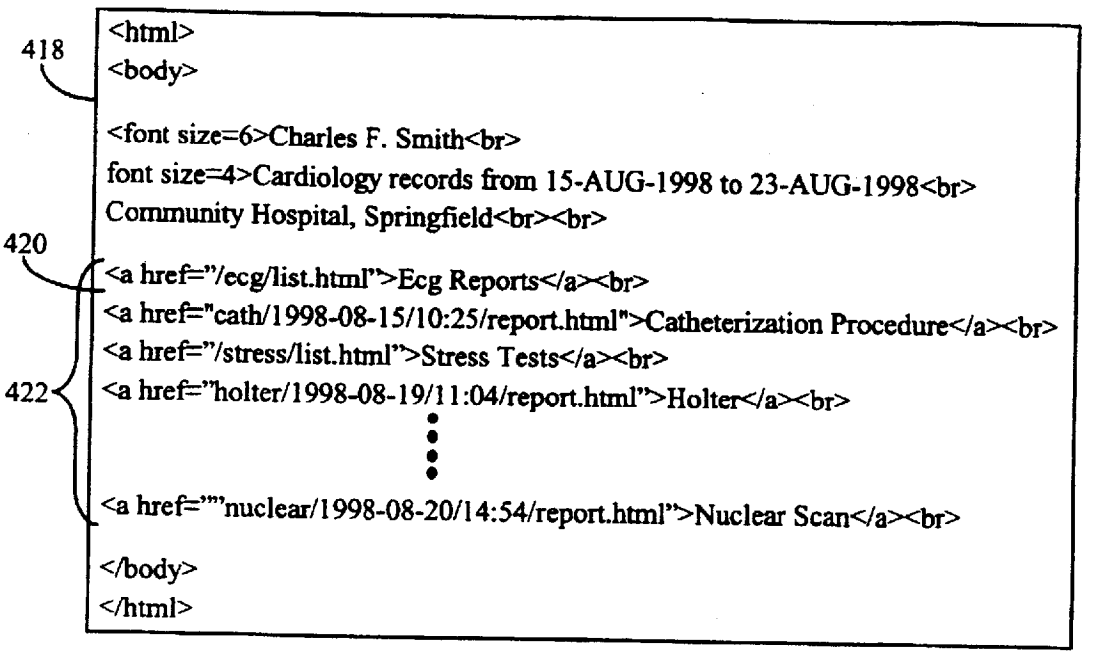


Figure 8A

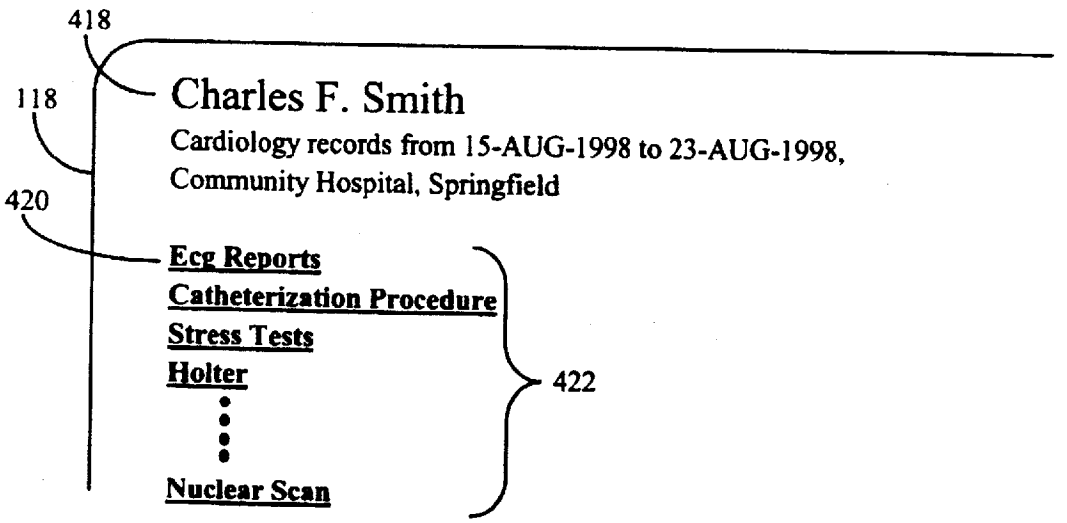


Figure 8B

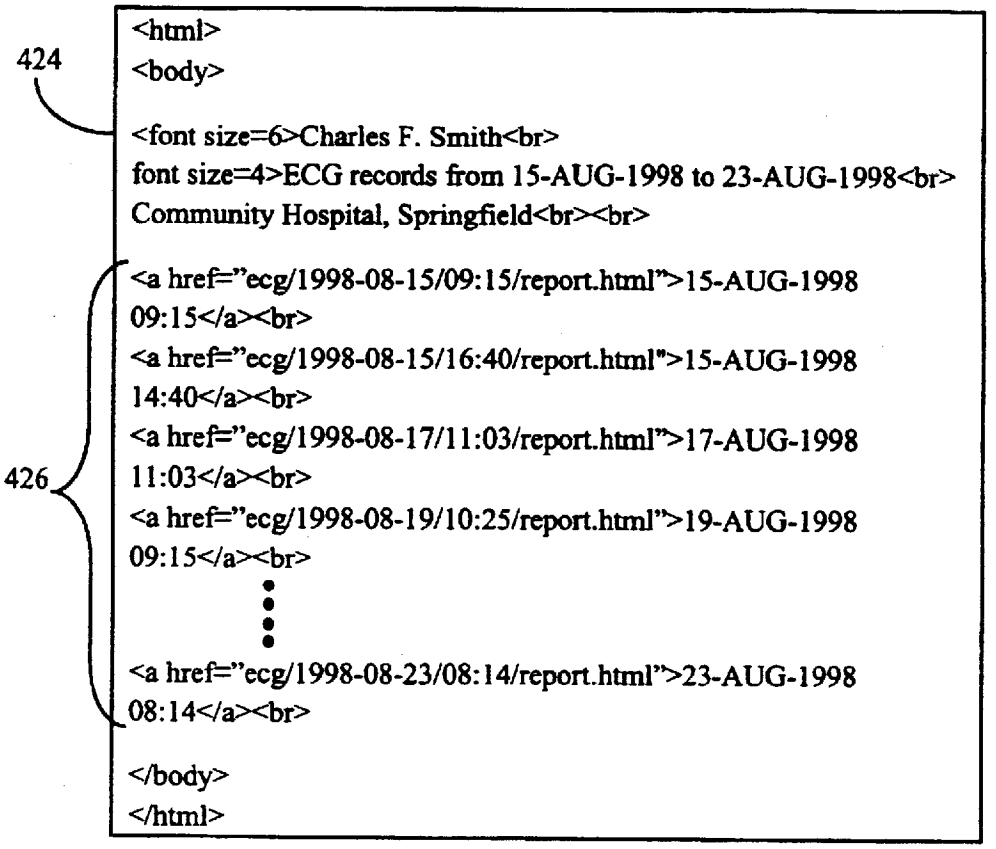


Figure 9A

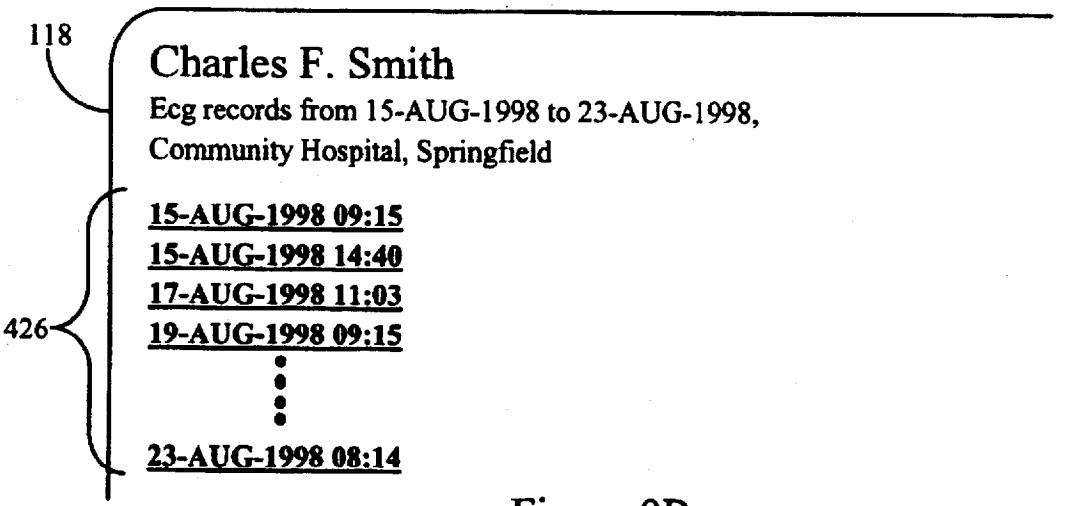


Figure 9B

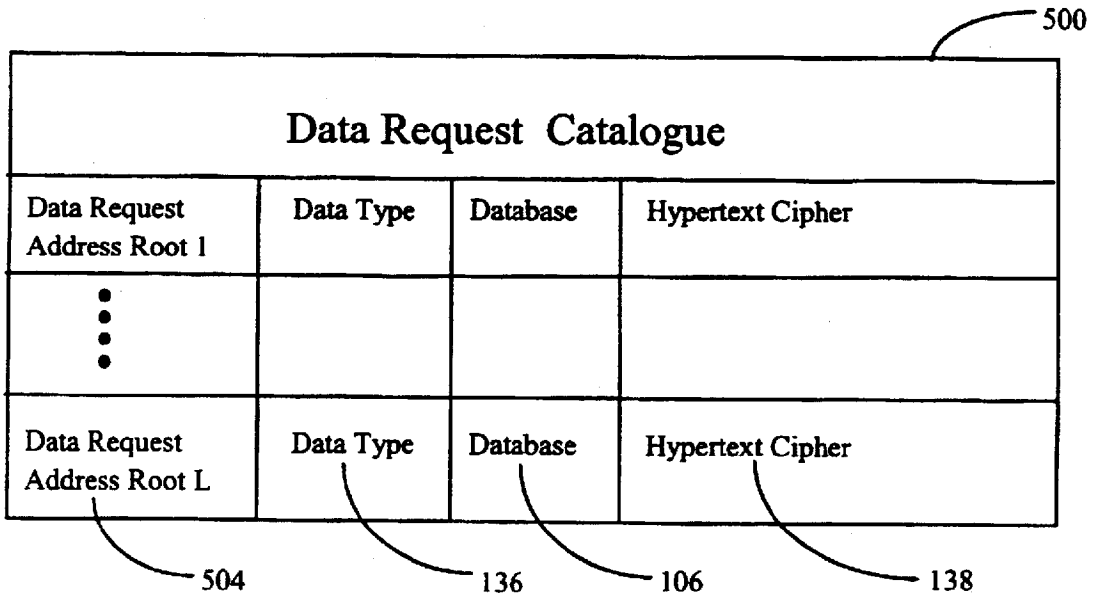


Figure 10

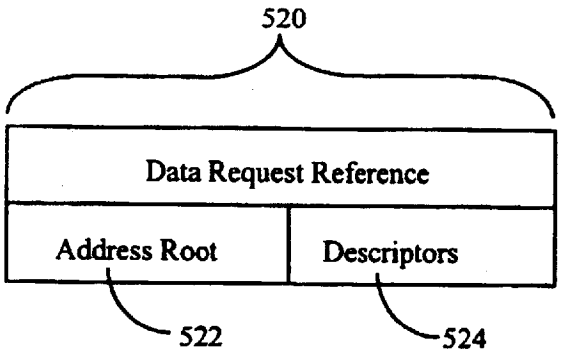


Figure 11

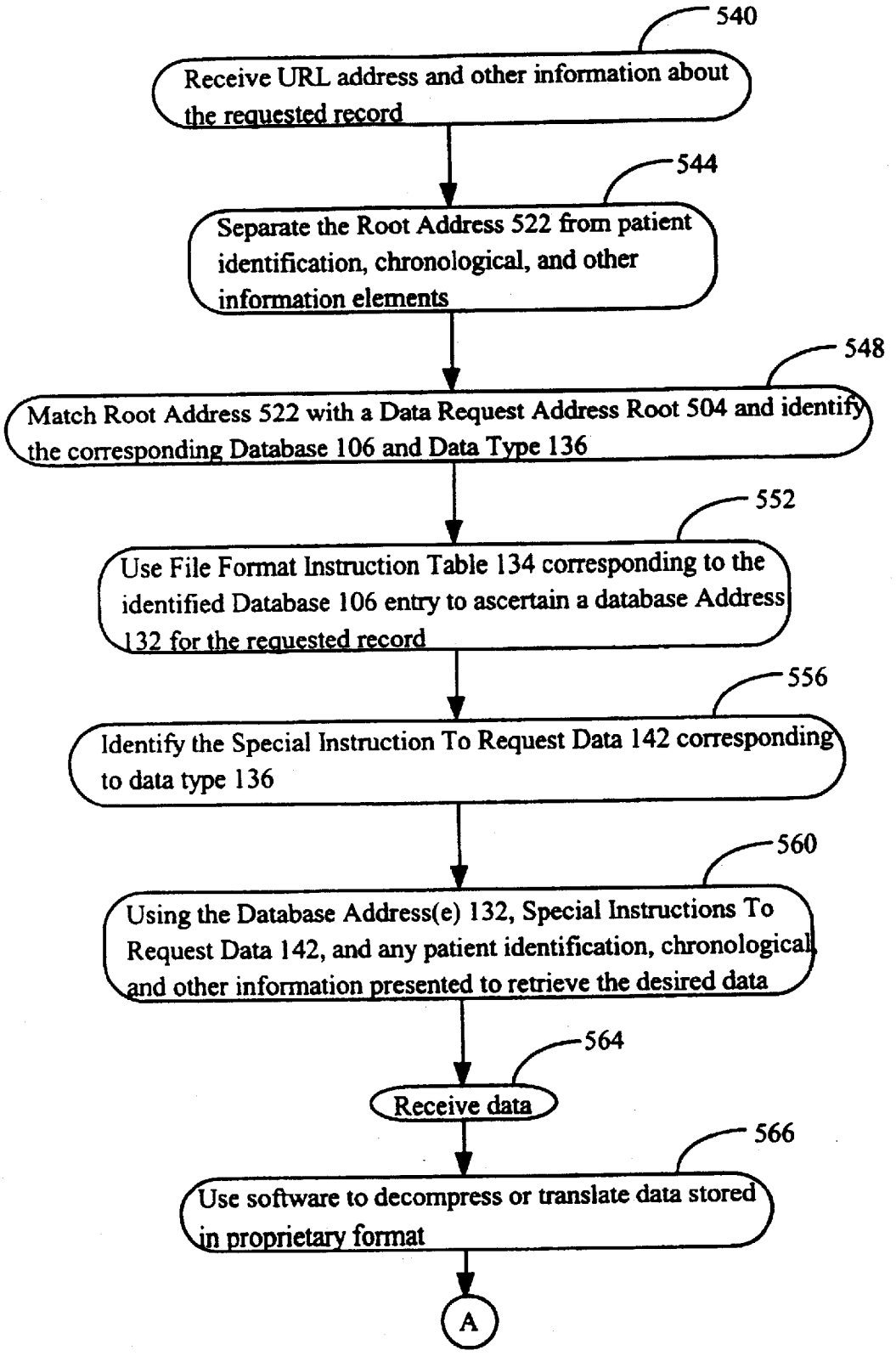

540
Receive URL address and other information about the requested record
544
Separate the Root Address 522 from patient identification, chronological, and other information elements
548
Match Root Address 522 with a Data Request Address Root 504 and identify the corresponding Database 106 and Data Type 136
552
Use File Format Instruction Table 134 corresponding to the identified Database 106 entry to ascertain a database Address 132 for the requested record
556
Identify the Special Instruction To Request Data 142 corresponding to data type 136
560
Using the Database Address(e) 132, Special Instructions To Request Data 142, and any patient identification, chronological and other information presented to retrieve the desired data
564
Receive data
566
Use software to decompress or translate data stored in proprietary format
A

Figure 12A

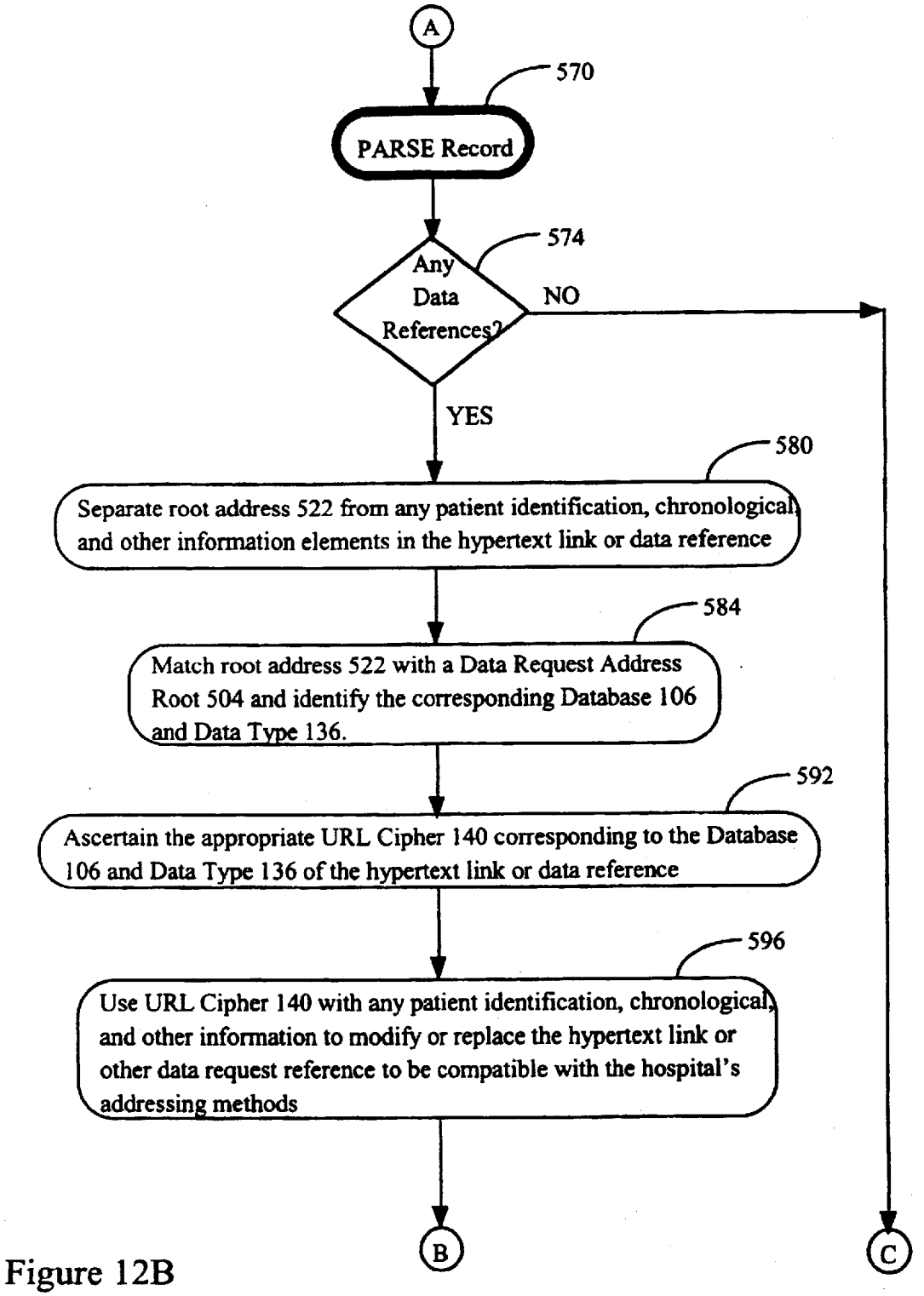


Figure 12B

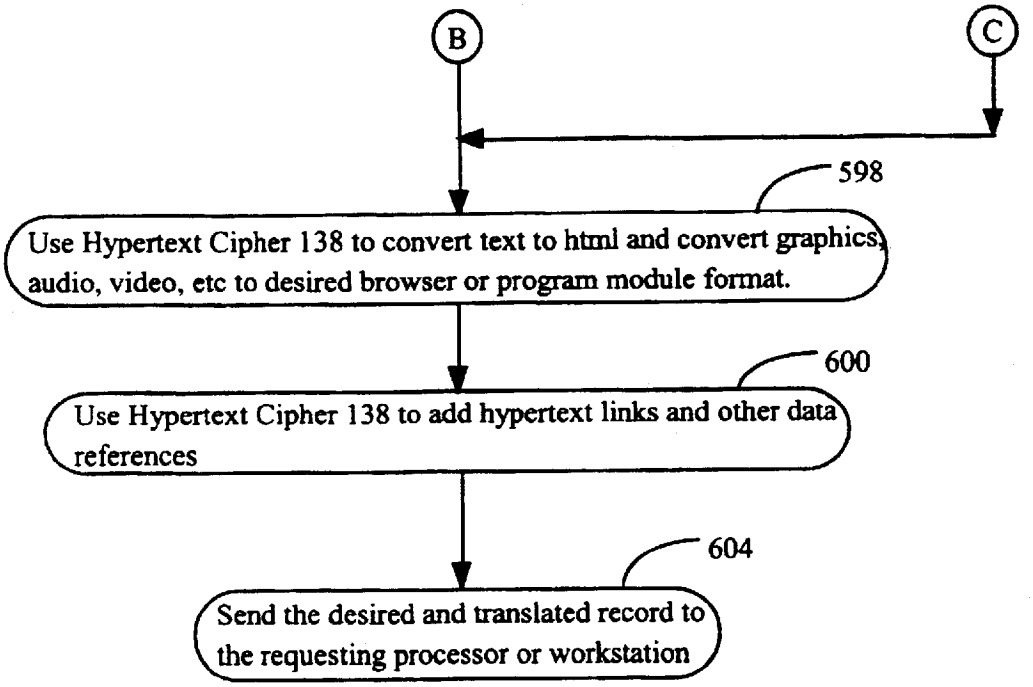

B
C
598
Use Hypertext Cipher 138 to convert text to html and convert graphics, audio, video, etc to desired browser or program module format.
600
Use Hypertext Cipher 138 to add hypertext links and other data references
604
Send the desired and translated record to the requesting processor or workstation

Figure 12C

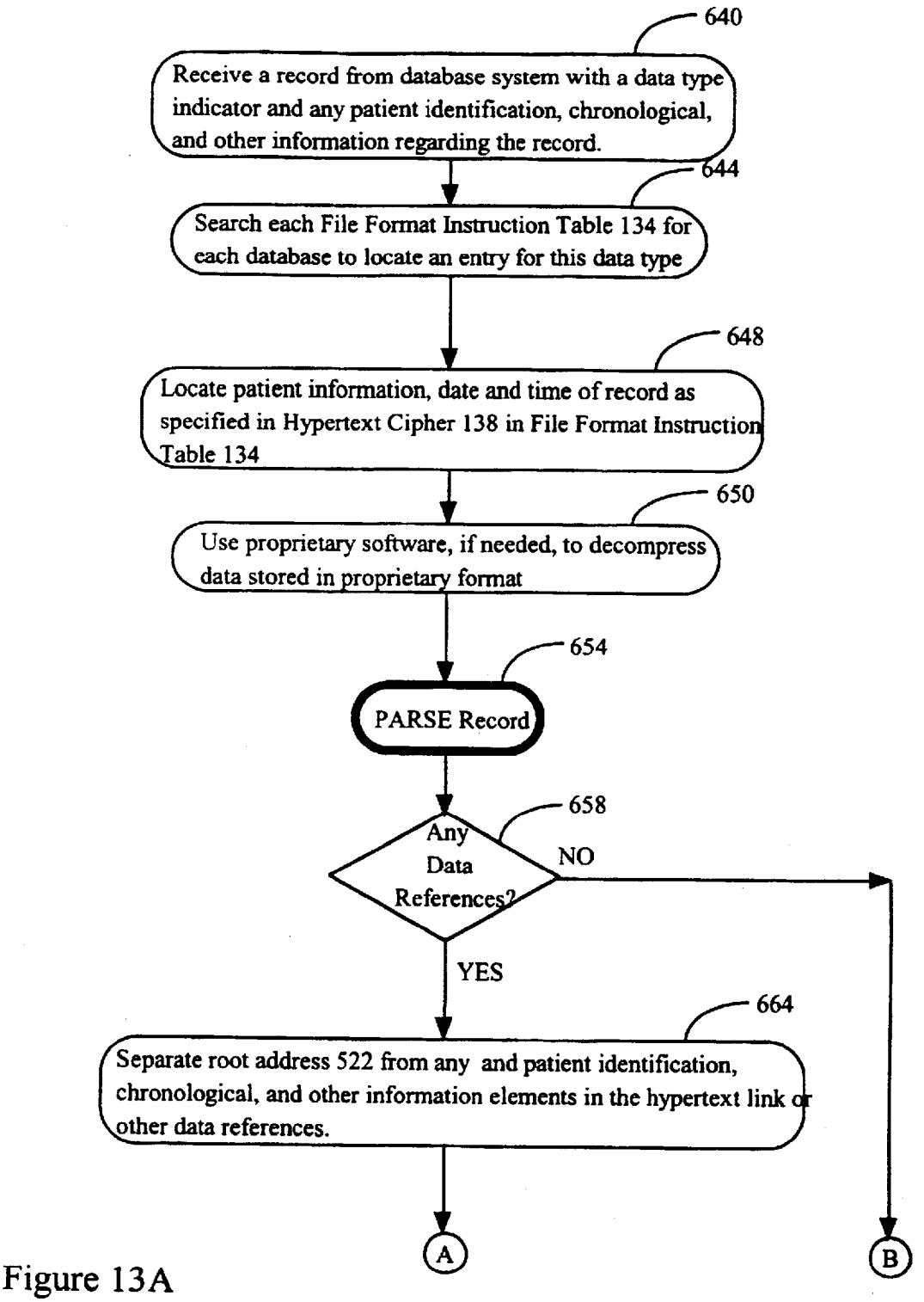


Figure 13A

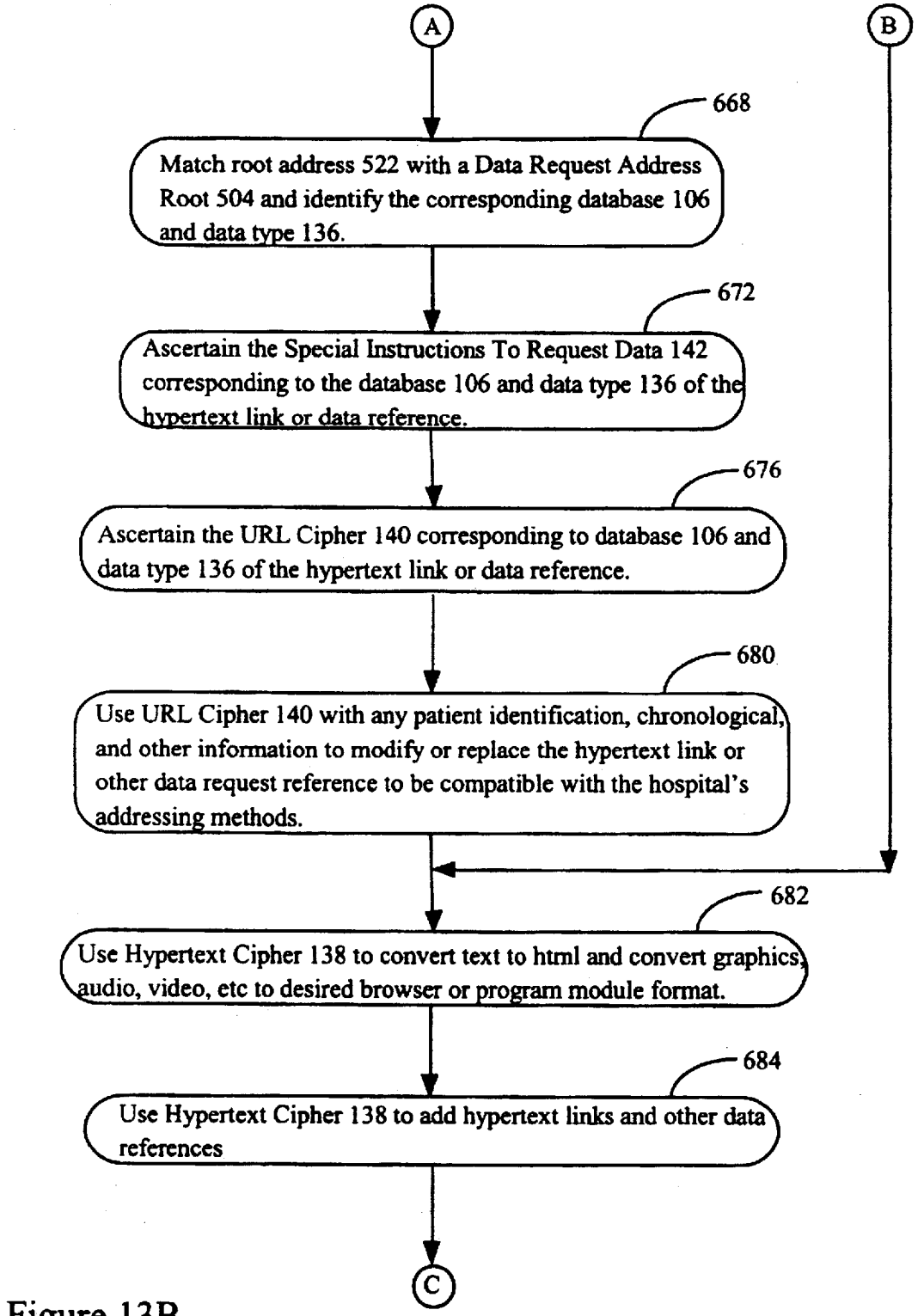

A
B
668
Match root address 522 with a Data Request Address Root 504 and identify the corresponding database 106 and data type 136.
672
Ascertain the Special Instructions To Request Data 142 corresponding to the database 106 and data type 136 of the hypertext link or data reference.
676
Ascertain the URL Cipher 140 corresponding to database 106 and data type 136 of the hypertext link or data reference.
680
Use URL Cipher 140 with any patient identification, chronological, and other information to modify or replace the hypertext link or other data request reference to be compatible with the hospital's addressing methods.
682
Use Hypertext Cipher 138 to convert text to html and convert graphics, audio, video, etc to desired browser or program module format.
684
Use Hypertext Cipher 138 to add hypertext links and other data references
C

Figure 13B

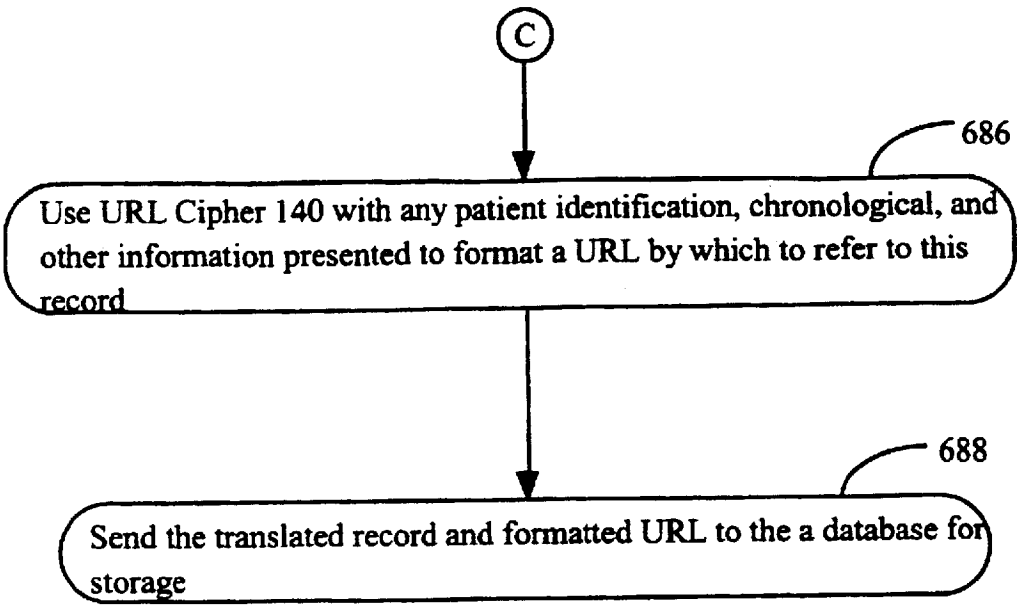

C
686
Use URL Cipher 140 with any patient identification, chronological, and other information presented to format a URL by which to refer to this record
688
Send the translated record and formatted URL to the a database for storage

Figure 13C

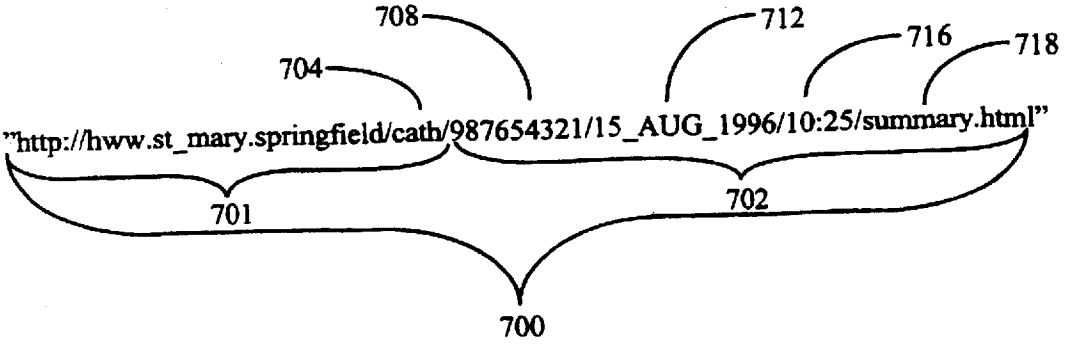


Figure 14A

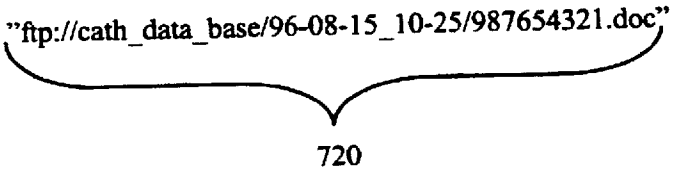


Figure 14B

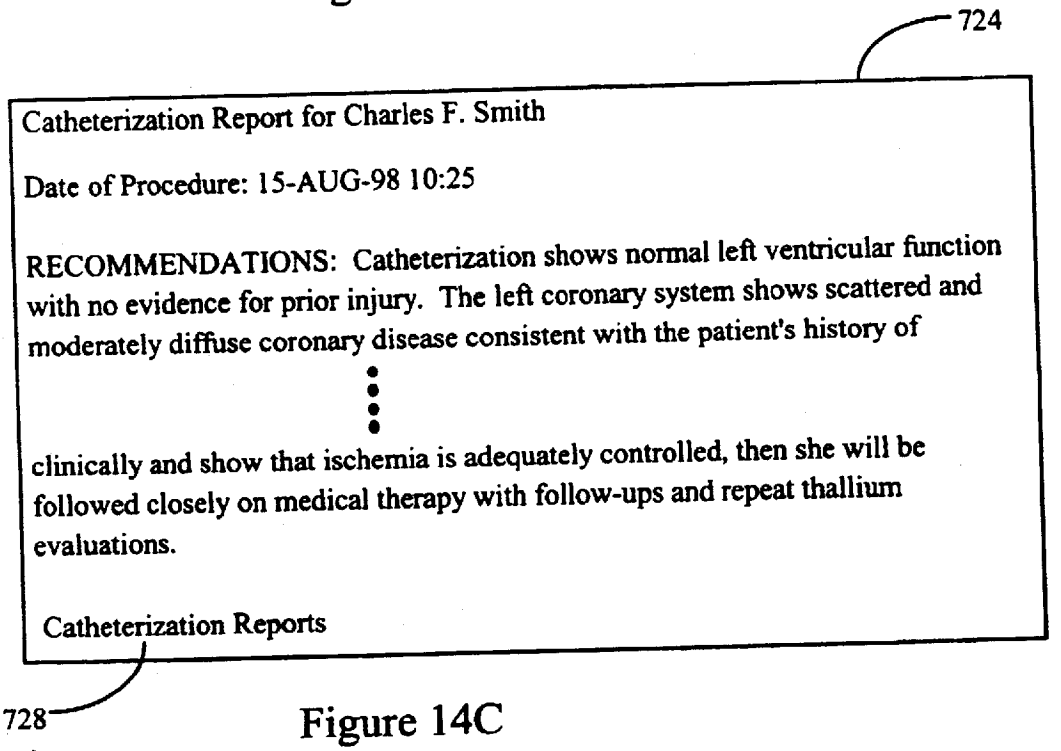


Figure 14C

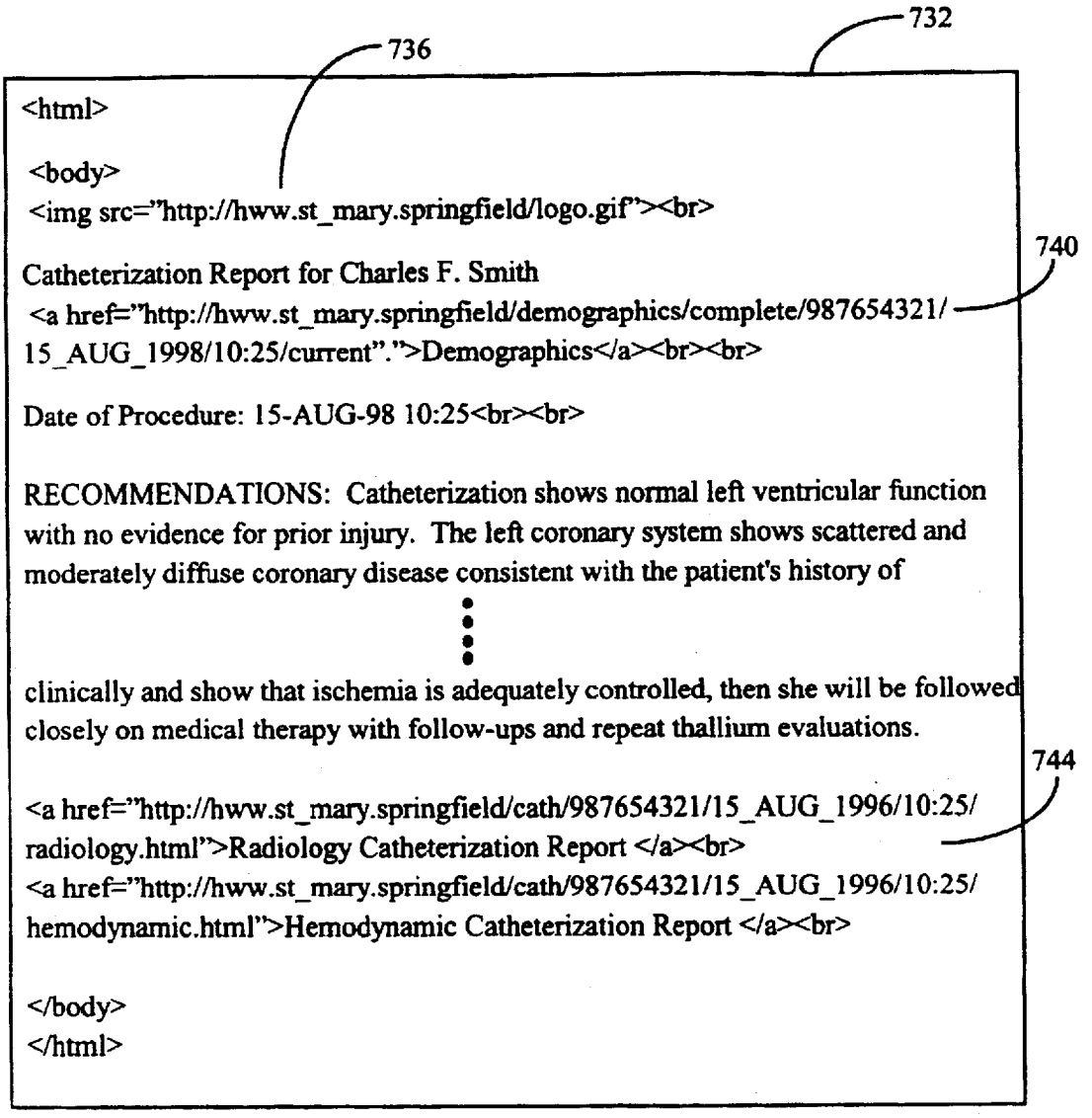


Figure 14D

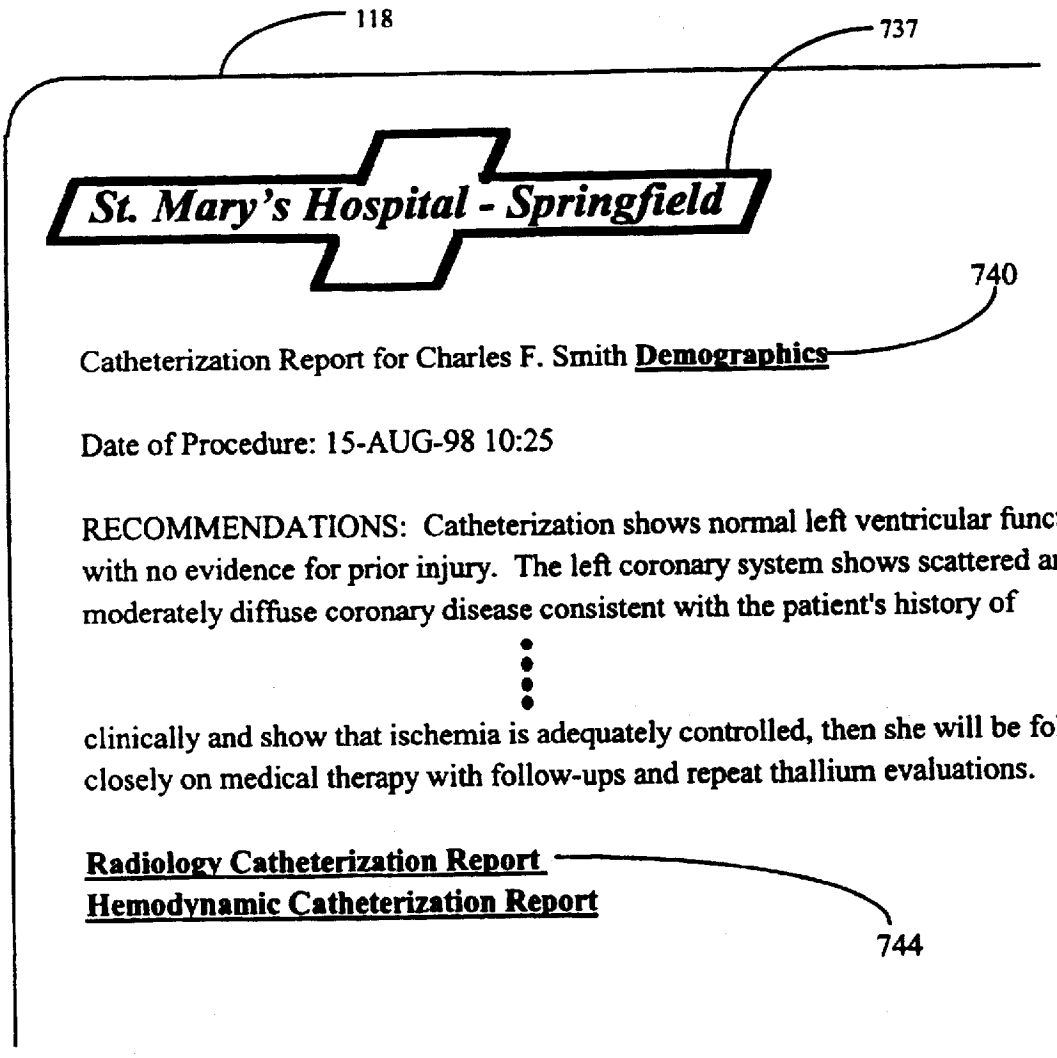


Figure 14E

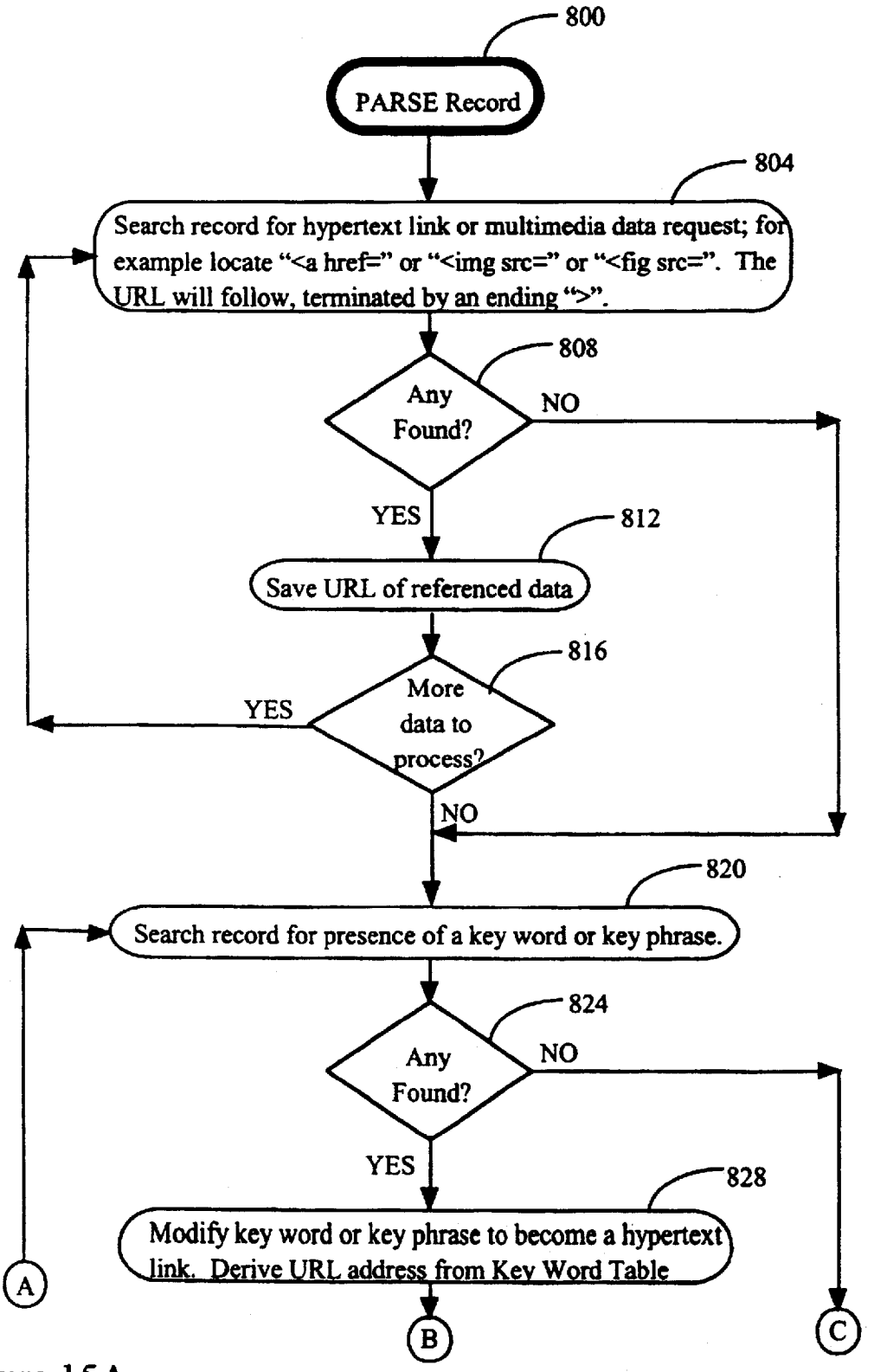

800
PARSE Record
804
Search record for hypertext link or multimedia data request; for example locate "<a href=" or "<img src=" or "<fig src=". The URL will follow, terminated by an ending ">".
808
Any Found?
NO
YES
812
Save URL of referenced data
816
More data to process?
YES
NO
820
Search record for presence of a key word or key phrase.
824
Any Found?
NO
YES
828
Modify key word or key phrase to become a hypertext link. Derive URL address from Key Word Table
A
B
C

Figure 15A

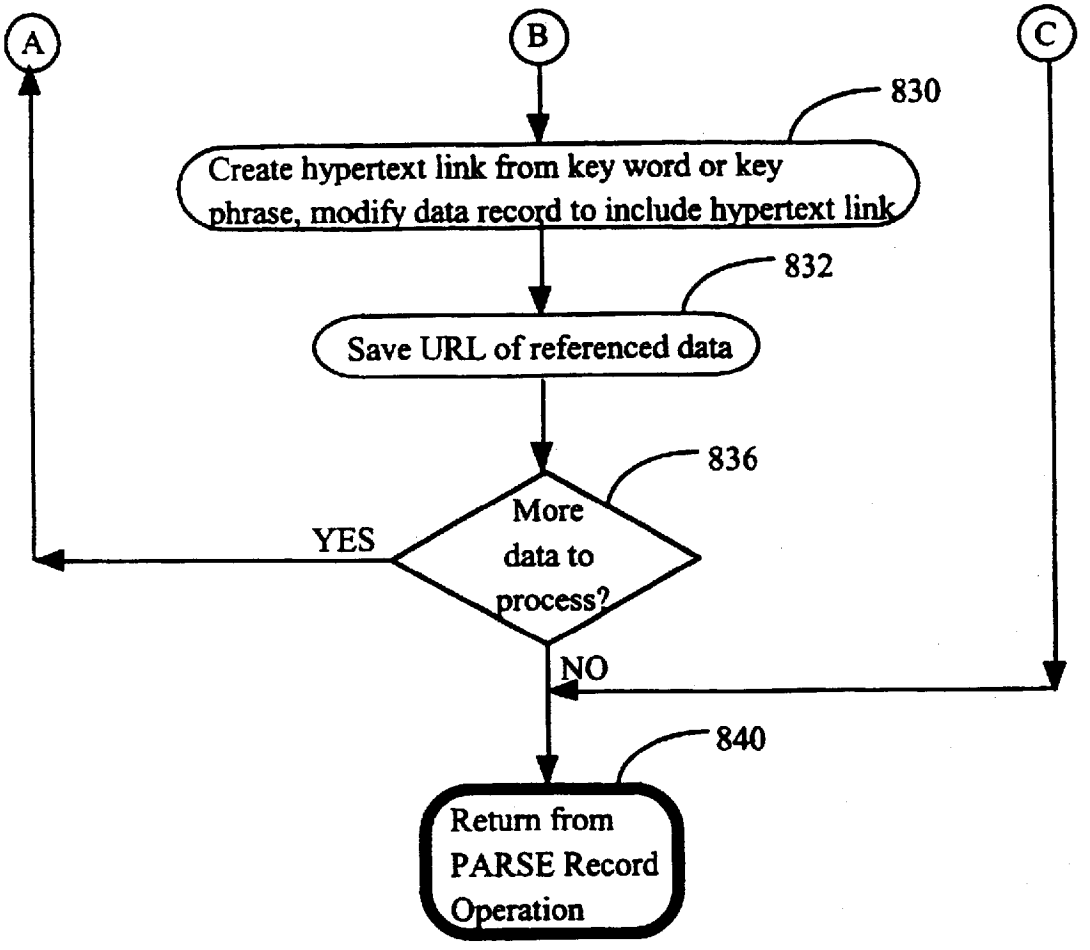

A
B
C
830
Create hypertext link from key word or key phrase, modify data record to include hypertext link
832
Save URL of referenced data
836
More data to process?
YES
NO
840
Return from PARSE Record Operation

Figure 15B

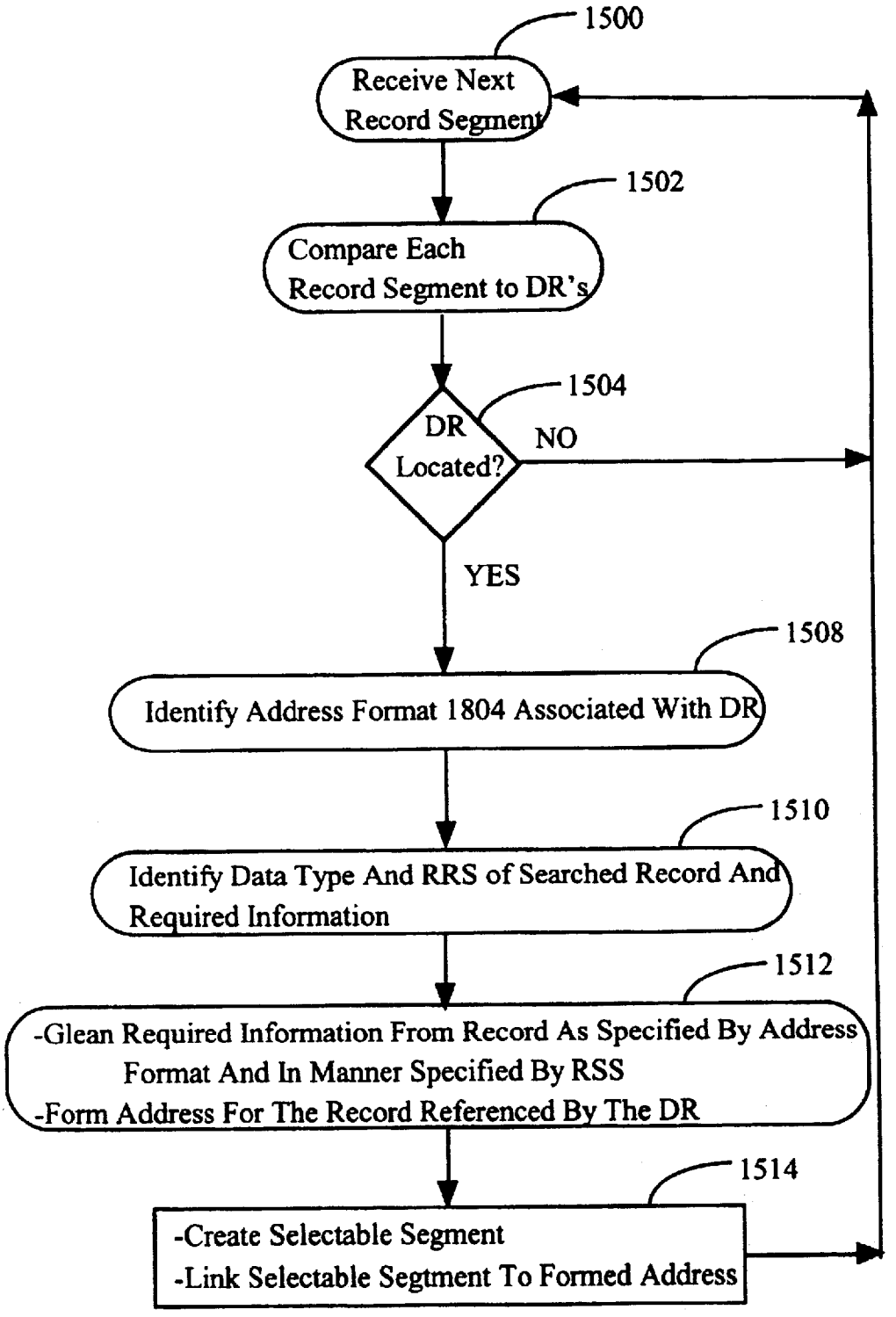

1500
Receive Next Record Segment
1502
Compare Each Record Segment to DR's
1504
DR Located?
NO
YES
1508
Identify Address Format 1804 Associated With DR
1510
Identify Data Type And RRS of Searched Record And Required Information
1512
-Glean Required Information From Record As Specified By Address Format And In Manner Specified By RSS
-Form Address For The Record Referenced By The DR
1514
-Create Selectable Segment
-Link Selectable Segtment To Formed Address

Figure 16

Text As It Appears in the Word Processor

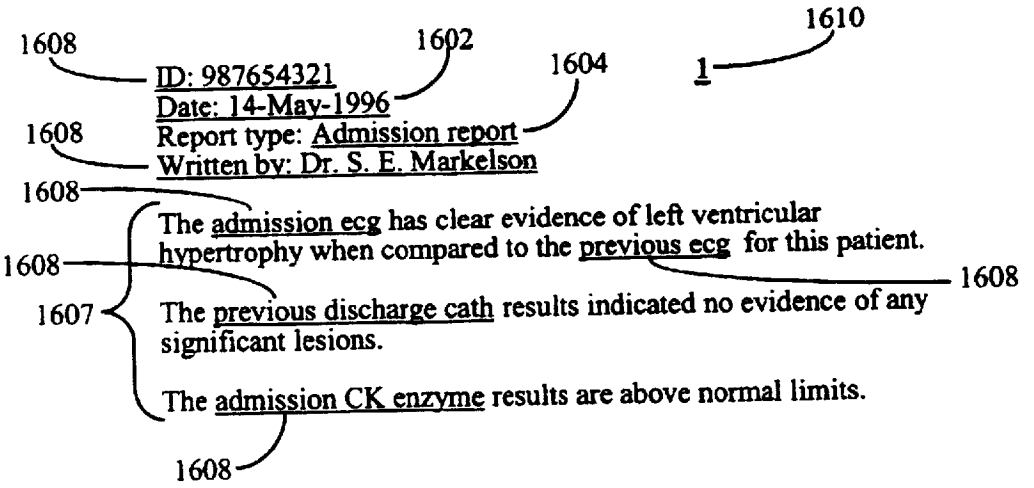


Figure 17

Text after Being Converted to HTML with Hypertext Links Added

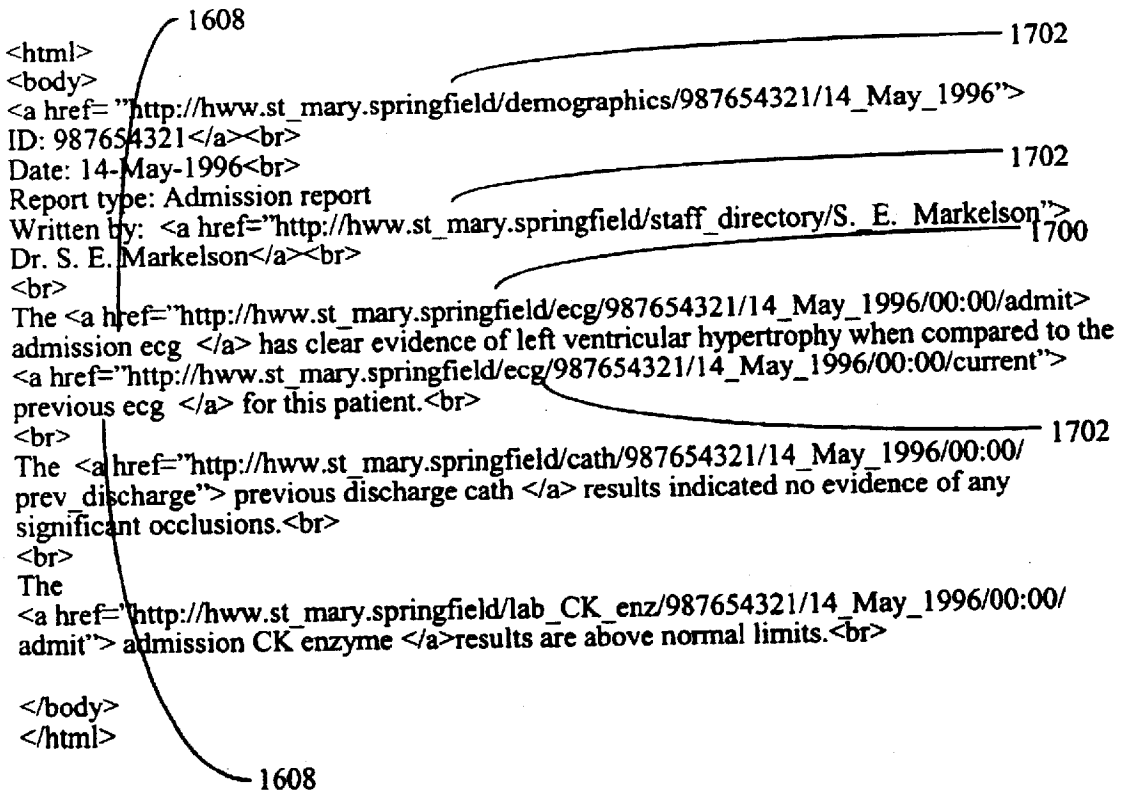


Figure 18

Text As Viewed via Word Processor 14 or Browser

ID: 987654321
Date: 14-May-1996
Report type: Admission report
Written by: Dr. S. E. Markelson

The admission ecg has clear evidence of left ventricular hypertrophy when compared to the current ecg for this patient.

The previous discharge cath results indicated no evidence of any significant lesions.

The admission CK enzyme results are above normal limits.

Figure 19

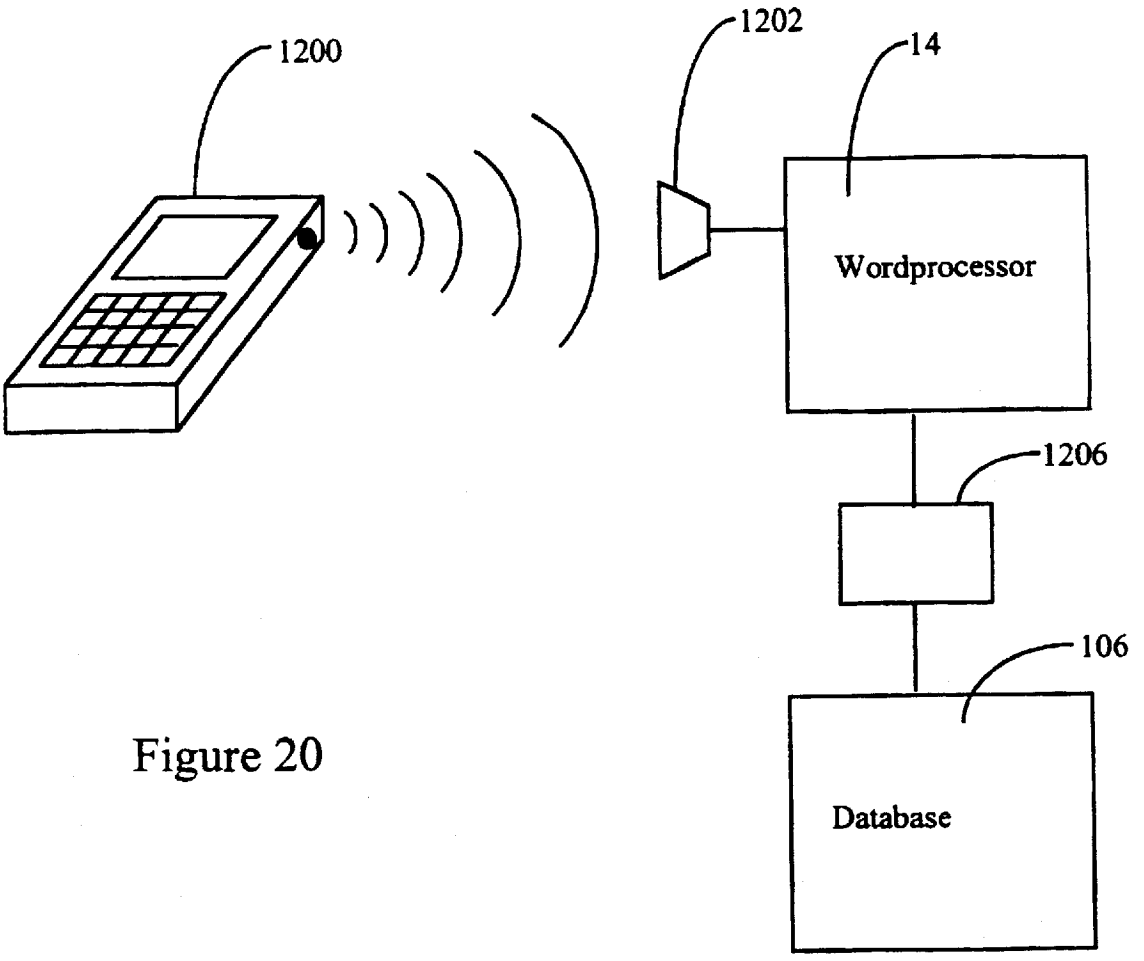


Figure 20

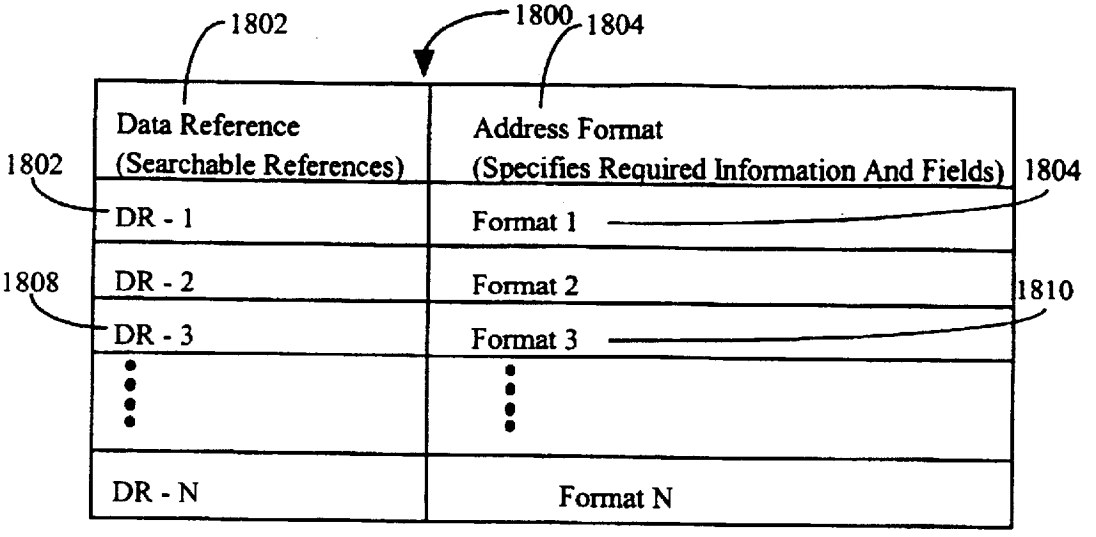


Figure 21

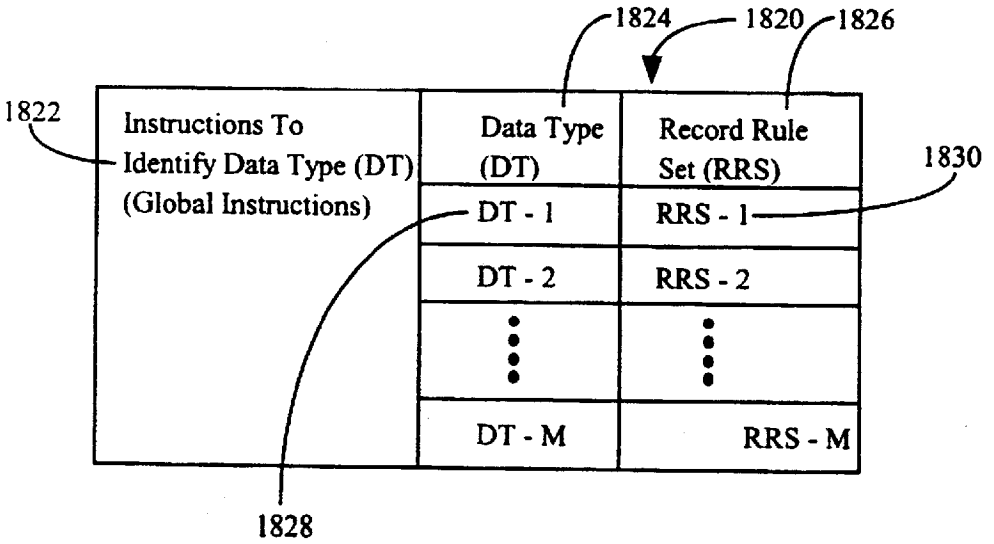


Figure 22

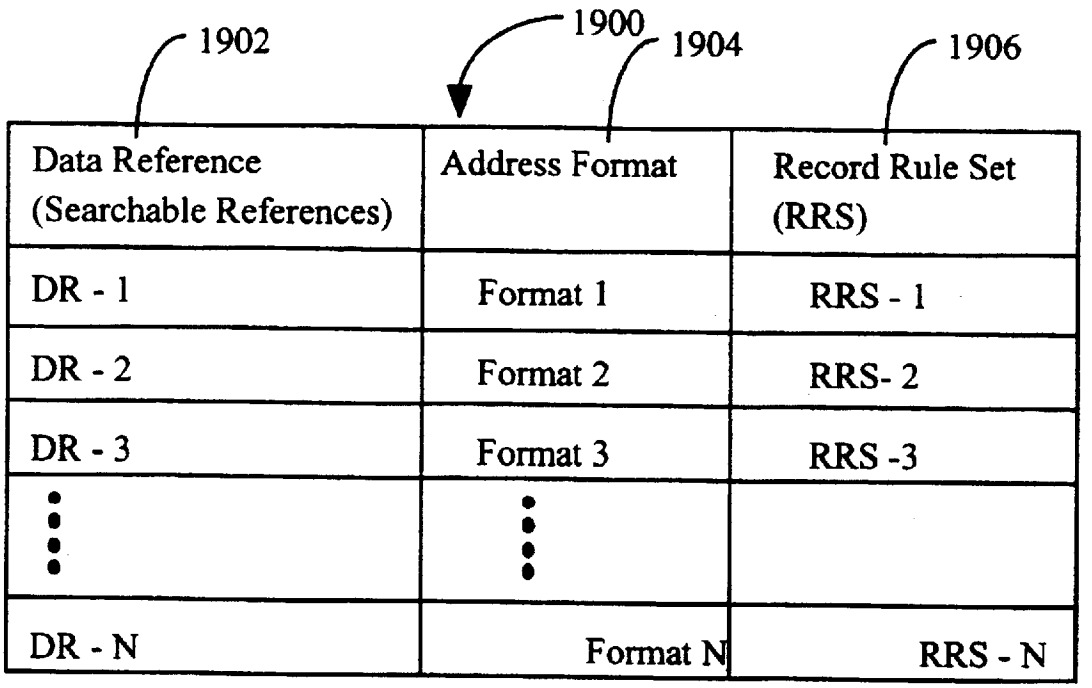

1902
1900
1904
1906
Data Reference (Searchable References)
Address Format
Record Rule Set (RRS)
DR - 1
Format 1
RRS - 1
DR - 2
Format 2
RRS- 2
DR - 3
Format 3
RRS -3
DR - N
Format N
RRS - N

Figure 23

METHOD AND SYSTEM FOR AUTOMATED DATA STORAGE AND RETRIEVAL

CROSS-REFERENCE TO RELATED APPLICATIONS

This patent application is a continuation-in-part of U.S. patent application Ser. No. 09/247,349 which was filed on Feb. 10, 1999, which is entitled "Method and System for Automated Data Storage and Retrieval" and which is a continuation-in-part of U.S. patent application Ser. No. 08/727,293 which was filed on Oct. 9, 1996, is entitled "Method and System for Automated Data Storage and Retrieval with Uniform Addressing Scheme", which issued as U.S. Pat. No. 5,895,461 which claimed priority from provisional application Ser. No. 60/023,126 which was filed on Jul. 30, 1996. Application Ser. No. 09/247,349 was is also a continuation-in-part of U.S. patent application Ser. No. 08/871,818 which was filed on Jun. 9, 1997, is entitled "System and Method for Translating, Collecting and Archiving Patient Records" and which issued as U.S. Pat. No. 5,903,889 and there through the present invention is also a continuation-in-part of U.S. Pat. No. 5,903,889.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not applicable.

BACKGROUND OF THE INVENTION

The present invention relates to the collection, storage, and retrieval of data on computer systems. More particularly, the present invention relates to a computer system for retrieving, modifying, and storing a plurality of topically, textually, or audio-visually related data records of a plurality of formats on a plurality of databases in conformance with a hypertext-linked, predefined topical organization.

When a patient is in a hospital, either as an inpatient or an outpatient, a variety of information concerning the patient may be collected and recorded. This may be in the form of observations, measurements, lab results, vital sign indicators, procedure reports and associated graphics. Over a long period of treatment, hundreds of pages of information may accumulate in the patient's record.

While the patient is in the hospital, it is typical that many different care givers, administrators, or insurance company employees will desire to view a part of the patient's cumulative record. The conventional paper chart is not always useful, as there is only one copy of it, and some laboratory tests may not be entered into the chart on a timely basis. To solve this problem, hospitals have used a variety of database systems such as hospital information systems (HIS) and clinical information systems (CIS) to store and present patient information on computer displays. However, there is still a substantial amount of data that does not get placed into these systems. A variety of factors may inhibit an automated process of comprehensive retrieval of a patient's data, such as incompatible communication protocols and formatting schemes between computer systems, non-digitized data records including pictures and standardized forms, and the lack of adequate computer interfacing support for low-cost medical instruments or devices. It is also typical that word processing documents, rather than being automatically collected by a database system, are simply printed in the form of a paper copy to be inserted into the conventional chart.

While various standardization committees have been established, e.g., HL-7, DOCOM, and IEEE, to develop

common addressing schemes for hospital data, to date none have defined a consistent format to use for storing and retrieving data. For the sake of simplicity or due to limited resources, many manufacturers that use one or more of these standards choose to use only a portion of them; consequently their systems remain only partially compatible.

Furthermore, even many hospitals with database systems lack a centralized retrieval system because related hospital reports are often stored on separate databases. For example, a patient's radiology catheterization report and hemodynamic catheterization reports may be created and stored in separate databases, though as far as the physician who performed the catheterization procedure is concerned, these two reports are really just one procedure and should be associated with each other. For further example, a physician reviewing an admission report may find that it references laboratory tests or observations made contemporaneous with or previous to the patient arriving at the hospital. Should the physician decide to review these other records, she will have to perform additional searches to locate them. In some cases, this often cumbersome and time-consuming process results in care givers refraining from making complete use of the available patient information.

In many hospitals when a patient is discharged, a paper copy of these records is made and sent to the admitting physician for his own record keeping purposes. The collection, copying, and storage of all of these records is a very time-consuming and labor-intensive activity. Further, the generally high risk of human error may manifest itself in the failure to return records to the correct patients file or incorrect storage of a patient's entire file, effectively forfeiting the misplaced information. The physician is simultaneously confronted with the responsibility of filing and storing the paper copy in his own office.

Some hospitals have purchased laboratory or information systems capable of long term storage of various records. While this may assist the hospital in retrieving past records, it may not help the admitting physician in referring to them, for he may not have access to the data directly or may not have the specific software required to retrieve the data. So with such advanced systems the physician is still provided with a paper copy for his records.

Furthermore, many existing laboratory and information systems record information in a variety of inconsistent formats. Some of these formats are proprietary to the manufacturer of the specific system. Each system may use a separate database scheme to gain access to the data. Substantial efforts to get these systems to communicate with each other have not yielded satisfactory results. For example, many large medical information systems use complicated data exchange protocols; but these protocols are unwieldy for simple, often portable instruments which lack the hardware and software capacity to conform to such protocols.

Some reports may be created using a wordprocessor. These may originate in a department of the hospital or in a physician's office. These reports, which may be kept in a conventional file cabinet, are not always included with the rest of the patient's reports.

What is needed is an effective alternative to creating paper records that must be copied and meticulously tracked, an alternative that would permit physicians to access the data economically and easily in their own offices. Such a system would permit a system user to enter a keyword to retrieve a specific data record of a patient, retrieve the requested record from whichever database it is stored to, reformat the data

record with hypertext links to related patient records, and return the requested record to the system user for display on a browser. The system would preferably use a mark-up language such as the well-known Hypertext Markup Language (HTML) or JAVA so that it could utilize inexpensive, standard software packages. The system would also be operable to format data records stored on the various databases of the computer network systematically, periodically, or automatically upon the creation of new, or the modification of existing, data records. The system would be operable to collect all data records pertaining to a specific patient, doctor, or other subject, modify them to support display through a Java applet, Internet browser, or other universal display standard, generate additional patient files to organize the data records in a hypertext directory structure, and store the data records and files on a mass-media storage device such as a CD-ROM.

In addition, it would be advantageous to have a system which permits both the storage and the retrieval of data records according to a standardized addressing scheme which can be determined solely by the use of certain keywords known to the various users of the system. Such a system could employ standard word processing software to enable multiple users to create and reference the various data records. The system would recognize certain keywords entered by the user during creation of the data record and use those keywords to determine the appropriate location (e.g., database, directory and file name) to store the record according to a predetermined addressing scheme. Similarly, it is desirable for the users of the system to be able to locate particular data records using a few keywords without having to know the complexities of which database the record is on, the format of the record, the file name or the directory address.

BRIEF SUMMARY OF THE INVENTION

It is an object of this invention to provide a means of processing and converting existing data records formatted, structured, and accessed according to a multitude of disparate standards to common standards by which they may be accessed, controlled, and/or displayed through a single interactive display program. It is another object of this invention to provide conventions for exploring data records for references to contextually related data records and modifying, generating, embedding, and appending links and data-retrieving codes in and to said related data records, whereby to organize said related data records in a hypertext tree structure. A further object of this invention is to store a group of related data records organized in a hypertext tree structure to a mass storage device, such as a hard disk or CD-ROM, through which the data records may be retrieved, displayed, and controlled through a single interactive display program. In order to minimize costs and maximize end-user accessibility, the standards and conventions used by the present invention for modifying data records and addressing schemes facilitate display through a widely-familiar and low-cost display program such as an Internet browser.

The invention may be adapted for use in a wide variety of applications, and is suitable for any environment in which numerous data records having one or multiple forms and/or formats are to be collected, stored, archived, retrieved, or translated. By way of illustration and not by way of limitation, the invention is presented in the context of a hospital environment, in which typically there are numerous computer systems in use by various health care professionals in one or several hospitals, and each professional often desires to have access to the patient records created by other professionals in that or other hospitals.

A typical setting for the present invention provides multiple databases and workstations linked via a network wherein the databases store data records in a variety of formats and the workstations utilize user interfaces to input, retrieve, and manipulate data records. The present invention utilizes specification tables identifying each of the information processors or databases used by the hospital, the types of data records stored by the databases, and instructions and algorithms for accessing, modifying, and processing data records and their addresses, depending on the data type. Similar specification tables are also kept to identify each workstation where wordprocessor, spreadsheet, or other records, including those downloaded from portable medical devices, may be held.

When a system user at a workstation linked to the hospital computer network equipped with the present invention submits a request for a particular patient record, the invention parses the data request for an address root and other pertinent information about the data record to be retrieved, which may include the time and date the data record was created or last modified and a patient ID. Using this information incorporated in the data request and in the specification tables, the invention modifies the existing data request into a URL or other addressing convention, as necessary, to retrieve the data record from the appropriate database.

After retrieving the data record, the invention may modify it to make it compatible with a standard supported by the common interactive display browser used by the system. For example, the invention may convert a text document to an HTML document or convert graphics, video, or audio records to browser or Java-enabled formats. Further, depending on the formatting specifications for a particular data type, the invention may identify key words, links, and programming codes embedded in the data record, modifying them and inserting additional hypertext links and programming codes as necessary. For example, it may be desirable that a hypertext link referencing the patient's demographics and insurance information be inserted into each record reporting on the patients condition, status, or profile for quick and easy referral. As another example, it may be desirable to place a hypertext link in a radiology catheterization report that references the hemodynamic catheterization report and vice versa so that each refers to the other.

In this manner a hospital may use Internet or Intranet compatible databases with databases that are not compatible, and may choose to use URL addresses of its own choice independent of what the individual vendors have chosen. The administrator may also preprogram the data translation and collection system to link reports together as appropriate, so that care givers may more quickly and directly refer to relevant or related information. The translation process described here may be used on a dedicated system for this purpose or may be distributed among several processors including those of the database systems.

Another aspect of the present invention includes means for receiving, processing, and storing hospital records systematically, periodically, or automatically as they are created or modified. In this mode of operation data records may be pre-formatted according to the hospital's specifications, allowing for quicker record retrieval during subsequent data requests. The translation operation may be allocated to a dedicated system for this purpose or may be distributed among several processors, including those of the database systems.

A further aspect of the present invention includes means for periodically retrieving and filing the contents of a

designated area of each workstation's disk. For example, word processing documents generated at a workstation may be stored in a designated area, such as a special "collection" drive or folder, to which the hospital computer network has access. The invention would retrieve the data records stored in the collection folder, and identify, interpret, and modify them before storing them in an appropriate database.

Yet another aspect of the present invention includes means for retrieving, processing, and storing all of a patient's data records that are available on the hospital's computer network onto a mass media storage device, such as a CD-ROM. For example, this process may be initiated by submitting a collection request identifying the patient's ID number or other identifier uniquely identifying the patient. The invention submits requests, passwords, macros, and programming codes, as appropriate, to each of the databases and workstations that include portions of the patients cumulative record. Each record retrieved is processed and modified as above-as if the particular record had been requested by a system user. The invention not only collects applicable data records, but also multimedia clips, applets, browser extensions, "plug-ins," and other application modules addressed by programming codes embedded in the patients data records. Substitute files explaining the absence of a linked record or module are created for data records or modules regarded as inappropriate for storage and distribution on an unsecured or uncontrolled medium. The invention would also create a "master file for the patient analogous to a "home page" for a website or the root directory of a tree structure, containing links to other patient-related files and data records. The master file may have hypertext links to patient records and to additional (secondary) control files, which in turn have hyperlinks to more patient data. After completing these collection routines, the invention would transfer the collection of data records, applets, browser extensions, and other data and programming modules to a mass-storage device. In this manner a patients cumulative patient record could be stored on a single CD-ROM or other high-density storage device, cheaply distributed to other hospitals or health care professionals serving the patient, and be conveniently accessed by those hospitals and health care professionals.

The present invention also provides a wordprocessor employing a standardized addressing scheme, wherein the wordprocessor recognizes keywords entered by a user who is either creating and storing a data record or attempting to locate a data record among numerous data records at different addresses on a plurality of computer databases.

Thus, the present invention provides first a plurality of databases on which a variety of data records are stored. The databases are in communication with one or more processors which interpret input data from a user interface and direct the storage and retrieval of data records. The databases and processors may be linked via a network, or one or more of the databases may communicate locally with an associated processor, as in a personal computer. The invention also provides a plurality of user interfaces, such as combinations of keyboards, video displays, microphones with voice recognition, and other input devices (e.g. rf receiver, etc.), through which system users create, store, retrieve and display data records. These user interfaces can be simple terminals which communicate with a processor and a database over the network, or they can be part of an integrated interface/processor combination, such as in a personal computer.

For accepting keywords from the user and determining the storage location of a data record to be stored or retrieved, the invention includes a wordprocessor having certain defined functions. For the creation of data records, the wordprocessor accepts various information from the user to identify the user and the type of record being created, as well as other information which may uniquely identify the record and its storage location after the record is completed and saved, or "published." The wordprocessor uses these keywords, or specialized information fields, to determine the location at which the record is to be stored and employs a standardized addressing scheme compatible with or comparable to the Universal Resource Locator (URL) addressing used on the global computer network (Internet). The wordprocessor automatically creates a link between the keyword in the data record and the address of the data record on the computer system.

The wordprocessor also includes a function which compares text entered by the user to a predetermined list of keywords known to be used in the system and may prompt the user for a different keyword when no match is found. Once the user enters a sufficient number of recognized keywords to uniquely identify the data record being created or sought, the wordprocessor determines the unique address of the data record according to a predetermined, standardized addressing scheme so that the record may be stored or retrieved.

When a data record is created and stored, the wordprocessor creates a link, in the manner of a hypertext link, between a keyword uniquely identifying the particular record and its unique address (URL) on the computer system. To this end, the wordprocessor identifies background information within the record and uses the background information and, in some cases, the keyword itself, to form the address link pointing to the record. This link points to the unique address of the record and will enable other users to retrieve the record when the same keyword is used in a request for a data record. In the same manner, other data records containing this same keyword will contain a link to that record, permitting users to create data records which refer to other data records by use of a hypertext link.

The wordprocessor included in the invention contains a monitoring function which monitors free text entered by the user to determine whether the user is creating a hypertext reference at a place in the data record other than in a specified keyword field. This monitoring function continuously surveys text/data being input by the user so that hypertext links in a data record or report can be created by the user at will.

The wordprocessor also includes an editing function which permits keywords, or hypertext references, in data records to be edited and determines whether a user is changing the keyword to another keyword or a non-keyword. This editing function attempts to match changed keywords with other known keywords to determine whether the user is referencing a different data record. The wordprocessor treats keywords or keyword phrases as singularities which cannot be edited without either deleting the link (URL or hyperlink) associated with the keyword or changing the hypertext link to a different hypertext link.

The addressing scheme and hypertext links of the invention are suitable to be created by and used with conventional tools in common use for publishing documents on the Internet. Data records containing keywords and hypertext links may be created in a markup language such as Hypertext Markup Language (HTML), and the addressing scheme may comport with Internet URL addressing. Thus, the invention provides Internet/Intranet capabilities and may be

operated with relatively inexpensive, commercially available HTML formatting software and Internet browser software. Other hypertext link preparation methods and other addressing schemes are possible.

The invention further includes a system for linking first record references to a first record wherein the references are in a second record, the system comprising a database (DB) including at least one address format specifying an address format of the first record address and a processor linked to the DB and running a pulse sequencing program to perform the steps of receiving the second record, analyzing the second record to identify references to the first record and when a first record reference is identified, using information from the second record to form the address of the first record as specified by the address format.

Preferably the address format also specifies required information for forming the address for the first record, the DB further includes at least one record rule set (RRS) corresponding to the address format, the RRS specifying rules for gleaning the required information from a record and, wherein, when the first record is referenced in the second record, the processor gleans the required information from the second record in the manner specified by the RRS.

Also, preferably, the DB also includes a data reference (DR) which is associated with the address format and wherein, when searching for a reference to the first record, the processor searches for instances of the DR. In one aspect the program includes a wordprocessor, the DR is a text name associated with the first record and the first record address is a markup language data reference.

In one embodiment the system is also for creating markup language data references between the first record references and the first record, the processor also, when a first record reference is identified, provides the first record reference to a user as a selectable segment and links the selectable segment to the first record via the first record address such that, when the selectable segment is selected, the first record is provided to the user.

Where a selectable segment is provided, the processor may further, after the selectable segment is provided, perform the steps of, when the second record is accessed, monitor changes to the second record and, when the selectable segment is modified, de-linking the selectable segment and the first record.

In another aspect the DB includes a plurality of address formats and their associated RRSs and DRs. In this case, the processor searches the second record for any of the DRs and, when any of the DRs is identified, the processor identifies the associated address format and RRS, gleans the required information from the second record in the manner specified by the associated RRS and forms the address corresponding to the first record.

In yet another aspect the system includes an interface for entering the second record, the second record entered in record segments and, wherein, the processor runs the program as second record segments are entered via the interface.

In another aspect the system is for use with a data specifying device wherein the step of receiving includes receiving the second record from the data specifying device.

Furthermore, the invention includes a system which receives database records, each record including a separate information set and characterized by at least one data type, for a specific record, the system using the specific record's information set to construct a record address which enables easy subsequent record access, the system comprising a database (DB) including at least one address format which is associated with the at least one record type and which specifies a unique set of required information to form a record address for the record type. The system also including a processor linked to the DB and running a pulse sequencing program to perform the steps of, for the specific record receiving the information set, confirming the data type and the associated address format, analyzing the information set to glean the required information, using the required information to form a record address as specified by the address format and storing the record at the record address.

In one embodiment there are a plurality of data types, the DB includes a separate address format for each of the different data types and the step of confirming includes the steps of determining the data type and the associated address format.

In one embodiment the DB further includes at least a separate record rule set (RRS) corresponding to each of the address formats, each RRS specifying a unique set of rules for gleaning required information from a record and, wherein, the processor gleans required information in the manner specified by the RRS.

Preferably the program is a first application program and the processor also performs a second application program to link stored records which are referenced in a first record to the referenced records, to this end the processor further performing the steps of, after at least one record is stored, searching the first record for a reference to a stored record, when a reference to a stored record is identified, determining the address associated with the referenced record, providing the reference to a user as a selectable segment and linking the selectable segment to the referenced stored record via the record address such that, when the selectable segment is selected, the record is provided to the user.

Preferably, the processor provides a data reference (DR) for the record information set, the DR useable to refer to the record in other records, the processor, when searching for a reference in the first record, searching for the DR.

Also, preferably, after the record address is formed, the processor also correlates the DR with the record address and stores the DR along with the record address, the processor determining the address associated with a reference by identifying the address associated with an identified DR.

In one embodiment the RRS is a first RRS and the DB also associates a second RRS with the address format, the second RRS specifying rules for gleaning the required information from the first record, when a DR is gleaned from an information set, the processor also correlating the DR with the address format and storing the DR along with the address format, the processor determining the address associated with a reference by, when a DR is identified, identifying the address format associated with the DR, identifying the second RRS associated with the identified address format and the required information specified by the identified address format, gleaning the required information from the first record as specified by the second RRS and forming the record address using the required information and as specified by the address format.

The invention further includes a system which receives database records, each record including a separate information set and characterized by at least one data type, for a specific record, the system using the specific record's information set to identify a record address which enables easy subsequent record access. The system is also for use with a data specifying device which provides the database records, including at least one field specifying a data reference. The

system comprises a receiver for receiving records from the specifying device and a processor linked to the receiver and running a pulse sequencing program to perform the steps of, for a specific record, receiving the information set, identifying the DR, using the DR to identify a record address for the record and storing the record at the record address.

The specifying device may be a hand held device or a database or some other suitable specifying device (e.g. a dictaphone). In addition, the specifying device, in addition to specifying the DR, may also specify other information which is used to identify the address.

According to yet another embodiment of the invention, a record being searched may be characterized by a data type and the data type may be associated with a specific RRS for gleaning information therefrom and, when a DR is identified in a record, the system may determine the data type of the searched record to identify the RRS to be used to glean the required information.

Moreover, the invention further includes methods to be used in conjunction with the systems described above.

These and other objects, advantages and aspects of the invention will become apparent from the following description. In the description, reference is made to the accompanying drawings which form a part hereof, and in which there is shown a preferred embodiment of the invention. Such embodiment does not necessarily represent the full scope of the invention and reference is made therefor, to the claims herein for interpreting the scope of the invention.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

FIG. **1** is a block diagram of a medical computer network according to the present invention, including a plurality of databases for data record storage and a data translation and collection system;

FIG. **2** is a graphical illustration of a physician office workstation;

FIGS. **3A** and **3B** are tables showing the contents of the "Database Table" and "File Format Instruction Table" maintained and used by the data translation and collection system to access, translate, reformat, and store data records kept on databases on the medical computer network;

FIGS. **4A** and **4B** are tables showing the contents of "List of Workstations" and "Report List" maintained and used by the data translation and collection system to access, translate, reformat, and store data records kept on workstations on the medical computer network;

FIGS. **5A–5F** are functional flow charts showing the steps used to collect and process a related set of data records from various databases, create a structured set of control files containing hypertext links to the collected data records, and store the data records and control files to a storage device;

FIG. **6A** is a graphical representation of a master file in HTML format through which all of a single patient's medical records created at a hospital equipped with the present invention may be viewed;

FIG. **6B** is a graphical representation of the master file of FIG. **6A** as viewed by a system user with a network browser;

FIG. **7A** is a graphical representation of a secondary control file in HTML format providing a hypertext-link embedded discharge report;

FIG. **7B** is a graphical representation of the secondary control file of FIG. **7A** as viewed by a system user with a network browser;

FIG. **8A** is a graphical representation of another secondary control file in HTML format providing a structured list of hypertext links to a plurality of cardiology reports;

FIG. **8B** is a graphical representation of the secondary control file of FIG. **8A** as viewed by a system user with a network browser;

FIG. **9A** is a graphical representation of a tertiary control file in HTML format providing a list of electrocardiogram reports;

FIG. **9B** is a graphical representation of the tertiary control file of FIG. **9A** as viewed by a system user with a network browser;

FIG. **10** is a table showing the contents of the "Data Request Catalogue" maintained and used by the data translation and collection system to discriminate the data type and database location of a requested data record from the alphanumeric string requesting the data;

FIG. **11** is a graphical representation of a data request containing an address root and descriptor;

FIGS. **12A–12C** are a functional flow chart showing the steps used to receive a request for a data record, translate the request, retrieve the data record, and reformat the data record prior to sending it to its requested destination;

FIGS. **13A–13C** are a functional flow chart showing the steps by which the data translation and collection system processes a data record received or retrieved from a workstation or database system on the medical computer network, reformat the data record, assign it a URL address, and deliver it to a database for storage;

FIGS. **14A–14B** are textual representations of a URL address as received and reformatted by the data translation and collection system;

FIG. **14C** is a graphical representation of a report referenced by the URL address of FIG. **14B** as it would be viewed by a system user through a network browser;

FIG. **14D** is a textual representation of the report of FIG. **14C** as modified to include data references in the form of HTML codes;

FIG. **14E** is a graphical representation of the modified report in FIG. **14D** as it would be viewed by a system user through a network browser;

FIGS. **15A–15B** are a functional flow chart showing the steps by which a data record is parsed to locate data references within it;

FIG. **16** is a flow chart illustrating a method of forming or building record addresses using a record rule set and an address format in real time according to the present invention;

FIG. **17** is a is a graphical representation of a sample patient report during its creation by a user of the computer system according to the invention;

FIG. **18** is a graphical representation of text of the report of FIG. **17** after being converted to HTML format and having hypertext links to URL addresses substituted for their associated data references;

FIG. **19** is a graphical representation of the report of FIG. **17** with hypertext links, as viewed by a system user with a network browser or other request handler routine;

FIG. **20** is a schematic of specifying devices and a processor of the present invention;

FIG. **21** is a table used by the wordprocessor of the present invention to build data base addresses for linking first record references in a second record to the first record;

FIG. **22** is a table used in conjunction with the table of FIG. **21** for gleaning information from a second record which is required to form a data base address for a first record which is referenced in a second record; and

FIG. **23** is a table similar to the tables of FIGS. **21** and **22** which is used in another embodiment of the present invention to build data base addresses for linking first record references in a second record to the first record.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to FIG. **1**, the invention is illustrated as a medical computer network **100**, including a plurality of hospital based workstations **102** (which may be personal computers), a plurality of physician office workstations **104** which may also be personal computers, a plurality of databases **106** which may be provided by a multitude of vendors with separate data structures and data elements. The computer network **100** may also comprise an Admit, Discharge, and Transfer (ADT) system **108**, a data translation and collection system **110**, and a Hospital Information System (HIS) **111**. The data translation and collection system **110** is not necessarily a separate physical element of the medical computer network **100**, but is represented that way in the preferred embodiment for purposes of illustration only. It may be alternately recognized as a program application or even an aspect of a network operating system, the operations of which may be distributed over and performed by many different processors, workstations, and databases on the medical computer network **100**.

Databases **106**, computer systems **108**, **110**, **111**, workstations **102**, and physician office workstations **104** may communicate with each other via a communication network **112**, which may be a combination of local and wide area networks, using Ethernet, serial line, wireless, or other communication standards. Communication network **112** may also be arranged in such a manner to be part of the Internet or as an individual Intranet Each workstation **102**, **104** includes a "collection" folder **105**, a user interface **103** which may include a network browser or similar display, entry, and retrieval program, a separate database **2** and a special wordprocessor **14**. User interface **103** may be any means for permitting users to create data records and/or retrieve data records from the medical computer network **100** capable of supporting a network browser, such as well known keyboard and video terminal combinations or voice recognition hardware and software.

Wordprocessor **14** runs under the direction of user interface/processor **103** and governs the creation of data records, the recognition of keywords in the data records, the composition of hypertext links between the keywords and the data records, and the retrieval of data records in response to keywords input to the wordprocessor by the user or contained within a data record.

FIG. **2** shows a typical physician office workstation **104** comprising a personal computer **107** which may include a display **118**, and a CD-ROM drive **117** or other means of mass storage which may be removable.

The data translation and collection system **110** maintains a file referred to as Database Table **130**, whose contents are partially seen in FIG. **3A**. For each database **106** included on the medical computer network **100**, an entry is made in the Database Register **131** of the Database Table **130**. Corresponding to each entry in the Database Register **131** is an address or addresses field **132** used to access the database on communication network **112** and a separate File Format Instruction Table **134**.

A partial list of the contents of File Format Instruction Table **134** is seen in FIG. **3B**, which includes records of each data type **136** stored by the database **106**. Corresponding to each data type **136** in File Format Instruction Table **134** is a set of special instructions or program codes **142** used to translate a request for such data to format appropriate to the data type and database from which the requested information may be retrieved. Also corresponding to each data type **136** is a hypertext cipher or record rule set (RRS) **138** providing special instructions or codes used to add data references (such as hypertext links) and to format the data, which instructions or codes may include decompression algorithms. In addition, the RRS **138** also specifies rules for gleaning information from a record which can be used to form a record address. Hereinafter the terms hypertext cipher and RRS may be used interchangeably to mean the same thing.

Further corresponding to each data type **136** is a URL cipher or address format **140** used to generate an address to store the designated type of data. The address format **140** specifies information which is required to form an address for an associated record type and also specifies the order of the information in the resulting address. Hereinafter the terms address format and URL cipher may be used interchangeably to mean the same thing.

The data translation and collection system **110** may also retain a file referred to as Data Request Catalogue or database table (DBT) **500**, whose contents are partially seen in FIG. **10**. The Data Request Catalogue or DBT **500** includes an array of Data Request Address Roots or data references (DRs) **504**, to each element of which corresponds fields identifying the data type **136**, the database **106** in which the data type is located, and hypertext cipher **138** (which is kept also in the File Format Instruction Tables **134** (FIG. **3B**) of the Database Table **130** (FIG. **3A**)). This file may be accessed when a request for data is received by the data translation and collection system **110** to recognize the matching data request address root **504** which identifies the data type **136** and the database **106** on which it is kept.

A. Responding to Data Requests and Providing Translation

FIGS. **12A–12C** describe the operation of the data translation and collection system **110** (FIG. **1**) in responding to requests to retrieve data, translating those requests to conform to the format required by the applicable database, retrieving the data, reformatting the data, and delivering the data to the appropriate destination.

Commencing with FIG. **12A**, in step **540** the data translation and collection system **110** receives a data record reference **520** (FIG. **11**) in the form of a data request containing an address root **522** and descriptors **524** about the requested record. In some instances, a data request will originate from a system user accessing a hypertext link on a document displayed by the system's interactive display browser. In other instances, the data request will originate from a database or workstation application program. There may be several non-uniform but mutually distinguishable data request formats among the several hospital databases **106** (FIG. **1**) on the medical computer network **100** (FIG. **1**). Alternately, data requests may be uniformly and compatibly formatted for all records stored by various hospital databases **106** (FIG. **1**). For example, the data requests may be in the form of a URL with optional data fields sent with it to assist in identifying the record to be retrieved.

In step **544**, the address root **522** (FIG. **11**) of the data record reference **520** may be determined by removing the descriptors **524**-any patient identification, chronological details, or other non-addressing information-from the

received data request. The descriptors **524** are temporarily stored for use in step **560**.

In step **548**, a search is performed to locate a match for the address root **522** (FIG. 11) of the data record reference **520** among the data request address roots **504** (FIG. **10**) listed in the data request catalogue **500** (FIG. **10**). By finding a matching data request root address **504**, the invention immediately identifies the data type **136** (FIG. **10**), the database **106** used to store this data, and the hypertext cipher **138** providing special instructions used to format and add data references to the data.

In step **552** the database **106** identified in step **548** is in turn referenced in Database Table **130** (FIG. 3A) to its corresponding File Format Instruction Table **134** (FIG. 3A) to determine the address(es) **132** (FIG. 3A) of the database **106** storing the data.

In step **556** the data type **136** identified in step **548** is cross-referenced with the File Format Instruction Table **134** (FIG. 3A) identified in step **552** to locate the special instructions to request data **142** (FIG. 3B) used to translate the request to a format appropriate to the data type and database from which the requested information may be retrieved. These instructions may, for example, include passwords or macros needed to retrieve the data.

In step **560** a code is constructed using the database address(es) **132** identified in step **552**, the special instructions to request data **142** identified in step **556**, and the descriptors **524**-the patient identification, chronological details, or other non-addressing information-stored in step **540**. The code is submitted to the appropriate database to produce the requested data record.

After the database has produced the requested data record, the record may in step **564** be received by the data translation and collection system **110** for additional processing.

The steps by which the data translation and collection system **110** processes the selected data record are shown in FIGS. 12B and 12C. In step **566**, the system uses the hypertext cipher **138** to determine whether or not the data is stored in a proprietary format. If it is, the applicable proprietary software is used to decompress or translate the data. This may be done on the manufacturer's database **106**, another computer processing system, or by the data translation and collection system **110** itself. in step **570**, the record is parsed, discussed intra, to locate date references: hypertext links, multi-media requests, and key words or phrases. If none are found, the process advances to step **598**, discussed infra.

If there are data references, they may in steps **580** through **596** be reformatted so that the URL addresses are compatible with addressing protocols used by the hospital. In step **580**, the address root **522** (FIG. 11) of the hypertext link or other data record reference **520** may be determined by removing the descriptors **524**—any patient identification, chronological details, or other non-addressing information—from the received data request. The descriptors **524** are temporarily stored for use in step **596**. In step **584**, a match for address root **522** is sought among the data request address roots **504** (FIG. **10**) listed in Data Request Catalogue **500**, which locates the Database **106** and Data Type **136** corresponding to the matching Data Request Address Root **504**. In step **592**, the identified Database **106** and Data Type **136** are referenced in Database Table **130** (FIG. 3A) and the corresponding File Format Instruction Table **134** (FIG. **38**) to acquire the appropriate URL cipher **140** (FIG. 3B). In step **596**, the URL cipher **140** processes the descriptors **524**—the patient identification, chronological detail, and other information—extracted in step **580** to modify or replace the hypertext link or other data reference found in the selected record. Steps **580** through **596** may be performed for each hypertext link and reference to other data records found in the selected record.

For some types of data records, the URL cipher **140** will generate addresses compatible with database formatting standards such as SQL or Oracle.

In step **598** the data translation and collection system **110**, using the Hypertext Cipher **138**, converts any text portion of the selected data record into a browser compatible format, such as HTML format, and any graphics, audio, video, or other non-text information into a browser, plug-in, or Java compatible format.

In step **600**, the data translation and collection system **110** inserts hypertext links or other references to the selected record in accordance with the hypertext cipher **138** identified in step **548**. If directed by the hypertext cipher **138**, the record may also be interpreted and modified or reformatted.

In step **604**, the data translation and collection system **110**, having retrieved and translated the requested record, forwards the record to the destination selected by the requesting workstation or processor.

B. Receiving Patient Records for Translation and Address Formatting

FIGS. 13A–13C set forth an alternate embodiment of the operation of the data translation and collection system **110** (FIG. 1) with particular reference to receiving, translating, and formatting data records to facilitate access through browsers and hypertext links for future users. This embodiment is similar to that set forth in FIGS. 12A–12C but may proceed independently of and prior to a request for such data. Thus in this embodiment the data translation and collection system **110** may serve to organize and format a patient's records prior to their being requested by a member of the medical staff.

Commencing with FIG. 13A, in step **640** the data translation and collection system **110** receives a data record from a database **106** which may include or be appended to other information specifying patient identification, chronological detail, the data type, and other information regarding the record. In step **644**, the data translation and collection system searches each File Format Instruction Table **134** (FIG. 3A) corresponding to each entry in Database Register **131** of Database Table **130** to locate the data type **136** of the received data record. In step **648**, the hypertext cipher **138** (FIG. 3B) of File Format Instruction Table **134** may be used to parse the record to identify additional information, such as patient information and the date and time of the record.

In step **650**, the system uses the hypertext cipher **138** (FIG. 3B) to determine whether or not the data is stored in a proprietary format. If it is, the applicable proprietary software is used to decompress or translate the data. This may be done on the manufacturer's database **106**, another computer processing system, or by the data translation and collection system **110** itself.

In step **654**, the record is parsed, discussed infra, to locate data references: hypertext links, multi-media requests, and key words or phrases. If none are found, the process advances to step **682**, discussed infra.

If hypertext links or references to other data records are found, they may in steps **664** through **680** be reformatted so that the URL addresses are compatible with addressing protocols used by the hospital. In step **664**, the root address

522 of the data record reference 520—which may be in the form of a hypertext link—is extracted as in step 544 (FIG. 12A). Similarly, any descriptors 524—such as patient identification, chronological detail, or other non-addressing information—contained in the data record reference 520 is also extracted and temporarily stored. In step 668, a match for this address root is sought among the Data Request Address Roots 504 (FIG. 10) listed in Data Request Catalogue 500 (FIG. 10), which locates the Database 106 (FIG. 10) and Data Type 136 (FIG. 10) corresponding to the matching Data Request Address Root 504. In step 672, the data type 136 and Database 106 identified in step 668 are cross-referenced with their corresponding File Format Instruction Table 134 (FIG. 3A) to locate the special instructions to request data 142 (FIG. 3B). In step 676, the identified Database 106 and Data Type 136 are referenced in Database Table 130 (FIG. 3A) and the corresponding File Format Instruction Table 134 (FIG. 3B) to acquire the appropriate URL cipher 140 (FIG. 3B). In step 680, the URL cipher 140 processes the descriptors 524—the patient identification, chronological detail, and other information—extracted in step 664 to modify or replace the hypertext link or other data reference found in the received data record. Steps 664 through 680 may be performed for each hypertext link and reference to other data records found in the received data record.

In step 682 the data translation and collection system 110, using the Hypertext Cipher 138, converts any text portion of the selected data record into a browser compatible format, such as HTML format, and any graphics, audio, video, or other non-text information into a browser, plug-in, or Java compatible format.

In step 684, the data translation and collection system 110 inserts hypertext links or other references to the received data record in accordance with the hypertext cipher 138 (FIG. 3B) identified in step 548. If directed by the hypertext cipher 138, the record may also be interpreted and modified or reformatted. In step 686, the URL cipher 140 corresponding to the data type 136 (FIG. 3B) of the received data record processes the descriptors 524—the patient identification, chronological detail, and other information stored or extracted in steps 640 or 648—to format a URL by which the received data record may be accessed.

In step 688, the data translation and collection system 110, having translated and formatted the received data record, forwards the record and its formatted URL to an appropriate database 106 for storage.

C. Operation of the Hypertext and URL Ciphers of the Data Translation and Collection System

FIGS. 14A–14D set forth an example of the hypertext and URL processing performed by the data translation and collection system 110 (FIG. 1) in response to a request for a data record. FIG. 14A proffers, by way of example, a URL address 700 that may be consistent with a standard hospital format, that is received by the data translation and collection system 110. Embedded in this URL address 700 is information regarding the type of data 704, the patient's identification 708, the date 712 and time 716 of the data requested, and a report designator 718. The type of data 704, combined with additional information, is an example of an address root 701 and the information referred to as 708, 712, 716 and 720 are examples of descriptors 702. The data translation and collection system 110, by following steps 544 through 560 as set forth in FIG. 12A, reformats the data request into a new data request 720 (FIG. 14B), which is compatible with the database system 106 (FIG. 1) holding this data.

FIG. 14C sets forth an example of a report 724 that may be produced by a database 106 in response to the data request 720 of FIG. 14B. Initially, the report is only a conventional text document. The data translation and collection system 110 (FIG. 1) may then convert the report into an HTML-compatible format 732 (FIG. 14D), inserting data request 736 and hypertext links 740 and 744 according to the hypertext cipher 138 (FIG. 3B). The hypertext links 744 may be inserted based upon the recognition of phrases or special character sequences, such as "Catheterization Reports" 728, in the report, which may vary from report to report of the same data type depending on the each report's contents.

FIG. 14E shows the text report 724 with imported image 737 as displayed on computer display 118 using a network browser software package after the report has been translated and modified. A system user seeking additional information regarding the patient's demographics could select hypertext link 740. A system user seeking either the radiology or hemodynamic report for this procedure could select the appropriate hypertext link 744.

D. Data Collection and Translation for Storage and Retrieval

FIGS. 5A–5E set forth a second alternate embodiment of the operation of the data translation and collection system 110 (FIG. 1) with particular reference to the steps used by the data translation and collection system 110 to retrieve and format data to produce a complete, organized, hypertext-linked, and browser-compatible collection of records pertaining to a person, place, thing, or event. This operation may be initiated by a system user executing the appropriate command or may be executed routinely and automatically by the hospital's Admit, Discharge, and Transfer (ADT) system or Hospital Information System (HIS) during a patient's stay or when a patient is discharged.

In step 200, the data translation and collection system 110 (FIG. 1) receives a patient identification number, which may originate from a staffed workstation 102 (FIG. 1) or automatically from the ADT system 108 (FIG. 1) or HIS 111 (FIG. 1). This may be done, for example, when a patient is admitted or after one has been discharged. In step 204, the data translation and collection system 110 may request the dates for which the system user desires to collect data for the patient or the most recent admission dates from the ADT system 108 or HIS 111. In step 208, a file containing a list of received records will be opened, if previously created, or generated, if not previously created. Similarly, in step 212, a file containing a list of records to be retrieved is opened, if previously created, or generated, if not previously created. In step 216, the data translation and collection system 110 references database table 130 (FIG. 3A) to locate and retrieve the first database entry in the database register 131 (FIG. 3A), the corresponding File Format Instruction table 134 (FIGS. 3A and 3B) and the first data type 136 (FIG. 3B) in the File Format instruction table 134.

Steps 220 through 276 set forth an iterative search of all the databases 106 (FIG. 1) on the medical computer network 100 (FIG. 1) to collect, translate, and format all records relevant to a patient's medical history.

In step 220, the field containing special instructions to request data 142 (FIG. 3B) corresponding to the data type 136 (FIG. 3B) being referenced by the data translation and collection system 110 is used to construct and format a message which is sent, using address 132 (FIG. 3A), to the database 106 being referenced in database register 131 (FIG. 3A). This message may incorporate passwords, macros, or

other codes, as necessary, to retrieve the data. In step **224** a data record is retrieved.

In step **226**, the record is parsed per the Hypertext Cipher **138** (FIG. 3B) in the File Format Instruction Table **134** (FIG. 3B) to derive the date and time of the record. In step **228** the file name is added to the list of received records opened in step **208**. In step **232**, the list of records to retrieve opened in step **212** is checked for a reference to the retrieved record. If a reference exists, in step **236** it is removed from the list of records to retrieve.

In step **240**, the system uses the Hypertext Cipher **138** (FIG. 3B) to determine whether or not the data is stored in a proprietary format. If it is, the applicable proprietary software is used to decompress or translate the data. This may be done on the manufacturer's database **106**, another computer processing system, or by the data translation and collection system **110** itself.

In step **246**, the record is parsed, discussed infra, to locate data references: hypertext links, multi-media requests, and key words or phrases. If none are found, the process advances to step **266**, discussed infra. If there are data references, a check is made to determine if the data being referenced had been located previously (step **254**). If it had not been previously located, the record is added to the List of Records to Retrieve (step **258**). In ste **262**, all hypertext links and other data requests are reformatted through use of the URL Ciphers **140**, maintained by the data translation and collection system **110** for each Data Type **136**. This is done in a manner similar to steps **580** to **596**, discussed supra. Thus when the retrieved data record is later displayed, secondary files referenced by it will be included for display and the system user will not be presented an incomplete record.

In step **266**, whether there were data references or not, Hypertext Cipher **138** is used to convert text to HTML format, graphics, audio, video, or other non-text information into a browser, plug-in, or Java compatible format.

While not shown in the flow chart of FIGS. **5A–5E**, if the data translation and collection system **110** retrieves a record that includes a program code module such as a Java applet, the data translation and collection system will attempt to retrieve a copy of the applet from an address specified by the applet program code, generate a new address for the applet copy which will be stored with the patients data record collection, and modify the program code module to reflect the new address. Similarly, if the data translation and collection system **110** retrieves a record that requires a browser extension or "plug-in" in order to be properly viewed, a copy of the applicable extension or "plug-in" is also retrieved for storage with the patient's data record collection.

For purposes of privacy or security, the medical computer network **100** may deny access to some data records in the list of records to be retrieved. In such instances a substitute file, indicating that the requested file is confidential or has not been included, is created and stored, and its reference substituted for the reference to the confidential data record.

In step **268**, a retrieved data record may be further modified, such as by inserting additional hypertext links or data requests to the record per the hypertext cipher **138** (FIG. 3B). Also, the URL cipher **140** corresponding to the data type **136** (FIG. 3B) of the retrieved data record is used to format a URL by which the retrieved data record may be accessed. Further, the data translation and collection system **110** creates and opens an appropriate file folder and file to store the converted retrieved record as specified by the URL cipher **140** field.

In step **276**, the File Format Instruction Table **134** (FIG. 3B) for the instant database is checked to determine if additional data types **136** are available. If so, in step **272** the process of steps **220** through **276** is repeated for the next data type **136**. If the search has been performed for each data type in the instant database, the search proceeds to the next database indicated in database register **131** (FIG. 3A), starting with its first data type **136** and proceeding, in similar fashion, through each of its data types **136**, until the search has been performed for every data type **136** of every database in database table **130** (FIG. 3A). The procedure progresses to step **284** after completing this search through the registered databases.

In step **284**, the list of records to retrieve opened in step **212** is examined for the existence of records or program modules that have not yet been retrieved. If the list is empty, the data collection for the patient has been completed and the process advances to step **324** (FIG. 5E), discussed infra. If the list is not empty, in step **288** a request is sent for the first entry remaining in the list, which may be for a data record or a program module. If it is a data record, after it is retrieved, it is checked in step **290** for encryption and decoded, if necessary, using proprietary software.

In step **298**, the record is parsed, discussed infra, to locate data references: hypertext links, multi-media requests, and key words or phrases. If none are found, the process advances to step **314**, discussed infra. If there are data references a check is made to determine if the data begin referenced had been located previously (step **302**). If it had not been previously located, the record is added to the List of Records to Retrieve (step **306**). In step **310**, all hypertext links and other data requests are reformatted through use of the URL Ciphers **140**, maintained by the data translation and collection system **110** for each data type **136**. This way, the URL or other data request addresses are compatible with the addressing convention to be used on the storage medium to which the records will be written. When the retrieved data record is later displayed through a network browser, secondary files referenced by the retrieved data record are made easily and quickly accessible to the system user with the click of a mouse.

In step **314**, whether there were data references or not, Hypertext Cipher **138** is used to convert text to HTML format, graphics, audio, video, or other non-text information into a browser, plug-in, or Java compatible format.

In step **316**, the data translation and collection system **110** creates and opens an appropriate file folder and file to store the converted retrieved record, either as specified by the URL cipher **140** field (FIG. 3B) (if the retrieved record is part of the patient's file), or with a distinctive file name (if the retrieved record is not part of the patient's file, e.g., a physician's biographical background). In step **320**, the retrieved record or program module, as it may be, is removed from the list of records to retrieve, and steps **284** through **320** are repeated until the list is empty.

E. Workstation Data Collection and Translation

The operation of the data translation and collection system **110** (FIG. 1) set forth in FIGS. **5A–5E** may be initiated in other ways. In one mode of operation, the databases **106** (FIG. 1) on the hospital's communication network **112** may send data for each patient to the data translation and collection system **110** periodically or after the patient is discharged. In another mode of operation, the workstations **102** on the hospital's communication network **112** may send reports, such as those produced by wordprocessors, to the

data translation and collection system for translation and storage. In yet another mode of operation, the data translation and collection system may have access to a drive, directory, or folder on one or more of the workstations **102** residing on the communication network, from which it may search and retrieve data records.

System users such as physicians often produce reports, such as word processing files, on their own workstations **102** or **104** (FIG. **1**) that are relevant to a patient's condition, status, or profile, and which merit inclusion in the data translation and collection system **110** of the present invention. This need may be accommodated by placing any report that is to be retrieved by the data translation and collection system **110** in a special folder **105** named "Collection." The data translation and collection system **110** may maintain a file containing a Workstation Data Table **150**, as set forth in FIG. **4A**, which includes the addresses **152** of all workstations **102** and physician office workstations **104** and file access commands **154** or passwords used to gain access to files stored in each workstation's "Collection" folder **105**. The data translation and collection system **110** may also maintain a Workstation File Formatting Instruction Table **158**, as set forth in FIG. **4B**, which includes each report name and corresponding file name and data formatting instructions **162** and Workstation URL cipher **166**.

On a periodic basis or as instructed, a program in the data translation and collection system **110** (FIG. **1**) may determine if there are any files in the special "Collection" folder **105** in each workstation **102, 104**. If any files exist, the file access commands **154** (FIG. **4A**) may be sent to the workstation so that the files may be transferred to the data translation and collection system **110**. This may be done using the file transfer protocol, FTP, of the Internet/Intranet or by other data transfer methods.

If the user of the workstation **102, 104** creates reports, that when stored use a file name formatted according to file name and data formatting instructions **162** (FIG. **48**), the file may be recognized as being a specific file for a patient. For example, the file named "Cath987654321" may correspond to a catheterization report for the patient whose identification number is 987654321. Appending the date and time to the file name may be used to further identify the report. Alternatively, file name and formatting instructions **162** may require that the date and time be located within the report itself. Similarly, the report name and/or the patient's identification information may be incorporated in the report or its file name. In either case, once the file and its file name are received and recognized, the file may be processed in the same manner that data records retrieved from databases are processed as set forth in FIGS. **5A–5E**, but using the Workstation Data Table **150** and its Workstation File Formatting Instruction Tables **158** in place of the Database Table **130** and its File Formatting Instruction Tables **134**.

Instruments or medical devices whose reports are not stored as part of any database **106** (FIG. **1**), but that are capable of writing data to a floppy disk or transmitting information via an infrared or serial line connection to a workstation **102, 104** may also store patient reports in the data translation and collection system **110**. To do so, the individual reports may be written to a floppy disk using any file name defined by the file name and data formatting instructions **162** (FIG. **4B**). The reports may be copied manually or automatically from the floppy in workstation **102, 104** to the "Collection" folder **105**, which may be periodically checked to see if there is any data in it to be retrieved, or the reports may be automatically read by the workstation and sent to the data translation and collection system **110**. The reports so sent may be incorporated with any others received for this patient and may be provided with a new destination file according to the Workstation URL Cipher **166**.

In this manner reports from wordprocessors or from mobile medical devices may be collected for display and storage and may be assigned structured file names to assist in their retrieval whether on line or when placed on line as with a CD-ROM device **117** (FIG. **2**).

F. Creation, Structuring, and Mass-Media Storage of Patient Data

Commencing with step **324**, FIG. **5F** sets forth the process by which a patient's many data records may be stored on a mass storage device. In step **324**, a master file **400** (FIG. **6A**) is created as data records are received or when all the data records have been retrieved. The master file name may be based on the patient's name or identification number. In some cases it may be desirable to create the same file twice, using the patient's name for the file name once and the patient's identification number for the other. In step **328**, secondary, tertiary, or other subdominant control files **418** (FIG. **8A**) may also be created. A "master" file is roughly analogous to a root directory or a home page on a website, for through this single file all the patient's data records may be accessed through hypertext links. While a "master" file may contain text, graphics, video, or audio, it contains a list of links to other reports-for example, the discharge report link **408** (FIG. **6B**) to discharge report **412** (FIG. **7A**) —or links to other "control" files—for example, the cardiology link **404** to cardiology control file **418** (FIG. **8A**). "Control" files are roughly analogous to subdirectories. Although they incorporate text, graphics, or other multimedia features, they serve primarily to present an organized collection of links to related patient records.

The URL Cipher **140** corresponding to the date type **136** of each data record retrieved, can be used to determine whether a hypertext link to retrieve the data record from the mass storage device is to be placed in a master, secondary, or tertiary control file.

In step **332**, the master and control files are written to a mass storage device along with the data records if they have not been previously been written to the mass storage device. A CD-ROM disk that has a patient's data written to it may be given to appropriate physicians for their own storage and use. To view the contents of the CD-ROM, a physician would need only to insert it into the CD-ROM drive **117** (FIG. **2**) of a physician workstation **104** (FIG. **1**) and run a network browser program. By using the File command the physician could refer to the CD-ROM drive **117**, which would list the name of the master file **400**, which may be the same or similar to the patient's name.

FIG. **6A** sets forth an example of the contents of the master file **400**. Besides identifying the patient and the dates and source of the medical records, the master file has a series of hypertext links **402** either to distinct reports, such as hypertext link **408**, or to secondary control files, such as hypertext link **404**. FIG. **6B** sets forth how master file **400** (FIG. **6A**) might appear through a browser program when presented on display **118**. Note that hypertext links **402** are displayed in a different font format, as is the convention with browser programs. The system user may select a hypertext link by moving a pointing device such as a mouse over the text and pressing an activation button. The browser will automatically retrieve the file specified in the hypertext link **402** from the CD-ROM and present it.

If the system user selects the hypertext link **408** specifying discharge report in the master file **400**, the system user will be presented with the patient's single discharge report **412** (FIG. 7A). FIG. 7A sets forth some of the HTML codes which may be used to format discharge report **412**. Hypertext link **416** may be selected to retrieve the specified catheterization report from the CD-ROM. The hypertext link URL address has been modified as needed to make it compatible with the storage structure of the CD-ROM. FIG. 7B sets forth how discharge report **412** (FIG. 7A) might appear through a browser program when presented on display **118**.

FIG. 8A sets forth the contents of a secondary control file **418** that a browser program would present if the system user, while viewing master file **400** (FIG. 6A), selected the hypertext link **404** specifying cardiology data. Secondary control file **418** presents the system user with various types of cardiology reports to choose from in the form of hypertext links **422**. For those cardiology tests for which there is only one report available, the hypertext link specifies the URL address of that report. For those tests for which there are several reports available, the hypertext link **420** may specify the URL of a tertiary control file. FIG. 8B sets forth how secondary control file **418** (FIG. 8A) might appear through a browser program when presented on display **118**.

FIG. 9A sets forth the contents of tertiary control file **424** that a browser program would present if the system user, while viewing secondary control file **418** (FIG. 8B), selected the hypertext link **420** specifying electrocardiograph reports. This file presents the system user with a list of all the electrocardiograph reports, during the dates selected, to choose from in the form of hypertext links **426**. For each electrocardiograph report the hypertext link specifies the URL address of the report. FIG. 9B sets forth how tertiary control file **424** (FIG. 9A) and its list of electrocardiograph reports might appear through a browser program when presented on display **118**.

G. Parsing to Locate Data References

FIG. 15A illustrates how a data record is parsed. A data record is parsed to locate data references by searching it for text corresponding to a hypertext link or a multimedia data request. If one is found, the URL is located after the initial control sequence and will be saved (step **812**) for use after the parsing is completed. If none are found, or when the record has been completely parsed, another pass can be made to search for data references in the form of key words or key phrases (step **820**).

A key word or phrase is a recognized text string that is to be converted into a hypertext link. As an example, the data reference indicated by the phrase, "Admission ECG," can be converted (steps **828**, **830**) into the following hypertext link:

Admission ECG.

The expression "03may1997" is the date the data record being parsed was created. The patient ID (987654321), the date, and other descriptors are available from steps **200** and **226**, or from steps **544** or **560**. A wide variety of medical expressions can be recognized as key words or phrases, and appropriate hypertext links created from them. The URL of the hyperlink is saved for later use (step **832**). When the entire record has been searched (step **836**), the URLs of the located data references are returned to the section of the flow chart that requested the record to be parsed (step **840**).

H. Building Addresses Using Gleaned Information and Data References in Batch

It is contemplated that abbreviated database tables which includes at least some of the information included in database table **130**, file format instruction tables **134** (including hypertext ciphers, URL ciphers and special instructions), tables **150** and **158** and data request catalogue **500** can be used to "build" addresses of records referenced in a specific record during batch processing of the specific record (i.e. after the entire first record has been entered).

To this end, referring now to FIG. **21**, a first abbreviated database table **1800** preferably includes only data references (DRs) **1802** and associated URL's or address formats **1804**. The DRs **1802**, like the keywords described above, are searchable terms or phrases which are commonly used to refer to specific occurrences which may be associated with stored records. For example, "admission ecg" may be a DR **1802**. Other DRs may include "post-op x-ray", "PET image", a date, a patient's identification number, etc. The address formats **1804**, like the URL's described above, specify required information needed to form an address of a record associated with the corresponding DR and also specify the sequence of address fields which have to be filled with the required information to form the address.

Referring to FIG. **22**, the second abbreviated database table **1820** includes global instructions **1822** and a list of data types **1824** and corresponding record rule sets (RRSs) **1826**. The global instructions **1822** include rules for identifying the data type **1824** of a record which is being searched (e.g. to identify key words or phrases for creating links to other records on the system databases **106**, **2**) in batch form. Different data types **1824** are associated with different record information configurations. For example, one five field record may include a date in the second field while another may include the date in the fourth field. Yet a third record may include the date in the second field but may also include a total of seven fields. These three records would be characterized by three different data types **1824**, each different data type **1824** having a different information configuration.

The global instructions **1822** may take any form which, given the data types **1824** used with the wordprocessor **14**, can be used to identify a data type **1824**. For example, wordprocessor **14** may always provide a single field specifically reserved for a character or symbol which identifies the record data type **1824**. For instance, an ecg report may always be entered into a HIS **111** (see FIG. **1**) using a first data type template including specific fields for specific information (e.g. patient ID, date, time, physician, etc.). In this case, when an ecg data type template is opened to form an ecg record, wordprocessor **14** automatically enters a "1" in the data type field. Similarly, when a PET scan report is entered into the HIS, assuming PET scan reports are of a second record type, wordprocessor automatically enters a "2" in the data type field.

In the alternative, other global instructions **1822** may specify rules by which wordprocessor **14** can glean information entered by a user into record or template fields to determine the data type. For example, where a user enters a date into a fourth field instead of into a second field, wordprocessor **14** can distinguish a unique data type **1824** or at least a sub-group of data types. Similarly, wordprocessor **14** may recognize specific terms entered into certain fields to identify data type **1824**. For example, when a user enters "ecg" into a first field, wordprocessor **14** may recognize a specific data type based on global instructions **1822**. While only two different examples of global instruction have been described herein, the examples are not meant to be exhaustive and other examples are contemplated.

Referring still to FIG. **22**, the data types **1824** correspond to the different field configurations of the various record

which are stored on the in the system databases **106, 2**. The RRSs **1826** each include a set of rules for gleaning information from a record which is being searched (e.g. to identify key words or phrases for creating links to other records on the system databases **106, 2**). For example, where DT-1 **1828** indicates a first record type having five fields including a first patient ID field, a second date field, a third report type field, a fourth physician name field and a fifth text field, the corresponding RRS-1 **1830** includes rules which specify that to glean the patient ID, date, report type and physician name, wordprocessor **14** must access the first, second, third and fourth fields, respectively.

Referring now to FIG. **16**, a batch method performed by wordprocessor **14** (see also FIG. **1**) is illustrated. The method of FIG. **16** will be described in the context of an exemplary process. To this end, it will be assumed that a plurality of records have already been stored at database addresses according to the methods described above and in accordance with the specifications of the tables of FIGS. **21** and **22**. Thus, each of the stored records is associated with a DR **1802** and is stored at a database address which has a format indicated by an associated address format **1804** in the table of FIG. **21**. For example, a record associated with DR-3 **1808** is stored at an address having a format consistent with Format 3 **1810**. Records which have already been stored will be referred to generally as first records.

In addition, referring also to FIG. **17**, it will be assumed that the record illustrated therein (hereinafter a "second record") has been entered into the HIS and stored on one of the system databases **106, 2**. The record in FIG. **17** is a report created using wordprocessor **14**. The record includes DRs **1608** which reference a plurality of the first data records. Initially it is contemplated that DRs **1608** would not be highlighted but that, after wordprocessor **14** forms links between DRs **1608** and records corresponding to the DRs **1608**, the DRs **1608** would be highlighted via bolding or a different color text. In FIG. **17** the DRs **1608** include "admission ecg," "previous ecg," "previous discharge cath," and "admission CK enzyme" referencing various stored first records.

For each of the DRs **1608** (i.e. keywords) in FIG. **17**, wordprocessor **14** is capable of recognizing these DRs and correlating the DRs with address formats **1804** via table **1800**. In addition, wordprocessor **14** is also capable of determining the data type **1828** of the record shown in FIG. **17** and an associated RRS **1826** using the global instructions **1822** from table **1820**.

For the purposes of this explanation it will be assumed that the data type **1824** for the record illustrated in FIG. **17** includes five fields. In addition to a text field **1607**, the four other fields include a patient ID field **1600**, a date field **1602**, a report type field **1604** and an author field **1606**. It is also assumed that each of fields **1600, 1602, 1604, 1606** and **1607** already includes the information illustrated in FIG. **17**.

In addition, the record of FIG. **17** also includes a data type field **1610**. In the present example, it is assumed that, at an earlier time, when a physician accesses wordprocessor **14** to create a record, the physician indicates the data type to the wordprocessor in some manner. Data type may be indicated by selecting a data type from a list (e.g. ecg, PET report, post op X-ray, etc.) or it may be typed or it may be indicated in any other manner. When the physician indicates a data type **1824**, word processor **14** places a character or a character string in data type field **1610** indicating the data type of the record being created. In addition, wordprocessor **14** thereafter can provide fields to be filled in which are consistent with the specified data type **1824**. In this case, it is assumed that "DT-1" indicates an admission report having fields **1600, 1602, 1604, 1606, 1607** and **1610**.

With the second record completed as illustrated in FIG. **17** and stored on one of the system databases **106, 2**, a command is issued to wordprocessor **14** to search the second record to identify any references to first records which occur in the second record and, when a reference to a first record is located, to form a link between the reference and the referenced first record. To this end, wordprocessor **14** performs the process of FIG. **16**. Wordprocessor **14** separately receives each phrase in field **1607** (where the phrases include groupings of N or less consecutive words where N is the maximum number of consecutive words which may be included in a DR **1802**), and compares each phrase to DRs **1802** in table **1800** to identify DRs **1802**. Where a phrase does not match a DR **1802**, wordprocessor **14** jumps to the next entered phrase.

Referring to FIGS. **16** and **17**, when the phrase "admission ecg" is received as a second record segment at step **1500**, at step **1502** wordprocessor **14** accesses table **1800** and compares the phrase "admission ecg" to the DRs **1802** until either a match is identified or until the phrase has been compared to all of the DRs **1802**. In the alternative, table **1800** entries may be stored alphabetically and wordprocessor **14** may be equipped to recognize the first letter in a phrase. In this case, wordprocessor **14** may compare the phrase only to DRs **1802** which begin with the first letter of the phrase being compared to speed up the comparison process. At decision block **1504**, where the phrase does not match a DR **1802**, control passes back to processes step **1500** where the next record phrase or segment is received for comparison.

However, at block **1504**, where the phrase matches a DR **1802**, control passes to block **1508**. In the present example it will be assumed that DR-3 **1808** corresponds to the phrase"admission ecg". Thus, the phrase "admission ecg" matches a DR-3 **1808** and control passes to block **1508**. At block **1508** wordprocessor **14** uses table **1800** to identify address format (i.e. URL cipher) **1810** which corresponding to DRs **1808**. As indicated above, the address format specifies a format of an address associated with DR-3 **1808** and also specifies the required information needed to form the record address.

Next, at process block **1510**, wordprocessor **14** accesses the global instructions **1822** in table **1820** and uses the rules therein to determine the data type **1824** of the second record. In the present example, the global instructions **1822** instruct the wordprocessor to access data type field **1610** to identify the data type **1824**. Accessing field **1610**, wordprocessor **14** determines that the second record data type is DT-1. Accessing table **1824**, wordprocessor **14** correlates data type DT-1 **1828** with RRS-1 **1830**. As indicated above, RRS-1 **1830** specifies rules for how to glean the required information from the record illustrated in FIG. **17**. For example, address format **1810** may require, among other information, a patient ID number and a date used to locate reports for a particular patient related to the date. In the present example, RRS-1 **1830** may specify that the information in field **1600** corresponds to a patient ID and that the information in field **1602** corresponds to the current date.

Now using RRS-1 **1830**, at process block **1512**, wordprocessor **14** gleans the required information as specified by address format **1810** from the second record in the manner specified by RRS-1 **1830**. To this end, in the present example, wordprocessor **14** gleans the patient ID number and the date from fields **1600** and **1602**.

Next, at block **1512**, wordprocessor **14** forms an address for the record referenced by DR-3 **1808**. At block **1514** wordprocessor **14** automatically highlights the DR **1608** "admission ecg" in text field **1607** thus providing the DR "admission ecg" as a selectable segment. In addition, wordprocessor **14** links the DR "admission ecg" to the formed address such that, when the DR "admission ecg", is selected (e.g. via a mouse controlled cursor or the like), the wordprocessor **14** automatically accesses the record stored at the formed address and provides the record to a user for review.

In addition to information facially included in the second record, wordprocessor **14** may also glean other information associated with the second record for building a record address. For example, wordprocessor **14** may be associated with a specific facility and therefore may associate every record generated by the wordprocessor **14** with the specific facility (i.e. St. Mary's Springfield). In this case, when gleaning information, wordprocessor **14** may also glean information specifying the specific facility which is required to form an address. This information may be gleaned from any of a variety of locations including admit system **108**, hospital information system **111** or some other system linked to a hospital database **106, 2**.

This process of comparing record segments to DRs **1802** and forming links between DRs **1802** and records referenced by the DRs **1802** is continued on the second record text. In the present example links are formed between phrases (i.e. DRs **1802**) "previous discharge cath" and "admission CK enzyme" and records referenced thereby. For example, the address linked to the phrase "admission ecg" is identified by number **1700** while the address linked to the phrase "previous discharge ecg" is identified by number **1702**.

In addition, information in fields **1600** through **1606** may also be recognizable DRs **1802**. In this case wordprocessor **14** also forms links between information in those fields and corresponding records by first determining if the information in a field is a DR **1802**, identifying the address format **1804** corresponding to the DR **1802**, identifying required information for forming an address, identifying a data type **1824** corresponding to the record being searched, identifying the RRS **1826** associated with the data type **1824**, gleaning the required information as indicated by the RRS **1826**, forming an address for the record referenced by the DR **1802** by combining the required information, providing the DR **1802** as a selectable segment and linking the selectable segment to the record via the formed address. For example, the address linked to the patient ID number and associated patient demographics is identified by number **1704** and the address linked to the staff directory information for Dr. Markelson is identified by number **1706**.

Referring now to FIG. **18** therein is illustrated an HTML document corresponding to the document of FIG. **17** including hypertext addresses formed according to the process of FIG. **16**. It can be seen that each of the DRs **1608** of FIG. **17** now includes a linking address.

Referring also to FIG. **19**, the document of FIG. **18** is illustrated as the document would be viewed via wordprocessor **14** or a standard network browser (not illustrated). As illustrated, six selectable segments have been highlighted (i.e. underlined), a separate selectable segment for each recognized DR.

It should be noted that while the above-referenced batch processing to build record addresses has been described in the context of the wordprocessor **14**, some other data translation and collection system (e.g. **110** in FIG. **1**) may be provided to perform exactly the same functions in batch format.

It should also be noted that while specific fields may be provided in a record template for entering specific types of information, in another embodiment, a single field may be able to receive more than one type of information from which a wordprocessor or other type of device could glean the separate types of information for building a record address. For example, in one embodiment, a patient's ID number, a date, a time and perhaps other information may be provided in a single field, the wordprocessor **14** able to parse information in the single field to identify specific types of information. In this case, it is contemplated that there would have to be rules to avoid duplication of specific types of information in the single field. For example, there would have to be a rule that no more than one patient ID could be provided in the field or that, if more than a single patient ID were provided in the field, one of the ID numbers would have to be selected for generating an address.

In yet another embodiment of the present invention, referring to FIGS. **21, 22** and **23**, portions of the data in tables **21** and **22** are combined to form even a more abbreviated database table **1900** which can be used to form database addresses for DRs which reference a first record in a second record. To this end, table **1900** includes a data reference (DR) list **1902**, and both address formats **1904** and record rule sets (RRSs) **1906** which are correlated with the DRs **1902**. In this embodiment the DR list **1902** and address formats **1904** have the same purposes and forms as the list in FIG. **21**.

However, instead of being linked to a second record type, a separate RRS **1906** is linked to each of the unique DRs **1902**. In this case, each RRS **1906** includes a set of rules which, independent of a second record's data type, indicate how to glean required information for forming a record address form the second record information. For example, a particular medical facility may require a patient ID be identified as "ID:__" followed by a 9 digit number. In this case, the RRSs **1906** would specify that the term "ID:__" followed by a 9 digit number is a patient identification which can be used to populate a patient ID field in an address format. Similarly, the RRSs **1906** in this example include other rules which can be used to glean information from a second record for forming an address.

Referring to FIG. **16**, the method illustrated therein would be essentially the same using the table of FIG. **23** instead of the tables of FIGS. **21** and **22** and therefore, the method using table **1900** will not be explained here in detail. The only difference in the method of FIG. **16** when using table **1900** is that, at process block **1510**, the wordprocessor does not have to identify the data type and identifies the RRS by correlating the RRS with the DR in table **1900**.

I. Real Time Operation Of Address Building Using Ciphers

U.S. patent application Ser. No. 08/727,293 which was filed on Oct. 9, 1996 and is entitled "Method and System for Automated Data Storage and Retrieval with Uniform Addressing Scheme" is incorporated herein by reference. That application teaches a system whereby, as record information is entered into a record, a wordprocessor analyzes the information in real time to identify keywords, roots or data references (DRs) which are references to other records which are stored or which may be subsequently stored on one of the system databases **106, 2**. It is contemplated that the abbreviated database tables illustrated in FIGS. **21** and **22** can be used to "build" addresses of records referenced in a specific record as the record is entered into a wordproces-

sor database **2**. Thus, the process of forming record addresses described above in the context of batch processing (i.e. processing after the records have already been entered and stored) can be practiced in real time.

To this end, referring again to FIG. **16**, the wordprocessor method illustrated therein can also be performed in real time. To describe real time operation of the wordprocessor **14** to generate links between records, once again, it will be assumed that a plurality of records have already been stored at database addresses according to the methods described above and in accordance with the specifications of the tables of FIGS. **21** and **22**. Thus, each of the stored records is associated with a DR **1802** and a corresponding data type **1828** and address format 140 **1804**. Records which have already been stored will be referred to generally as first records. In addition, referring also to FIG. **17**, it will be assumed that the second record illustrated therein is being entered into the database **2** in real time.

For each of the DRs **1802** (i.e. keywords), wordprocessor **14** is capable of recognizing these DRs **1802** in the document illustrated in FIG. **17** and correlating the DRs **1802** with address formats **1804** via table **1800**. In addition, wordprocessor **14** is also capable of determining the data type **1828** of the record shown in FIG. **17** as that record is being input in real time and an associated RRS **1826** using the global instructions **1822** from table **1820**.

As in the batch example above, the record of FIG. **17** includes a data type field **1610**. in the present example, it is assumed that, when a physician initially accesses wordprocessor **14** to create the record of FIG. **17**, the physician indicates the data type to the wordprocessor in some manner. For example, data type may be indicated by selecting a data type from a list (e.g. ecg, PET report, post op X-ray, etc.). When the physician indicates a data type **1824**, word processor **14** places a character or a character string in data type field **1610** indicating the data type of the record being created. In addition, wordprocessor **14** thereafter can provide fields to be filled which are consistent with the specified data type **1824**. Once again, it is assumed that "DT-1" indicates an admission report having fields **1600**, **1602**, **1604**, **1606**, **1607** and **1610**.

After fields **1600** through **1606** are filled, the physician is prompted to enter report text into field **1607**. During text entry, wordprocessor **14** performs the process of FIG. **16**. To this end, processor **14** receives each phrase entered into field **1607** (where the phrases include groupings of N or less consecutive words where N is the maximum number of consecutive words which may be included in a DR **1802**), and compares each phrase to DRs **1802** in table **1800** to identify DRs **1802**. Where a phrase does not match a DR **1802**, wordprocessor **14** jumps to the next entered phrase.

Referring to FIGS. **16** and **17**, when the phrase "admission ecg" is received as a second record segment at step **1500**, at step **1502** wordprocessor **14** accesses table **1800** and compares the phrase "admission ecg" to the DRs **1802** until either a match is identified or until the phrase has been compared to all of the DRs **1802**. At decision block **1504**, where the phrase does not match a DR **1802**, control passes back to processes step **1500** where the next record phrase or segment is received for comparison.

However, at block **1504**, where the phrase matches a DR **1802**, control passes to block **1508**. In the present example it will again be assumed that DR-3 **1808** corresponds to the phrase "admission ecg". Thus, the phrase "admission ecg" matches a DR-3 **1808** and control passes to block **1508**. At block **1508** wordprocessor **14** uses table **1800** to identify address format (i.e. URL cipher) **1810** which corresponding to DRs **1808**. As indicated above, the address format specifies a format of an address associated with DR-3 **1808** and also specifies the required information needed to form the record address.

Next, at process block **1510**, wordprocessor **14** accesses the global instructions **1822** in table **1820** and uses the rules therein to determine the data type **1824** of the second record. In the present example, the global instructions **1822** instruct the wordprocessor to access data type field **1610** to identify the data type **1824**. Accessing field **1610**, wordprocessor **14** determines that the second record data type is DT-1. Accessing table **1824**, wordprocessor **14** correlates data type DT-1 **1828** with RRS-1 **1830**. As indicated above, RRS-1 **1830** specifies rules for how to glean the required information from the record illustrated in FIG. **17**. For example, address format **1810** may require, among other information, a patient ID number and a date used to locate reports for a particular patient related to the date. In the present example, RRS-1 **1830** may specify that the information in field **1600** corresponds to a patient ID and that the information in field **1602** corresponds to the current date.

Now using RRS-1 **1830**, at process block **1512**, wordprocessor **14** gleans the required information as specified by address format **1810** from the second record in the manner specified by RRS-1 **1830**. To this end, in the present example, wordprocessor **14** gleans the patient ID number and the date from fields **1600** and **1602**.

Next, at block **1512**, wordprocessor **14** forms an address for the record referenced by DR-3 **1808**. At block **1514** wordprocessor **14** automatically highlights the DR **1608** "admission ecg" in text field **1607** thus providing the DR admission ecg" as a selectable segment. In addition, wordprocessor **14** links the DR "admission ecg" to the formed address such that, when the DR "admission ecg" is selected (e.g. via a mouse controlled cursor or the like), the wordprocessor **14** automatically accesses the record stored at the formed address and provides the record to a user for review.

This process of comparing record segments to DRs **1802** and forming links between DRs **1802** and records referenced by the DRs **1802** is continued as text is entered into the record text field **1607**. In the present example links are formed between phrases (i.e. DRs **1802**) "previous discharge cath" and "admission CK enzyme" and records referenced thereby.

In addition, as in the batch example above, information in fields **1600** through **1606** may also be recognizable DRs **1802**. Here, the only difference in wordprocessor **14** operation is that the DRs in fields **1600–1606** are recognized as the fields are filled and, if all of the required information is not yet entered for building an address, the wordprocessor **14** waits to build the address until all of the required information has been provided. For example, referring still to FIGS. **17** and **21**, where the patient ID field **1600** is recognized as a DR **1802** and the date in field **1602** is required to form an address to a record corresponding to the patient and the date, if the date has not yet been entered, the wordprocessor **14** must wait for the date to be entered prior to forming the address. In the alternative, the wordprocessor **14** may be programmed to wait until all fields **1600–1606** are filled prior to building any addresses for DRs (i.e. processing of information in the preliminary, non-text fields may be done in batch).

It should be understood that while the real time addressing method described above is described in the context of the tables of FIGS. **21** and **22**, the table of FIG. **23** may be used

instead, the only differences being that in step **1510** word-processor **14** does not identify the second record data type and that the wordprocessor **14** determines the RRS by correlating an RRS **1906** in table **1900** with an identified DR **1902**.

While a particular embodiment of the invention has been illustrated and described, it will be obvious to those skilled in the art that various changes and modifications may be made without sacrificing the advantages provided by the principle of construction disclosed herein. For example, while the present invention has been described above in the context of an interface wherein a physician enters record information via a keyboard, clearly the invention is not to be so limited. For example, data may be dictated into a system, the inventive wordprocessor including voice recognition software and identifying DRS as they are dictated and forming links between the DRS and records referenced thereby.

In the alternative, referring to FIG. **20**, the system may be used with a data specifying device such as a hand held information gathering device (HHD) **1200** which downloads record information via a receiver (e.g. rf) **1202** to a word-processor **14**, wordprocessor **14** generating address links either during the downloading process or in batch form after information transfer. One such HHD is described in U.S. Patent application Ser. No. 09/170,169 which is entitled "Data Collection Device and System", which is commonly owned with the present application and which is incorporated herein by reference. In this type of system, the HHD may in fact specify a DR which can be used by processor to identify a DB address in any of several different ways. For example, processor **14** may simply correlate the DR with a DB address. In the alternative, processor **14** may build an address using the DR and an associated address format. In addition to specifying the DR, it is also contemplated that the HHD specifies other information for forming the DB address (e.g. background or general data set information such as patient ID, time, date, etc.). Also, it should be recognized that any data specifying device may be used such as a database which indicates a DR or the like. To this end, see the DB **106** linked to processor **14** via a network router **1206** or other network device (i.e. a receiver) in FIG. **20**.

Furthermore, it should be recognized that the present invention may also be used in the case of a proxy server which routes address commands and requests among network devices. For example, the invention may build record addresses which do not actually match record addresses on a database but which, when provided to a proxy server, enable the proxy server to identify the actual record addresses and form a link.

To apprize the public of the scope of this invention, we make the following claims.

What is claimed is:

1. A method to be used with a processor linked to at least one database, the at least one database including at least one second record, a first record including at least one first record phrases, at least one first record phrase associated with a keyword phrase which references the at least one second record, the method for rendering second records accessible from within the first record and comprising the steps of:

receiving at least one first record phrase;

determining if the first record phrase is associated with a keyword phrase;

if the first record phrase is associated with a keyword phrase, identifying the at least one second record which is referenced by the keyword phrase; and

rendering the identified record accessible.

2. The method of claim **1** wherein the processor is linked to an interface and wherein the step of receiving includes receiving an indication via the interface that the processor should attempt to recognize a first record phrase as a keyword phrase.

3. The method of claim **2** wherein the indicated first record phrase corresponds to more than one keyword phrase, the step of determining includes identifying each keyword phrase corresponding to the first record phrase and the step of identifying at least one second record includes identifying a separate second record for each identified keyword phrase.

4. The method of claim **3** wherein the step of identifying each keyword phrase corresponding to the first record phrase includes the step of comparing the first record phrase to a list of defined keyword phrases and, when the referencing phrase includes a threshold number of characters in a defined keyword phrase, identifying the defined keyword phrase as a keyword phrase.

5. The method of claim **3** wherein the interface includes a display, the step of rendering records accessible includes displaying the keyword phrases on the display in a selectable format and facilitating selection of at least one of the displayed keyword phrases.

6. The method of claim **2** wherein the interface includes a display and the step of receiving an indication includes the step of receiving an indication via the display which identifies the first record phrase.

7. The method of claim **2** wherein the step of receiving an indication includes the step of receiving a leading character which designates a first record phrase.

8. The method of claim **7** wherein the processor facilitates entry of first record text and the method further includes the steps of, as first record text is entered, monitoring the text for the leading character and, after a leading character is recognized, identifying the phrase thereafter as a first record phrase.

9. The method of claim **2** wherein the processor facilitates entry of first record text and wherein the step of receiving includes, as first record text is entered, receiving each newly entered phrase in the first record as a first record phrase.

10. The method of claim **1** wherein the keyword phrase is a keyword.

11. The method of claim **1** wherein the step of determining includes comparing first record phrases to keyword phrases.

12. The method of claim **1** wherein the processor is linked to a display, the step of rendering includes visually presenting the keyword phrase in a format which is distinguishable from other first record phrases and linking the record associated therewith to the presented keyword phrase such that, when the presented keyword phrase is selected, the linked record is accessed.

13. The method of claim **1** wherein the step of identifying the at least one second record includes the step of identifying an address for the at least one second record.

14. The method of claim **13** wherein the step of identifying an address includes forming an address for the at least one second record.

15. The method of claim **1** wherein the first record is associated with other information which characterizes the keyword phrase and the step of identifying the at least one second record further includes the steps of identifying other information which further characterizes the keyword phrase and, based on the keyword phrase and other information, identifying the at least one second record associated therewith.

16. The method of claim **15** wherein the other information is within the first record text.

17. The method of claim 15 wherein the first record is stored at a first record address and the other information is associated with the first record address.

18. The method of claim 15 wherein the processor facilitates entry of first record text and the method is performed as first record text is entered.

19. The method of claim 15 wherein the keyword is a second keyword and the other information is a first keyword.

20. The method of claim 12 wherein the processor is also linked to an interface which facilitates modification of first record phrases and, wherein the method further includes the steps of, monitoring modifications to the first record phrases and, when a first record phrase associated with a second record is modified such that the first record phrase no longer corresponds to a keyword phrase, de-linking the keyword phrase and the previously associated second record.

21. The method of claim 2 wherein the interface includes at least one input device and the step of receiving includes receiving an indication via the input device.

22. A method to be used with a processor linked to at least one database, the database including at least one second record, a first record including phrases, at least one first record phrase associated with a keyword phrase which references the at least one second record, the method for indicating that the first record phrase in the first record references another record and comprising the steps of:

identifying at least one phrase in a first record which references a second record; and

indicating the identified first record phrase in the first record.

23. The method of claim 22 wherein each second record is associated with at least one keyword phrase and the processor accesses a list of defined keyword phrases and the step of identifying includes determining if a phrases in the first record is associated with a defined keyword phrase and, if a phrase in the first record is associated with a defined keyword phrase, identifying the phrase as a first record phrase.

24. The method of claim 22 wherein the processor is also linked to an interface which facilitates modification of first record text including modification of indicated first record phrases and, when an indicated first record phrase is modified, the method further includes the step of eliminating indication of the indicated first record phrase in the first record, first record for instances of each keyword phrase in the list.

25. The method of claim 22 wherein the step of indicating includes the step of highlighting the first record phrase in the first record.

26. The method of claim 22 wherein the processor is linked to a display for displaying at least the first record and wherein the step of indicating includes highlighting the first record phrase within the first record on the display.

27. The method of claim 23 wherein the processor is linked to a display for displaying at least the first record and wherein the step of indicating includes the step of providing the keyword phrase in a separate window on the display.

28. The method of claim 22 wherein the processor facilitates entry of first record text and the step of identifying includes, as text is entered, determining if newly entered characters are associated with second records.

29. The method of claim 28 wherein the step of determining includes monitoring entered text a leading character which designates a phrase and, after a leading character is recognized, determining if the characters thereafter are associated with a second record.

30. The method of claim 22 for indicating a plurality of first record phrases in a first record which reference second records and wherein the steps of identifying and indicating are performed for each first record phrase which references a second record.

31. The method of claim 22 further including the steps of, after identifying, forming a link between the indicated first record phrase and the referenced record.

32. The method of claim 22 wherein the step of indicating includes the step of rendering the first record phrase selectable, the method further including the steps of, after rendering the first record phrase selectable, monitoring for selection of the first record phrase and, when the first record phrase is selected, providing the second record.

33. An apparatus to be used with a processor linked to at least one database, the at least one database including at least one second record, a first record including at least one first record phrases, at least one first record phrase associated with a keyword phrase which references the at least one second record, the apparatus for rendering second records accessible from within the first record and comprising:

a processor running a pulse sequencing program to perform the steps of:

receiving at least one first record phrase;

determining if the first record phrase is associated with a keyword phrase;

if the first record phrase is associated with a keyword phrase, identifying the at least one second record which is referenced by the keyword phrase; and

rendering the identified record accessible.

34. The apparatus of claim 33 wherein the processor is linked to an interface and wherein the processor performs the step of receiving by receiving an indication via the interface that the processor should attempt to recognize a first record phrase as a keyword phrase and wherein the identified record is rendered accessible via the interface.

35. The apparatus of claim 34 wherein the indicated first record phrase corresponds to more than one keyword phrase, the processor performs the step of determining by identifying each keyword phrase corresponding to the first record phrase and performs the step of identifying at least one second record by identifying a separate second record for each identified keyword phrase.

36. The apparatus of claim 35 wherein the processor performs the step of identifying each keyword phrase corresponding to the first record phrase by comparing the first record phrase to a list of defined keyword phrases and, when the referencing phrase includes a threshold number of characters in a defined keyword phrase, identifying the defined keyword phrase as a keyword phrase.

37. The apparatus of claim 35 wherein the interface includes a display and the processor performs the step of rendering records accessible by displaying the keyword phrases on the display in a selectable format and facilitating selection of at least one of the displayed keyword phrases.

38. The apparatus of claim 34 wherein the interface includes a display and the processor performs the step of receiving an indication by receiving an indication via the display which identifies the first record phrase.

39. The apparatus of claim 34 wherein the processor performs the step of receiving an indication by receiving a leading character which designates a first record phrase.

40. The apparatus of claim 39 wherein the processor facilitates entry of first record text and the processor further runs the pulse sequencing program to perform the steps of, as first record text is entered, monitoring the text for the leading character and, after a leading character is recognized, identifying the phrase thereafter as a first record phrase.

41. The apparatus of claim 34 wherein the processor facilitates entry of first record text and wherein the processor performs the step of receiving by, as first record text is entered, receiving each newly entered phrase in the first record as a first record phrase.

42. The apparatus of claim 33 wherein the keyword phrase is a keyword.

43. The apparatus of claim 33 wherein the processor performs the step of determining by comparing first record phrases to keyword phrases.

44. The apparatus of claim 33 wherein the processor is linked to a display, the processor perform the step of rendering by visually presenting the keyword phrase in a format which is distinguishable from other first record phrases and linking the record associated therewith to the presented keyword phrase such that, when the presented keyword phrase is selected, the linked record is accessed.

45. The apparatus of claim 33 wherein the processor performs the step of identifying the at least one second record by identifying an address for the at least one second record.

46. The apparatus of claim 45 wherein the processor performs the step of identifying an address by forming an address for the at least one second record.

47. The apparatus of claim 33 wherein the first record is associated with other information which characterizes the keyword phrase and the processor performs the step of identifying the at least one second record by further causing the program to perform the steps of identifying other information which further characterizes the keyword phrase and, based on the keyword phrase and other information, identifying the at least one second record associated therewith.

48. The apparatus of claim 47 wherein the other information is within the first record text.

49. The apparatus of claim 47 wherein the first record is stored at a first record address and the other information is associated with the first record address.

50. The apparatus of claim 47 wherein the processor facilitates entry of first record text and the processor performs the steps as first record text is entered.

51. The method of claim 47 wherein the keyword is a second keyword and the other information is a first keyword.

52. The apparatus of claim 44 wherein the processor is also linked to an interface which facilitates modification of first record phrases and, wherein the processor runs the program to further perform the steps of, monitoring modifications to the first record phrases and, when a first record phrase associated with a second record is modified such that the first record phrase no longer corresponds to a keyword phrase, de-linking the keyword phrase and the previously associated second record.

53. The apparatus of claim 44 wherein the interface includes at least one input device and the processor performs the step of receiving by receiving an indication via the input device.

54. An apparatus to be used with a processor linked to at least one database, the database including at least one second record, a first record including phrases, at least one first record phrase associated with a keyword phrase which references the at least one second record, the apparatus for indicating that the first record phrase in the first record references another record and comprising:

a processor running a pulse sequencing program to perform the steps of:

identifying at least one phrase in a first record which references a second record; and

indicating the identified first record phrase in the first record.

55. The apparatus of claim 54 wherein each second record is associated with at least one keyword phrase and the processor accesses a list of defined keyword phrases and the processor performs the step of identifying by determining if a phrases in the first record is associated with a defined keyword phrase and, if a phrase in the first record is associated with a defined keyword phrase, identifying the phrase as a first record phrase.

56. The apparatus of claim 54 wherein the processor is also linked to an interface which facilitates modification of first record text including modification of indicated first record phrases and, when an indicated first record phrase is modified, the processor further causes the program to perform the step of eliminating indication of the indicated first record phrase in the first record.

57. The apparatus of claim 54 wherein the processor performs the step of indicating by highlighting the first record phrase in the first record.

58. The apparatus of claim 54 wherein the processor is linked to a display for displaying at least the first record and wherein the processor performs the step of indicating by highlighting the first record phrase within the first record on the display.

59. The apparatus of claim 55 wherein the processor is linked to a display for displaying at least the first record and wherein the processor performs the step of indicating by the step of providing the keyword phrase in a separate window on the display.

60. The apparatus of claim 54 wherein the processor facilitates entry of first record text and the processor performs the step of identifying by, as text is entered, determining if newly entered characters are associated with second records.

61. The apparatus of claim 60 wherein the processor performs the step of determining by monitoring entered text for a leading character which designates a phrase and, after a leading character is recognized, determining if the characters thereafter are associated with a second record.

62. The apparatus of claim 54 for indicating a plurality of first record phrases in a first record which reference second records and wherein the processor performs the steps of identifying and indicating for each first record phrase which references a second record.

63. The apparatus of claim 54 wherein the processor further runs the program to perform the steps of, after identifying, forming a link between the indicated first record phrase and the referenced record.

64. The apparatus of claim 54 wherein the processor performs the step of indicating by rendering the first record phrase selectable, the processor further running the program to perform the steps of, after rendering the first record phrase selectable, monitoring for selection of the first record phrase and, when the first record phrase is selected, providing the second record.

* * * * *

jc525 U.S. PTO 09/512125 02/24/00

ISSUE CLASSIFICATION	Class	Subclass
	707	3

PATENT NUMBER
6308171

6308171

U.S. UTILITY Patent Application

O.I.P.E.

SCANNED AA Q.A. Re

PATENT DATE
OCT 2 3 2001

APPLICATION NO.	CONT/PRIOR	CLASS	SUBCLASS	ART UNIT	EXAMINER
09/512125	D	707	3	~~2771~~ 2171	COLT, FRANTZ

APPLICANTS: Carlos De La Huerga

(CIP See 08/871,818)

TITLE: Method and system for automated data storage and retrieval

PTO-2040 12/99

ISSUING CLASSIFICATION

ORIGINAL		CROSS REFERENCE(S)						
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)					
707	3	707	1					
INTERNATIONAL CLASSIFICATION		705	2	3				
G06F 17/30								
G06F 17/60								

☐ Continued on Issue Slip Inside File Jacket

TERMINAL DISCLAIMER	DRAWINGS			CLAIMS ALLOWED	
☑ 09/512,125	Sheets Drwg. 30	Figs. Drwg. 43	Print Fig. 15A	Total Claims 64	Print Claim for O.G. 1
☐ The term of this patent subsequent to ______ (date) has been disclaimed.	Frantz Coby (Assistant Examiner)		04/02/01 (Date)	NOTICE OF ALLOWANCE MAILED 4-9-01	
☑ The term of this patent shall not extend beyond the expiration date of U.S Patent. No. 5,845,461	THOMAS BLACK SUPERVISORY PATENT EXAMINER TECHNOLOGY CENTER 2100 (Primary Examiner)		4/5/01 (Date)	ISSUE FEE Amount Due $1240.00	Date Paid 06-01-01
☐ The terminal ____ months of this patent have been disclaimed.	SChapman (Legal Instruments Examiner)		(Date)	ISSUE BATCH NUMBER E69	

WARNING:
The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

Form PTO-436A
(Rev. 6/99)

FILED WITH: ☐ DISK (CRF) ☐ FICHE ☐ CD-ROM
(Attached in pocket on right inside flap)

Formal Drawings ___ sheets set

(FACE)

PATENT APPLICATION
09512125

09/512125
02/24/00

INITIALS 3/13/00 12

CONTENTS

		Date Received (Incl. C. of M.) or Date Mailed			Date Received (Incl. C. of M.) or Date Mailed
1.	Application #30 papers.		42.		
2.	Rejection (3 months)	1-18-01	43.		
3.	Terminal Disclaimer	3-16-01	44.		
4.	Allowance	4-9-01 M.L.	45.		
5.	Formal Drawings (30 shts) set 1	6/4/01	46.		
6.			47.		
7.			48.		
8.			49.		
9.			50.		
10.			51.		
11.			52.		
12.			53.		
13.			54.		
14.			55.		
15.			56.		
16.			57.		
17.			58.		
18.			59.		
19.			60.		
20.			61.		
21.			62.		
22.			63.		
23.			64.		
24.			65.		
25.			66.		
26.			67.		
27.			68.		
28.			69.		
29.			70.		
30.			71.		
31.			72.		
32.			73.		
33.			74.		
34.			75.		
35.			76.		
36.			77.		
37.			78.		
38.			79.		
39.			80.		
40.			81.		
41.			82.		

(LEFT OUTSIDE)

POSITION	INITIALS	ID NO.	DATE
FEE DETERMINATION	SD	71058	3-3-00
O.I.P.E. CLASSIFIER		43	3/13/08
FORMALITY REVIEW	DW	72346	4-19-08
RESPONSE FORMALITY REVIEW			

INDEX OF CLAIMS

✔ ..... Rejected
= ..... Allowed
— (Through numeral)... Canceled
÷ ..... Restricted
N ..... Non-elected
I ..... Interference
A ..... Appeal
O ..... Objected

Claim Final	Claim Original	Date 01/09/08
1	(1)	✔
2	2	✔
3	3	✔
4	4	✔
5	5	✔
6	6	✔
7	7	✔
8	8	✔
9	9	✔
10	10	✔
11	11	✔
12	12	✔
13	13	✔
14	14	✔
15	15	✔
16	16	✔
17	17	✔
18	18	✔
19	19	✔
20	20	✔
21	21	✔
22	(22)	✔
23	23	✔
24	24	✔
25	25	✔
26	26	✔
27	27	✔
28	28	✔
29	29	✔
30	30	✔
31	31	✔
32	32	✔
33	(33)	✔
34	34	✔
35	35	✔
36	36	✔
37	37	✔
38	38	✔
39	39	✔
40	40	✔
41	41	✔
42	42	✔
43	43	✔
44	44	✔
45	45	✔
46	46	✔
47	47	✔
48	48	✔
49	49	✔
50	50	✔

Claim Final	Claim Original	Date 01/09/08
51	51	✔
52	52	✔
53	53	✔
54	(54)	✔
55	55	✔
56	56	✔
57	57	✔
58	58	✔
59	59	✔
60	60	✔
61	61	✔
62	62	✔
63	63	✔
64	64	✔
	65	
	66	
	67	
	68	
	69	
	70	
	71	
	72	
	73	
	74	
	75	
	76	
	77	
	78	
	79	
	80	
	81	
	82	
	83	
	84	
	85	
	86	
	87	
	88	
	89	
	90	
	91	
	92	
	93	
	94	
	95	
	96	
	97	
	98	
	99	
	100	

Claim Final	Claim Original	Date
	101	
	102	
	103	
	104	
	105	
	106	
	107	
	108	
	109	
	110	
	111	
	112	
	113	
	114	
	115	
	116	
	117	
	118	
	119	
	120	
	121	
	122	
	123	
	124	
	125	
	126	
	127	
	128	
	129	
	130	
	131	
	132	
	133	
	134	
	135	
	136	
	137	
	138	
	139	
	140	
	141	
	142	
	143	
	144	
	145	
	146	
	147	
	148	
	149	
	150	

If more than 150 claims or 10 actions staple additional sheet here

(LEFT INSIDE)

SEARCHED

Class	Sub.	Date	Exmr.
707	1	01/08/01	FC
707	3	01/08/01	FC
705	2	01/08/01	FC
705	3	01/08/01	FC
707	1	04/02/01	FC
707	3	04/02/01	FC
705	2	04/02/01	FC
705	3	04/02/01	FC

INTERFERENCE SEARCHED

Class	Sub.	Date	Exmr.
707	1	04/02/01	FC
707	3	04/02/01	FC
705	2	04/02/01	FC
705	3	04/02/01	FC

SEARCH NOTES
(INCLUDING SEARCH STRATEGY)

	Date	Exmr.
WEST (search)	01/08/01	FC
WEST (search)	01/05/01	FC
IEEE (online)	01/09/01	FC
ACM (NPL)	01/09/01	FC
WEST (search)	04/02/01	FC
IEEE	04/02/01	FC
ACM (NPL)	04/02/01	FC

(RIGHT OUTSIDE)

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

03/03/2000 SDAVIS 00000019 170055 09512125

01 FC:101 690.00 CH
02 FC:102 78.00 CH
03 FC:103 792.00 CH

PTO-1556
(5/87)

*U.S. GPO: 1999-459-062/19144

02-28-00 A

PTO/SB/05 (1/98)
Approved for use through 9/30/00, OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Please type a plus sign (+) inside this box +

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

jc571 U.S. PTO 02/24/00

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No,	250591.90163
First Inventor or Application Identifier	Carlos De La Huerga
Title	METHOD AND SYSTEM FOR AUTOMATED DATA STORAGE AND RETRIEVAL
Express Mail Label No,	EL 636 882 900 US

jc625 U.S. PTO 09/512115 02/24/00

APPLICATION ELEMENTS

See MPEP Chapter 600 concerning utility patent application contents.

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, D.C. 20231

1 [X] Fee transmittal Form
(Submit an original and a duplicate for fee processing)

2 [X] Specification [Total Pages 44]
(preferred arrangement set forth below)
- Descriptive title of the invention
- Cross References to Related Applications
- Statement Regarding Fed Sponsored R&D
- Reference to Microfiche Appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings *(if filed)*
- Detailed Description
- Claim(s)
- Abstract of the Disclosure

3 [X] Drawing(s) (35 USC 113) [Total Sheets 30]

4. Oath or Declaration [Total Pages 2]
a. [X] Newly executed (original or copy)
b. [ ] Copy from prior Application (37 CFR 1.63(d)) (for continuation/divisional with Box 17 completed)
[Note Box 5 below]
i. [ ] DELETION OF INVENTOR(S) Signed Statement attached deleting inventor(s) named in prior application, see 37 CFR 1.63(d)(2) and 1.33(b).

5 [ ] Incorporation By Reference (useable if Box 4b is checked) The entire disclosure of the prior application from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference herein.

6. [ ] Microfiche Computer Program *(Appendix)*

7. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
a. [ ] Computer readable Copy
b. [ ] Paper Copy (identical to computer copy)
c. [ ] Statement Verifying identity of above

ACCOMPANYING APPLICATION PARTS

8 [ ] Assignment Papers (cover sheet & documents)
9 [ ] 37 CFR 3.73(b) Statement (where there is an assignee) [ ] Power of Attorney
10 [ ] English Translation Document *(if applicable)*
11 [ ] Information Disclosure Statement (IDS)/PTO-1449 [ ] Copies of IDS Citations
12 [ ] Preliminary Amendment
13 [X] Return receipt postcard (MPEP 503) *(Should be specifically itemized)*
14 [ ] *Small Entity Statement(s) [ ] Statement filed in prior application Status still proper and desired
15 [ ] Certified copy of priority Document(s) *(if foreign priority is claimed)*
16 [ ] Other: Unsigned Declaration

* A new statement is required to pay small entity fees, except where one has been filed in a prior application and is being relied upon

17. If a CONTINUING APPLICATION, *check appropriate box and supply the requisite information:*
[ ] Continuation [ ] Divisional [X] Continuation-in-part (CIP) of prior application no. 09/247,349
Prior application information: Examiner: Group/Art Unit: 2771

18. CORRESPONDENCE ADDRESS

[ ] Customer Number or Bar Code Label [redacted] or [X] Correspondence address below

NAME	Michael A. Jaskolski				
ADDRESS	Quarles and Brady LLP				
	411 East Wisconsin Avenue				
CITY	Milwaukee	STATE	WI	ZIP CODE	53202
COUNTRY	USA	TELEPHONE	414/277-5711	FAX	414/271-3552

Name (Print/Type)	Michael A. Jaskolski	Registration No. (Attorney/Agent)	37,551
Signature	M. Jaskolski	Date	2/24/00

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

METHOD AND SYSTEM FOR AUTOMATED DATA STORAGE AND RETRIEVAL

CROSS-REFERENCE TO RELATED APPLICATIONS

This patent application is a continuation-in-part of U.S. Patent Application Number 09/247,349 which was filed on February 10, 1999, which is entitled "Method and System for Automated Data Storage and Retrieval" and which is a continuation-in-part of U.S. Patent Application Number 08/727,293 which was filed on October 9, 1996, is entitled "Method and System for Automated Data Storage and Retrieval with Uniform Addressing Scheme", which issued as U.S. Patent Number 5,895,461 which claimed priority from provisional application serial number 60/023,126 which was filed on July 30, 1996. Application Number 09/247,349 was is also a continuation-in-part of U.S. Patent Application Number 08/871,818 which was filed on June 9, 1997, is entitled "System and Method for Translating, Collecting and Archiving Patient Records" and which issued as U.S. Patent Number 5,903,889 and there through the present invention is also a continuation-in-part of U.S. Patent Number 5,903,889.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not applicable.

BACKGROUND OF THE INVENTION

The present invention relates to the collection, storage, and retrieval of data on computer systems. More particularly, the present invention relates to a computer system for retrieving, modifying, and storing a plurality of topically, textually, or audio-visually related data records of a plurality of formats on a plurality of databases in conformance with a hypertext-linked, predefined topical organization.

When a patient is in a hospital, either as an inpatient or an outpatient, a variety of information concerning the patient may be collected and recorded. This may be in the form of observations, measurements, lab results, vital sign indicators, procedure reports and associated graphics. Over a long period of treatment, hundreds of pages of information may accumulate in the patient's record.

While the patient is in the hospital, it is typical that many different care givers, administrators, or insurance company employees will desire to view a part of the patient's cumulative record. The conventional paper chart is not always useful, as there is only one copy of it, and some laboratory tests may not be entered into the chart on a timely basis. To solve this problem, hospitals have used a variety of database systems such as hospital information systems (HIS) and clinical information systems (CIS) to store and present patient information on computer displays. However, there is still a substantial amount of data that

does not get placed into these systems. A variety of factors may inhibit an automated process of comprehensive retrieval of a patient's data, such as incompatible communication protocols and formatting schemes between computer systems, non-digitized data records including pictures and standardized forms, and the lack of adequate computer interfacing support for low-cost medical instruments or devices. It is also typical that word processing documents, rather than being automatically collected by a database system, are simply printed in the form of a paper copy to be inserted into the conventional chart.

While various standardization committees have been established, e.g., HL-7, DOCOM, and IEEE, to develop common addressing schemes for hospital data, to date none have defined a consistent format to use for storing and retrieving data. For the sake of simplicity or due to limited resources, many manufacturers that use one or more of these standards choose to use only a portion of them; consequently their systems remain only partially compatible.

Furthermore, even many hospitals with database systems lack a centralized retrieval system because related hospital reports are often stored on separate databases. For example, a patient's radiology catheterization report and hemodynamic catheterization reports may be created and stored in separate databases, though as far as the physician who performed the catheterization procedure is concerned, these two reports are really just one procedure and should be associated with each other. For further example, a physician reviewing an admission report may find that it references laboratory tests or observations made contemporaneous with or previous to the patient arriving at the hospital. Should the physician decide to review these other records, she will have to perform additional searches to locate them. In some cases, this often cumbersome and time-consuming process results in care givers refraining from making complete use of the available patient information.

In many hospitals when a patient is discharged, a paper copy of these records is made and sent to the admitting physician for his own record keeping purposes. The collection, copying, and storage of all of these records is a very time-consuming and labor-intensive activity. Further, the generally high risk of human error may manifest itself in the failure to return records to the correct patient's file or incorrect storage of a patient's entire file, effectively forfeiting the misplaced information. The physician is simultaneously confronted with the responsibility of filing and storing the paper copy in his own office.

Some hospitals have purchased laboratory or information systems capable of long term storage of various records. While this may assist the hospital in retrieving past records, it may not help the admitting physician in referring to them, for he may not have access to the data directly or may not have the specific software required to retrieve the data. So with such advanced systems the physician is still provided with a paper copy for his records.

Furthermore, many existing laboratory and information systems record information in a variety of inconsistent formats. Some of these formats are proprietary to the manufacturer of the specific system. Each system may use a separate database scheme to gain access to the data. Substantial efforts to get these systems to communicate with each other have not

3

yielded satisfactory results. For example, many large medical information systems use complicated data exchange protocols; but these protocols are unwieldy for simple, often portable instruments which lack the hardware and software capacity to conform to such protocols.

Some reports may be created using a wordprocessor. These may originate in a department of the hospital or in a physician's office. These reports, which may be kept in a conventional file cabinet, are not always included with the rest of the patient's reports.

What is needed is an effective alternative to creating paper records that must be copied and meticulously tracked, an alternative that would permit physicians to access the data economically and easily in their own offices. Such a system would permit a system user to enter a keyword to retrieve a specific data record of a patient, retrieve the requested record from whichever database it is stored to, reformat the data record with hypertext links to related patient records, and return the requested record to the system user for display on a browser. The system would preferably use a mark-up language such as the well-known Hypertext Markup Language (HTML) or JAVA so that it could utilize inexpensive, standard software packages. The system would also be operable to format data records stored on the various databases of the computer network systematically, periodically, or automatically upon the creation of new, or the modification of existing, data records. The system would be operable to collect all data records pertaining to a specific patient, doctor, or other subject, modify them to support display through a Java applet, Internet browser, or other universal display standard, generate additional patient files to organize the data records in a hypertext directory structure, and store the data records and files on a mass-media storage device such as a CD-ROM.

In addition, it would be advantageous to have a system which permits both the storage and the retrieval of data records according to a standardized addressing scheme which can be determined solely by the use of certain keywords known to the various users of the system. Such a system could employ standard word processing software to enable multiple users to create and reference the various data records. The system would recognize certain keywords entered by the user during creation of the data record and use those keywords to determine the appropriate location (e.g., database, directory and file name) to store the record according to a predetermined addressing scheme. Similarly, it is desirable for the users of the system to be able to locate particular data records using a few keywords without having to know the complexities of which database the record is on, the format of the record, the file name or the directory address.

BRIEF SUMMARY OF THE INVENTION

It is an object of this invention to provide a means of processing and converting existing data records formatted, structured, and accessed according to a multitude of disparate standards to common standards by which they may be accessed, controlled, and/or displayed through a single interactive display program. It is another object of this invention to

provide conventions for exploring data records for references to contextually related data records and modifying, generating, embedding, and appending links and data-retrieving codes in and to said related data records, whereby to organize said related data records in a hypertext tree structure. A further object of this invention is to store a group of related data records organized in a hypertext tree structure to a mass storage device, such as a hard disk or CD-ROM, through which the data records may be retrieved, displayed, and controlled through a single interactive display program. In order to minimize costs and maximize end-user accessibility, the standards and conventions used by the present invention for modifying data records and addressing schemes facilitate display through a widely-familiar and low-cost display program such as an Internet browser.

The invention may be adapted for use in a wide variety of applications, and is suitable for any environment in which numerous data records having one or multiple forms and/or formats are to be collected, stored, archived, retrieved, or translated. By way of illustration and not by way of limitation, the invention is presented in the context of a hospital environment, in which typically there are numerous computer systems in use by various health care professionals in one or several hospitals, and each professional often desires to have access to the patient records created by other professionals in that or other hospitals.

A typical setting for the present invention provides multiple databases and workstations linked via a network wherein the databases store data records in a variety of formats and the workstations utilize user interfaces to input, retrieve, and manipulate data records. The present invention utilizes specification tables identifying each of the information processors or databases used by the hospital, the types of data records stored by the databases, and instructions and algorithms for accessing, modifying, and processing data records and their addresses, depending on the data type. Similar specification tables are also kept to identify each workstation where wordprocessor, spreadsheet, or other records, including those downloaded from portable medical devices, may be held.

When a system user at a workstation linked to the hospital computer network equipped with the present invention submits a request for a particular patient record, the invention parses the data request for an address root and other pertinent information about the data record to be retrieved, which may include the time and date the data record was created or last modified and a patient ID. Using this information incorporated in the data request and in the specification tables, the invention modifies the existing data request into a URL or other addressing convention, as necessary, to retrieve the data record from the appropriate database.

After retrieving the data record, the invention may modify it to make it compatible with a standard supported by the common interactive display browser used by the system. For example, the invention may convert a text document to an HTML document or convert graphics, video, or audio records to browser or Java-enabled formats. Further, depending on the formatting specifications for a particular data type, the invention may identify key words, links, and programming codes embedded in the data record, modifying them and inserting

additional hypertext links and programming codes as necessary. For example, it may be desirable that a hypertext link referencing the patient's demographics and insurance information be inserted into each record reporting on the patient's condition, status, or profile for quick and easy referral. As another example, it may be desirable to place a hypertext link in a radiology catheterization report that references the hemodynamic catheterization report and vice versa so that each refers to the other.

In this manner a hospital may use Internet or Intranet compatible databases with databases that are not compatible, and may choose to use URL addresses of its own choice independent of what the individual vendors have chosen. The administrator may also preprogram the data translation and collection system to link reports together as appropriate, so that care givers may more quickly and directly refer to relevant or related information. The translation process described here may be used on a dedicated system for this purpose or may be distributed among several processors including those of the database systems.

Another aspect of the present invention includes means for receiving, processing, and storing hospital records systematically, periodically, or automatically as they are created or modified. In this mode of operation data records may be pre-formatted according to the hospital's specifications, allowing for quicker record retrieval during subsequent data requests. The translation operation may be allocated to a dedicated system for this purpose or may be distributed among several processors, including those of the database systems.

A further aspect of the present invention includes means for periodically retrieving and filing the contents of a designated area of each workstation's disk. For example, word processing documents generated at a workstation may be stored in a designated area, such as a special "collection" drive or folder, to which the hospital computer network has access. The invention would retrieve the data records stored in the collection folder, and identify, interpret, and modify them before storing them in an appropriate database.

Yet another aspect of the present invention includes means for retrieving, processing, and storing all of a patient's data records that are available on the hospital's computer network onto a mass media storage device, such as a CD-ROM. For example, this process may be initiated by submitting a collection request identifying the patient's ID number or other identifier uniquely identifying the patient. The invention submits requests, passwords, macros, and programming codes, as appropriate, to each of the databases and workstations that include portions of the patient's cumulative record. Each record retrieved is processed and modified as above—as if the particular record had been requested by a system user. The invention not only collects applicable data records, but also multimedia clips, applets, browser extensions, "plug-ins," and other application modules addressed by programming codes embedded in the patient's data records. Substitute files explaining the absence of a linked record or module are created for data records or modules regarded as inappropriate for storage and distribution on an unsecured or uncontrolled medium. The invention would also create a "master file for the patient analogous to a "home page" for a website or the root directory of a tree structure, containing links to other patient-related files and data records.

The master file may have hypertext links to patient records and to additional (secondary) control files, which in turn have hyperlinks to more patient data. After completing these collection routines, the invention would transfer the collection of data records, applets, browser extensions, and other data and programming modules to a mass-storage device. In this manner a patient's cumulative patient record could be stored on a single CD-ROM or other high-density storage device, cheaply distributed to other hospitals or health care professionals serving the patient, and be conveniently accessed by those hospitals and health care professionals.

The present invention also provides a wordprocessor employing a standardized addressing scheme, wherein the wordprocessor recognizes keywords entered by a user who is either creating and storing a data record or attempting to locate a data record among numerous data records at different addresses on a plurality of computer databases.

Thus, the present invention provides first a plurality of databases on which a variety of data records are stored. The databases are in communication with one or more processors which interpret input data from a user interface and direct the storage and retrieval of data records. The databases and processors may be linked via a network, or one or more of the databases may communicate locally with an associated processor, as in a personal computer. The invention also provides a plurality of user interfaces, such as combinations of keyboards, video displays, microphones with voice recognition, and other input devices (e.g. rf receiver, etc.), through which system users create, store, retrieve and display data records. These user interfaces can be simple terminals which communicate with a processor and a database over the network, or they can be part of an integrated interface/processor combination, such as in a personal computer.

For accepting keywords from the user and determining the storage location of a data record to be stored or retrieved, the invention includes a wordprocessor having certain defined functions. For the creation of data records, the wordprocessor accepts various information from the user to identify the user and the type of record being created, as well as other information which may uniquely identify the record and its storage location after the record is completed and saved, or "published." The wordprocessor uses these keywords, or specialized information fields, to determine the location at which the record is to be stored and employs a standardized addressing scheme compatible with or comparable to the Universal Resource Locator (URL) addressing used on the global computer network (Internet). The wordprocessor automatically creates a link between the keyword in the data record and the address of the data record on the computer system.

The wordprocessor also includes a function which compares text entered by the user to a predetermined list of keywords known to be used in the system and may prompt the user for a different keyword when no match is found. Once the user enters a sufficient number of recognized keywords to uniquely identify the data record being created or sought, the wordprocessor determines the unique address of the data record according to a

7

predetermined, standardized addressing scheme so that the record may be stored or retrieved.

When a data record is created and stored, the wordprocessor creates a link, in the manner of a hypertext link, between a keyword uniquely identifying the particular record and its unique address (URL) on the computer system. To this end, the wordprocessor identifies background information within the record and uses the background information and, in some cases, the keyword itself, to form the address link pointing to the record. This link points to the unique address of the record and will enable other users to retrieve the record when the same keyword is used in a request for a data record. In the same manner, other data records containing this same keyword will contain a link to that record, permitting users to create data records which refer to other data records by use of a hypertext link.

The wordprocessor included in the invention contains a monitoring function which monitors free text entered by the user to determine whether the user is creating a hypertext reference at a place in the data record other than in a specified keyword field. This monitoring function continuously surveys text/data being input by the user so that hypertext links in a data record or report can be created by the user at will.

The wordprocessor also includes an editing function which permits keywords, or hypertext references, in data records to be edited and determines whether a user is changing the keyword to another keyword or a non-keyword. This editing function attempts to match changed keywords with other known keywords to determine whether the user is referencing a different data record. The wordprocessor treats keywords or keyword phrases as singularities which cannot be edited without either deleting the link (URL or hyperlink) associated with the keyword or changing the hypertext link to a different hypertext link.

The addressing scheme and hypertext links of the invention are suitable to be created by and used with conventional tools in common use for publishing documents on the Internet. Data records containing keywords and hypertext links may be created in a markup language such as Hypertext Markup Language (HTML), and the addressing scheme may comport with Internet URL addressing. Thus, the invention provides Internet/Intranet capabilities and may be operated with relatively inexpensive, commercially available HTML formatting software and Internet browser software. Other hypertext link preparation methods and other addressing schemes are possible.

The invention further includes a system for linking first record references to a first record wherein the references are in a second record, the system comprising a database (DB) including at least one address format specifying an address format of the first record address and a processor linked to the DB and running a pulse sequencing program to perform the steps of receiving the second record, analyzing the second record to identify references to the first record and when a first record reference is identified, using information from the second record to form the address of the first record as specified by the address format.

Preferably the address format also specifies required information for forming the address for the first record, the DB further includes at least one record rule set (RRS)

corresponding to the address format, the RRS specifying rules for gleaning the required information from a record and, wherein, when the first record is referenced in the second record, the processor gleans the required information from the second record in the manner specified by the RRS.

Also, preferably, the DB also includes a data reference (DR) which is associated with the address format and wherein, when searching for a reference to the first record, the processor searches for instances of the DR. In one aspect the program includes a wordprocessor, the DR is a text name associated with the first record and the first record address is a markup language data reference.

In one embodiment the system is also for creating markup language data references between the first record references and the first record, the processor also, when a first record reference is identified, provides the first record reference to a user as a selectable segment and links the selectable segment to the first record via the first record address such that, when the selectable segment is selected, the first record is provided to the user.

Where a selectable segment is provided, the processor may further, after the selectable segment is provided, perform the steps of, when the second record is accessed, monitor changes to the second record and, when the selectable segment is modified, de-linking the selectable segment and the first record.

In another aspect the DB includes a plurality of address formats and their associated RRSs and DRs. In this case, the processor searches the second record for any of the DRs and, when any of the DRs is identified, the processor identifies the associated address format and RRS, gleans the required information from the second record in the manner specified by the associated RRS and forms the address corresponding to the first record.

In yet another aspect the system includes an interface for entering the second record, the second record entered in record segments and, wherein, the processor runs the program as second record segments are entered via the interface.

In another aspect the system is for use with a data specifying device wherein the step of receiving includes receiving the second record from the data specifying device.

Furthermore, the invention includes a system which receives database records, each record including a separate information set and characterized by at least one data type, for a specific record, the system using the specific record's information set to construct a record address which enables easy subsequent record access, the system comprising a database (DB) including at least one address format which is associated with the at least one record type and which specifies a unique set of required information to form a record address for the record type. The system also including a processor linked to the DB and running a pulse sequencing program to perform the steps of, for the specific record receiving the information set, confirming the data type and the associated address format, analyzing the information set to glean the required information, using the required information to form a record address as specified by the address format and storing the record at the record address.

In one embodiment there are a plurality of data types, the DB includes a separate address format for each of the different data types and the step of confirming includes the steps of determining the data type and the associated address format.

In one embodiment the DB further includes at least a separate record rule set (RRS) corresponding to each of the address formats, each RRS specifying a unique set of rules for gleaning required information from a record and, wherein, the processor gleans required information in the manner specified by the RRS.

Preferably the program is a first application program and the processor also performs a second application program to link stored records which are referenced in a first record to the referenced records, to this end the processor further performing the steps of, after at least one record is stored, searching the first record for a reference to a stored record, when a reference to a stored record is identified, determining the address associated with the referenced record, providing the reference to a user as a selectable segment and linking the selectable segment to the referenced stored record via the record address such that, when the selectable segment is selected, the record is provided to the user.

Preferably, the processor provides a data reference (DR) for the record information set, the DR useable to refer to the record in other records, the processor, when searching for a reference in the first record, searching for the DR.

Also, preferably, after the record address is formed, the processor also correlates the DR with the record address and stores the DR along with the record address, the processor determining the address associated with a reference by identifying the address associated with an identified DR.

In one embodiment the RRS is a first RRS and the DB also associates a second RRS with the address format, the second RRS specifying rules for gleaning the required information from the first record, when a DR is gleaned from an information set, the processor also correlating the DR with the address format and storing the DR along with the address format, the processor determining the address associated with a reference by, when a DR is identified, identifying the address format associated with the DR, identifying the second RRS associated with the identified address format and the required information specified by the identified address format, gleaning the required information from the first record as specified by the second RRS and forming the record address using the required information and as specified by the address format.

The invention further includes a system which receives database records, each record including a separate information set and characterized by at least one data type, for a specific record, the system using the specific record's information set to identify a record address which enables easy subsequent record access. The system is also for use with a data specifying device which provides the database records, including at least one field specifying a data reference. The system comprises a receiver for receiving records from the specifying device and a processor linked to the receiver and running a pulse sequencing program to perform the steps of, for a specific record, receiving the information set, identifying

the DR, using the DR to identify a record address for the record and storing the record at the record address.

The specifying device may be a hand held device or a database or some other suitable specifying device (e.g. a dictaphone). In addition, the specifying device, in addition to specifying the DR, may also specify other information which is used to identify the address.

According to yet another embodiment of the invention, a record being searched may be characterized by a data type and the data type may be associated with a specific RRS for gleaning information therefrom and, when a DR is identified in a record, the system may determine the data type of the searched record to identify the RRS to be used to glean the required information.

Moreover, the invention further includes methods to be used in conjunction with the systems described above.

These and other objects, advantages and aspects of the invention will become apparent from the following description. In the description, reference is made to the accompanying drawings which form a part hereof, and in which there is shown a preferred embodiment of the invention. Such embodiment does not necessarily represent the full scope of the invention and reference is made therefor, to the claims herein for interpreting the scope of the invention.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

Fig. 1 is a block diagram of a medical computer network according to the present invention, including a plurality of databases for data record storage and a data translation and collection system;

Fig. 2 is a graphical illustration of a physician office workstation;

Figs. 3A and 3B are tables showing the contents of the "Database Table" and "File Format Instruction Table" maintained and used by the data translation and collection system to access, translate, reformat, and store data records kept on databases on the medical computer network;

Figs. 4A and 4B are tables showing the contents of "List of Workstations" and "Report List" maintained and used by the data translation and collection system to access, translate, reformat, and store data records kept on workstations on the medical computer network;

Figs. 5A-5F are functional flow charts showing the steps used to collect and process a related set of data records from various databases, create a structured set of control files containing hypertext links to the collected data records, and store the data records and control files to a storage device;

Fig. 6A is a graphical representation of a master file in HTML format through which all of a single patient's medical records created at a hospital equipped with the present invention may be viewed;

Fig. 6B is a graphical representation of the master file of Fig. 6A as viewed by a system user with a network browser;

Fig. 7A is a graphical representation of a secondary control file in HTML format providing a hypertext-link embedded discharge report;

Fig. 7B is a graphical representation of the secondary control file of Fig. 7A as viewed by a system user with a network browser;

Fig. 8A is a graphical representation of another secondary control file in HTML format providing a structured list of hypertext links to a plurality of cardiology reports;

Fig. 8B is a graphical representation of the secondary control file of Fig. 8A as viewed by a system user with a network browser;

Fig. 9A is a graphical representation of a tertiary control file in HTML format providing a list of electrocardiogram reports;

Fig. 9B is a graphical representation of the tertiary control file of Fig. 9A as viewed by a system user with a network browser;

Fig. 10 is a table showing the contents of the "Data Request Catalogue" maintained and used by the data translation and collection system to discriminate the data type and database location of a requested data record from the alphanumeric string requesting the data;

Fig. 11 is a graphical representation of a data request containing an address root and descriptor;

Figs. 12A-12C are a functional flow chart showing the steps used to receive a request for a data record, translate the request, retrieve the data record, and reformat the data record prior to sending it to its requested destination;

Figs. 13A-13C are a functional flow chart showing the steps by which the data translation and collection system processes a data record received or retrieved from a workstation or database system on the medical computer network, reformat the data record, assign it a URL address, and deliver it to a database for storage;

Figs. 14A-14B are textual representations of a URL address as received and reformatted by the data translation and collection system;

Fig. 14C is a graphical representation of a report referenced by the URL address of Fig. 14B as it would be viewed by a system user through a network browser;

Fig. 14D is a textual representation of the report of Fig. 14C as modified to include data references in the form of HTML codes;

Fig. 14E is a graphical representation of the modified report in Fig. 14D as it would be viewed by a system user through a network browser;

Figs. 15A-15B are a functional flow chart showing the steps by which a data record is parsed to locate data references within it;

Fig. 16 is a flow chart illustrating a method of forming or building record addresses using a record rule set and an address format in real time according to the present invention;

Fig. 17 is a is a graphical representation of a sample patient report during its creation by a user of the computer system according to the invention;

Fig. 18 is a graphical representation of text of the report of Fig. 17 after being converted to HTML format and having hypertext links to URL addresses substituted for their associated data references;

Fig. 19 is a graphical representation of the report of Fig. 17 with hypertext links, as viewed by a system user with a network browser or other request handler routine;

Fig. 20 is a schematic of specifying devices and a processor of the present invention;

Fig. 21 is a table used by the wordprocessor of the present invention to build data base addresses for linking first record references in a second record to the first record;

Fig. 22 is a table used in conjunction with the table of Fig. 21 for gleaning information from a second record which is required to form a data base address for a first record which is referenced in a second record; and

Fig. 23 is a table similar to the tables of Figs. 21 and 22 which is used in another embodiment of the present invention to build data base addresses for linking first record references in a second record to the first record.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to Fig. 1, the invention is illustrated as a medical computer network 100, including a plurality of hospital based workstations 102 (which may be personal computers), a plurality of physician office workstations 104 which may also be personal computers, a plurality of databases 106 which may be provided by a multitude of vendors with separate data structures and data elements. The computer network 100 may also comprise an Admit, Discharge, and Transfer (ADT) system 108, a data translation and collection system 110, and a Hospital Information System (HIS) 111. The data translation and collection system 110 is not necessarily a separate physical element of the medical computer network 100, but is represented that way in the preferred embodiment for purposes of illustration only. It may be alternately recognized as a program application or even an aspect of a network operating system, the operations of which may be distributed over and performed by many different processors, workstations, and databases on the medical computer network 100.

Databases 106, computer systems 108, 110, 111, workstations 102, and physician office workstations 104 may communicate with each other via a communication network 112, which may be a combination of local and wide area networks, using Ethernet, serial line, wireless, or other communication standards. Communication network 112 may also be arranged in such a manner to be part of the Internet or as an individual Intranet. Each workstation 102, 104 includes a "collection" folder 105, a user interface 103 which may include a network browser or similar display, entry, and retrieval program, a separate database 2 and a special wordprocessor 14. User interface 103 may be any means for permitting users to create data records and/or retrieve data records from the medical

computer network 100 capable of supporting a network browser, such as well known keyboard and video terminal combinations or voice recognition hardware and software.

Wordprocessor 14 runs under the direction of user interface/processor 103 and governs the creation of data records, the recognition of keywords in the data records, the composition of hypertext links between the keywords and the data records, and the retrieval of data records in response to keywords input to the wordprocessor by the user or contained within a data record.

Fig. 2 shows a typical physician office workstation 104 comprising a personal computer 107 which may include a display 118, and a CD-ROM drive 117 or other means of mass storage which may be removable.

The data translation and collection system 110 maintains a file referred to as Database Table 130, whose contents are partially seen in Fig. 3A. For each database 106 included on the medical computer network 100, an entry is made in the Database Register 131 of the Database Table 130. Corresponding to each entry in the Database Register 131 is an address or addresses field 132 used to access the database on communication network 112 and a separate File Format Instruction Table 134.

A partial list of the contents of File Format Instruction Table 134 is seen in Fig. 3B, which includes records of each data type 136 stored by the database 106. Corresponding to each data type 136 in File Format Instruction Table 134 is a set of special instructions or program codes 142 used to translate a request for such data to a format appropriate to the data type and database from which the requested information may be retrieved. Also corresponding to each data type 136 is a hypertext cipher or record rule set (RRS) 138 providing special instructions or codes used to add data references (such as hypertext links) and to format the data, which instructions or codes may include decompression algorithms. In addition, the RRS 138 also specifies rules for gleaning information from a record which can be used to form a record address. Hereinafter the terms hypertext cipher and RRS may be used interchangeably to mean the same thing.

Further corresponding to each data type 136 is a URL cipher or address format 140 used to generate an address to store the designated type of data. The address format 140 specifies information which is required to form an address for an associated record type and also specifies the order of the information in the resulting address. Hereinafter the terms address format and URL cipher may be used interchangeably to mean the same thing.

The data translation and collection system 110 may also retain a file referred to as Data Request Catalogue or database table (DBT) 500, whose contents are partially seen in Fig. 10. The Data Request Catalogue or DBT 500 includes an array of Data Request Address Roots or data references (DRs) 504, to each element of which corresponds fields identifying the data type 136, the database 106 in which the data type is located, and hypertext cipher 138 (which is kept also in the File Format Instruction Tables 134 (Fig. 3B) of the Database Table 130 (Fig. 3A)). This file may be accessed when a request for data is received by the

data translation and collection system 110 to recognize the matching data request address root 504 which identifies the data type 136 and the database 106 on which it is kept.

A. Responding to Data Requests and Providing Translation

Figs. 12A-12C describe the operation of the data translation and collection system 110 (Fig. 1) in responding to requests to retrieve data, translating those requests to conform to the format required by the applicable database, retrieving the data, reformatting the data, and delivering the data to the appropriate destination.

Commencing with Fig. 12A, in step 540 the data translation and collection system 110 receives a data record reference 520 (Fig. 11) in the form of a data request containing an address root 522 and descriptors 524 about the requested record. In some instances, a data request will originate from a system user accessing a hypertext link on a document displayed by the system's interactive display browser. In other instances, the data request will originate from a database or workstation application program. There may be several non-uniform but mutually distinguishable data request formats among the several hospital databases 106 (Fig. 1) on the medical computer network 100 (Fig. 1). Alternately, data requests may be uniformly and compatibly formatted for all records stored by various hospital databases 106 (Fig. 1). For example, the data requests may be in the form of a URL with optional data fields sent with it to assist in identifying the record to be retrieved.

In step 544, the address root 522 (Fig. 11) of the data record reference 520 may be determined by removing the descriptors 524—any patient identification, chronological details, or other non-addressing information—from the received data request. The descriptors 524 are temporarily stored for use in step 560.

In step 548, a search is performed to locate a match for the address root 522 (Fig. 11) of the data record reference 520 among the data request address roots 504 (Fig. 10) listed in the data request catalogue 500 (Fig. 10). By finding a matching data request root address 504, the invention immediately identifies the data type 136 (Fig. 10), the database 106 used to store this data, and the hypertext cipher 138 providing special instructions used to format and add data references to the data.

In step 552 the database 106 identified in step 548 is in turn referenced in Database Table 130 (Fig. 3A) to its corresponding File Format Instruction Table 134 (Fig. 3A) to determine the address(es) 132 (Fig. 3A) of the database 106 storing the data.

In step 556 the data type 136 identified in step 548 is cross-referenced with the File Format Instruction Table 134 (Fig. 3A) identified in step 552 to locate the special instructions to request data 142 (Fig. 3B) used to translate the request to a format appropriate to the data type and database from which the requested information may be retrieved. These instructions may, for example, include passwords or macros needed to retrieve the data.

In step 560 a code is constructed using the database address(es) 132 identified in step 552, the special instructions to request data 142 identified in step 556, and the

descriptors 524—the patient identification, chronological details, or other non-addressing information—stored in step 540. The code is submitted to the appropriate database to produce the requested data record.

After the database has produced the requested data record, the record may in step 564 be received by the data translation and collection system 110 for additional processing.

The steps by which the data translation and collection system 110 processes the selected data record are shown in Figs. 12B and 12C. In step 566, the system uses the hypertext cipher 138 to determine whether or not the data is stored in a proprietary format. If it is, the applicable proprietary software is used to decompress or translate the data. This may be done on the manufacturer's database 106, another computer processing system, or by the data translation and collection system 110 itself.

In step 570, the record is parsed, discussed infra, to locate date references: hypertext links, multi-media requests, and key words or phrases. If none are found, the process advances to step 598, discussed infra.

If there are data references, they may in steps 580 through 596 be reformatted so that the URL addresses are compatible with addressing protocols used by the hospital. In step 580, the address root 522 (Fig. 11) of the hypertext link or other data record reference 520 may be determined by removing the descriptors 524—any patient identification, chronological details, or other non-addressing information—from the received data request. The descriptors 524 are temporarily stored for use in step 596. In step 584, a match for address root 522 is sought among the data request address roots 504 (Fig. 10) listed in Data Request Catalogue 500, which locates the Database 106 and Data Type 136 corresponding to the matching Data Request Address Root 504. In step 592, the identified Database 106 and Data Type 136 are referenced in Database Table 130 (Fig. 3A) and the corresponding File Format Instruction Table 134 (Fig. 3B) to acquire the appropriate URL cipher 140 (Fig. 3B). In step 596, the URL cipher 140 processes the descriptors 524—the patient identification, chronological detail, and other information—extracted in step 580 to modify or replace the hypertext link or other data reference found in the selected record. Steps 580 through 596 may be performed for each hypertext link and reference to other data records found in the selected record.

For some types of data records, the URL cipher 140 will generate addresses compatible with database formatting standards such as SQL or Oracle.

In step 598 the data translation and collection system 110, using the Hypertext Cipher 138, converts any text portion of the selected data record into a browser compatible format, such as HTML format, and any graphics, audio, video, or other non-text information into a browser, plug-in, or Java compatible format.

In step 600, the data translation and collection system 110 inserts hypertext links or other references to the selected record in accordance with the hypertext cipher 138 identified in step 548. If directed by the hypertext cipher 138, the record may also be interpreted and modified or reformatted.

In step 604, the data translation and collection system 110, having retrieved and translated the requested record, forwards the record to the destination selected by the requesting workstation or processor.

B. Receiving Patient Records for Translation and Address Formatting

Figs. 13A-13C set forth an alternate embodiment of the operation of the data translation and collection system 110 (Fig. 1) with particular reference to receiving, translating, and formatting data records to facilitate access through browsers and hypertext links for future users. This embodiment is similar to that set forth in Figs. 12A-12C but may proceed independently of and prior to a request for such data. Thus in this embodiment the data translation and collection system 110 may serve to organize and format a patient's records prior to their being requested by a member of the medical staff.

Commencing with Fig. 13A, in step 640 the data translation and collection system 110 receives a data record from a database 106 which may include or be appended to other information specifying patient identification, chronological detail, the data type, and other information regarding the record. In step 644, the data translation and collection system searches each File Format Instruction Table 134 (Fig. 3A) corresponding to each entry in Database Register 131 of Database Table 130 to locate the data type 136 of the received data record. In step 648, the hypertext cipher 138 (Fig. 3B) of File Format Instruction Table 134 may be used to parse the record to identify additional information, such as patient information and the date and time of the record.

In step 650, the system uses the hypertext cipher 138 (Fig. 3B) to determine whether or not the data is stored in a proprietary format. If it is, the applicable proprietary software is used to decompress or translate the data. This may be done on the manufacturer's database 106, another computer processing system, or by the data translation and collection system 110 itself.

In step 654, the record is parsed, discussed infra, to locate data references: hypertext links, multi-media requests, and key words or phrases. If none are found, the process advances to step 682, discussed infra.

If hypertext links or references to other data records are found, they may in steps 664 through 680 be reformatted so that the URL addresses are compatible with addressing protocols used by the hospital. In step 664, the root address 522 of the data record reference 520—which may be in the form of a hypertext link—is extracted as in step 544 (Fig. 12A). Similarly, any descriptors 524—such as patient identification, chronological detail, or other non-addressing information—contained in the data record reference 520 is also extracted and temporarily stored. In step 668, a match for this address root is sought among the Data Request Address Roots 504 (Fig. 10) listed in Data Request Catalogue 500 (Fig. 10), which locates the Database 106 (Fig. 10) and Data Type 136 (Fig. 10) corresponding to the

matching Data Request Address Root 504. In step 672, the data type 136 and Database 106 identified in step 668 are cross-referenced with their corresponding File Format Instruction Table 134 (Fig. 3A) to locate the special instructions to request data 142 (Fig. 3B). In step 676, the identified Database 106 and Data Type 136 are referenced in Database Table 130 (Fig. 3A) and the corresponding File Format Instruction Table 134 (Fig. 3B) to acquire the appropriate URL cipher 140 (Fig. 3B). In step 680, the URL cipher 140 processes the descriptors 524—the patient identification, chronological detail, and other information—extracted in step 664 to modify or replace the hypertext link or other data reference found in the received data record. Steps 664 through 680 may be performed for each hypertext link and reference to other data records found in the received data record.

In step 682 the data translation and collection system 110, using the Hypertext Cipher 138, converts any text portion of the selected data record into a browser compatible format, such as HTML format, and any graphics, audio, video, or other non-text information into a browser, plug-in, or Java compatible format.

In step 684, the data translation and collection system 110 inserts hypertext links or other references to the received data record in accordance with the hypertext cipher 138 (Fig. 3B) identified in step 548. If directed by the hypertext cipher 138, the record may also be interpreted and modified or reformatted. In step 686, the URL cipher 140 corresponding to the data type 136 (Fig. 3B) of the received data record processes the descriptors 524—the patient identification, chronological detail, and other information stored or extracted in steps 640 or 648—to format a URL by which the received data record may be accessed.

In step 688, the data translation and collection system 110, having translated and formatted the received data record, forwards the record and its formatted URL to an appropriate database 106 for storage.

C. Operation of the Hypertext and URL Ciphers of the Data Translation and Collection System

Figs. 14A-14D set forth an example of the hypertext and URL processing performed by the data translation and collection system 110 (Fig. 1) in response to a request for a data record. Fig. 14A proffers, by way of example, a URL address 700 that may be consistent with a standard hospital format, that is received by the data translation and collection system 110. Embedded in this URL address 700 is information regarding the type of data 704, the patient's identification 708, the date 712 and time 716 of the data requested, and a report designator 718. The type of data 704, combined with additional information, is an example of an address root 701 and the information referred to as 708, 712, 716 and 720 are examples of descriptors 702. The data translation and collection system 110, by following steps 544 through 560 as set forth in Fig. 12A, reformats the data request into a new data request 720 (Fig. 14B), which is compatible with the database system 106 (Fig. 1) holding this data.

Fig. 14C sets forth an example of a report 724 that may be produced by a database 106 in response to the data request 720 of Fig. 14B. Initially, the report is only a conventional text document. The data translation and collection system 110 (Fig. 1) may then convert the report into an HTML-compatible format 732 (Fig. 14D), inserting data request 736 and hypertext links 740 and 744 according to the hypertext cipher 138 (Fig. 3B). The hypertext links 744 may be inserted based upon the recognition of phrases or special character sequences, such as "Catheterization Reports" 728, in the report, which may vary from report to report of the same data type depending on the each report's contents.

Fig. 14E shows the text report 724 with imported image 737 as displayed on computer display 118 using a network browser software package after the report has been translated and modified. A system user seeking additional information regarding the patient's demographics could select hypertext link 740. A system user seeking either the radiology or hemodynamic report for this procedure could select the appropriate hypertext link 744.

D. Data Collection and Translation for Storage and Retrieval

FIGS 5A-5E set forth a second alternate embodiment of the operation of the data translation and collection system 110 (Fig. 1) with particular reference to the steps used by the data translation and collection system 110 to retrieve and format data to produce a complete, organized, hypertext-linked, and browser-compatible collection of records pertaining to a person, place, thing, or event. This operation may be initiated by a system user executing the appropriate command or may be executed routinely and automatically by the hospital's Admit, Discharge, and Transfer (ADT) system or Hospital Information System (HIS) during a patient's stay or when a patient is discharged.

In step 200, the data translation and collection system 110 (Fig. 1) receives a patient identification number, which may originate from a staffed workstation 102 (Fig. 1) or automatically from the ADT system 108 (Fig. 1) or HIS 111 (Fig. 1). This may be done, for example, when a patient is admitted or after one has been discharged. In step 204, the data translation and collection system 110 may request the dates for which the system user desires to collect data for the patient or the most recent admission dates from the ADT system 108 or HIS 111. In step 208, a file containing a list of received records will be opened, if previously created, or generated, if not previously created. Similarly, in step 212, a file containing a list of records to be retrieved is opened, if previously created, or generated, if not previously created. In step 216, the data translation and collection system 110 references database table 130 (Fig. 3A) to locate and retrieve the first database entry in the database register 131 (Fig. 3A), the corresponding File Format Instruction table 134 (Figs. 3A and 3B) and the first data type 136 (Fig. 3B) in the File Format Instruction table 134.

Steps 220 through 276 set forth an iterative search of all the databases 106 (Fig. 1) on the medical computer network 100 (Fig. 1) to collect, translate, and format all records relevant to a patient's medical history.

(9)

In step 220, the field containing special instructions to request data 142 (Fig. 3B) corresponding to the data type 136 (Fig. 3B) being referenced by the data translation and collection system 110 is used to construct and format a message which is sent, using address 132 (Fig. 3A), to the database 106 being referenced in database register 131 (Fig. 3A). This message may incorporate passwords, macros, or other codes, as necessary, to retrieve the data. In step 224 a data record is retrieved.

In step 226, the record is parsed per the Hypertext Cipher 138 (Fig. 3B) in the File Format Instruction Table 134 (Fig. 3B) to derive the date and time of the record. In step 228 the file name is added to the list of received records opened in step 208. In step 232, the list of records to retrieve opened in step 212 is checked for a reference to the retrieved record. If a reference exists, in step 236 it is removed from the list of records to retrieve.

In step 240, the system uses the Hypertext Cipher 138 (Fig. 3B) to determine whether or not the data is stored in a proprietary format. If it is, the applicable proprietary software is used to decompress or translate the data. This may be done on the manufacturer's database 106, another computer processing system, or by the data translation and collection system 110 itself.

In step 246, the record is parsed, discussed infra, to locate data references: hypertext links, multi-media requests, and key words or phrases. If none are found, the process advances to step 266, discussed infra. If there are data references, a check is made to determine if the data being referenced had been located previously (step 254). If it had not been previously located, the record is added to the List of Records to Retrieve (step 258). In ste 262, all hypertext links and other data requests are reformatted through use of the URL Ciphers 140, maintained by the data translation and collection system 110 for each Data Type 136. This is done in a manner similar to steps 580 to 596, discussed supra. Thus when the retrieved data record is later displayed, secondary files referenced by it will be included for display and the system user will not be presented an incomplete record.

In step 266, whether there were data references or not, Hypertext Cipher 138 is used to convert text to HTML format, graphics, audio, video, or other non-text information into a browser, plug-in, or Java compatible format.

While not shown in the flow chart of Figs. 5A-5E, if the data translation and collection system 110 retrieves a record that includes a program code module such as a Java applet, the data translation and collection system will attempt to retrieve a copy of the applet from an address specified by the applet program code, generate a new address for the applet copy which will be stored with the patient's data record collection, and modify the program code module to reflect the new address. Similarly, if the data translation and collection system 110 retrieves a record that requires a browser extension or "plug-in" in order to be properly viewed, a copy of the applicable extension or "plug-in" is also retrieved for storage with the patient's data record collection.

For purposes of privacy or security, the medical computer network 100 may deny access to some data records in the list of records to be retrieved. In such instances a

substitute file, indicating that the requested file is confidential or has not been included, is created and stored, and its reference substituted for the reference to the confidential data record.

In step 268, a retrieved data record may be further modified, such as by inserting additional hypertext links or data requests to the record per the hypertext cipher 138 (Fig. 3B). Also, the URL cipher 140 corresponding to the data type 136 (Fig. 3B) of the retrieved data record is used to format a URL by which the retrieved data record may be accessed. Further, the data translation and collection system 110 creates and opens an appropriate file folder and file to store the converted retrieved record as specified by the URL cipher 140 field.

In step 276, the File Format Instruction Table 134 (Fig. 3B) for the instant database is checked to determine if additional data types 136 are available. If so, in step 272 the process of steps 220 through 276 is repeated for the next data type 136. If the search has been performed for each data type in the instant database, the search proceeds to the next database indicated in database register 131 (Fig. 3A), starting with its first data type 136 and proceeding, in similar fashion, through each of its data types 136, until the search has been performed for every data type 136 of every database in database table 130 (Fig. 3A). The procedure progresses to step 284 after completing this search through the registered databases.

In step 284, the list of records to retrieve opened in step 212 is examined for the existence of records or program modules that have not yet been retrieved. If the list is empty, the data collection for the patient has been completed and the process advances to step 324 (Fig. 5E), discussed infra. If the list is not empty, in step 288 a request is sent for the first entry remaining in the list, which may be for a data record or a program module. If it is a data record, after it is retrieved, it is checked in step 290 for encryption and decoded, if necessary, using proprietary software.

In step 298, the record is parsed, discussed infra, to locate data references: hypertext links, multi-media requests, and key words or phrases. If none are found, the process advances to step 314, discussed infra. If there are data references a check is made to determine if the data begin referenced had been located previously (step 302). If it had not been previously located, the record is added to the List of Records to Retrieve (step 306). In step 310, all hypertext links and other data requests are reformatted through use of the URL Ciphers 140, maintained by the data translation and collection system 110 for each data type 136. This way, the URL or other data request addresses are compatible with the addressing convention to be used on the storage medium to which the records will be written. When the retrieved data record is later displayed through a network browser, secondary files referenced by the retrieved data record are made easily and quickly accessible to the system user with the click of a mouse.

In step 314, whether there were data references or not, Hypertext Cipher 138 is used to convert text to HTML format, graphics, audio, video, or other non-text information into a browser, plug-in, or Java compatible format.

In step 316, the data translation and collection system 110 creates and opens an appropriate file folder and file to store the converted retrieved record, either as specified by the URL cipher 140 field (Fig. 3B) (if the retrieved record is part of the patient's file), or with a distinctive file name (if the retrieved record is not part of the patient's file, e.g., a physician's biographical background). In step 320, the retrieved record or program module, as it may be, is removed from the list of records to retrieve, and steps 284 through 320 are repeated until the list is empty.

E. Workstation Data Collection and Translation

The operation of the data translation and collection system 110 (Fig. 1) set forth in Figs. 5A-5E may be initiated in other ways. In one mode of operation, the databases 106 (Fig. 1) on the hospital's communication network 112 may send data for each patient to the data translation and collection system 110 periodically or after the patient is discharged. In another mode of operation, the workstations 102 on the hospital's communication network 112 may send reports, such as those produced by wordprocessors, to the data translation and collection system for translation and storage. In yet another mode of operation, the data translation and collection system may have access to a drive, directory, or folder on one or more of the workstations 102 residing on the communication network, from which it may search and retrieve data records.

System users such as physicians often produce reports, such as word processing files, on their own workstations 102 or 104 (Fig. 1) that are relevant to a patient's condition, status, or profile, and which merit inclusion in the data translation and collection system 110 of the present invention. This need may be accommodated by placing any report that is to be retrieved by the data translation and collection system 110 in a special folder 105 named "Collection." The data translation and collection system 110 may maintain a file containing a Workstation Data Table 150, as set forth in Fig. 4A, which includes the addresses 152 of all workstations 102 and physician office workstations 104 and file access commands 154 or passwords used to gain access to files stored in each workstation's "Collection" folder 105. The data translation and collection system 110 may also maintain a Workstation File Formatting Instruction Table 158, as set forth in Fig. 4B, which includes each report name and corresponding file name and data formatting instructions 162 and Workstation URL cipher 166.

On a periodic basis or as instructed, a program in the data translation and collection system 110 (Fig. 1) may determine if there are any files in the special "Collection" folder 105 in each workstation 102, 104. If any files exist, the file access commands 154 (Fig. 4A) may be sent to the workstation so that the files may be transferred to the data translation and collection system 110. This may be done using the file transfer protocol, FTP, of the Internet/Intranet or by other data transfer methods.

If the user of the workstation 102, 104 creates reports, that when stored use a file name formatted according to file name and data formatting instructions 162 (Fig. 4B), the file may be recognized as being a specific file for a patient. For example, the file named "Cath987654321" may correspond to a catheterization report for the patient whose identification number is 987654321. Appending the date and time to the file name may be used to further identify the report. Alternatively, file name and formatting instructions 162 may require that the date and time be located within the report itself. Similarly, the report name and/or the patient's identification information may be incorporated in the report or its file name. In either case, once the file and its file name are received and recognized, the file may be processed in the same manner that data records retrieved from databases are processed as set forth in Figs. 5A-5E, but using the Workstation Data Table 150 and its Workstation File Formatting Instruction Tables 158 in place of the Database Table 130 and its File Formatting Instruction Tables 134.

Instruments or medical devices whose reports are not stored as part of any database 106 (Fig. 1), but that are capable of writing data to a floppy disk or transmitting information via an infrared or serial line connection to a workstation 102, 104 may also store patient reports in the data translation and collection system 110. To do so, the individual reports may be written to a floppy disk using any file name defined by the file name and data formatting instructions 162 (Fig. 4B). The reports may be copied manually or automatically from the floppy in workstation 102, 104 to the "Collection" folder 105, which may be periodically checked to see if there is any data in it to be retrieved, or the reports may be automatically read by the workstation and sent to the data translation and collection system 110. The reports so sent may be incorporated with any others received for this patient and may be provided with a new destination file according to the Workstation URL Cipher 166.

In this manner reports from wordprocessors or from mobile medical devices may be collected for display and storage and may be assigned structured file names to assist in their retrieval whether on line or when placed on line as with a CD-ROM device 117 (Fig. 2).

F. Creation, Structuring, and Mass-Media Storage of Patient Data

Commencing with step 324, Fig. 5F sets forth the process by which a patient's many data records may be stored on a mass storage device. In step 324, a master file 400 (Fig. 6A) is created as data records are received or when all the data records have been retrieved. The master file name may be based on the patient's name or identification number. In some cases it may be desirable to create the same file twice, using the patient's name for the file name once and the patient's identification number for the other. In step 328, secondary, tertiary, or other subdominant control files 418 (Fig. 8A) may also be created. A "master" file is roughly analogous to a root directory or a home page on a website, for through this single file all the patient's data records may be accessed through hypertext links. While a "master" file may contain text, graphics, video, or audio, it contains a list of links to other reports—for

example, the discharge report link 408 (Fig. 6B) to discharge report 412 (Fig. 7A)—or links to other "control" files—for example, the cardiology link 404 to cardiology control file 418 (Fig. 8A). "Control" files are roughly analogous to subdirectories. Although they incorporate text, graphics, or other multimedia features, they serve primarily to present an organized collection of links to related patient records.

The URL Cipher 140 corresponding to the date type 136 of each data record retrieved, can be used to determine whether a hypertext link to retrieve the data record from the mass storage device is to be placed in a master, secondary, or tertiary control file.

In step 332, the master and control files are written to a mass storage device along with the data records if they have not been previously been written to the mass storage device. A CD-ROM disk that has a patient's data written to it may be given to appropriate physicians for their own storage and use. To view the contents of the CD-ROM, a physician would need only to insert it into the CD-ROM drive 117 (Fig. 2) of a physician workstation 104 (Fig. 1) and run a network browser program. By using the File command the physician could refer to the CD-ROM drive 117, which would list the name of the master file 400, which may be the same or similar to the patient's name.

Fig. 6A sets forth an example of the contents of the master file 400. Besides identifying the patient and the dates and source of the medical records, the master file has a series of hypertext links 402 either to distinct reports, such as hypertext link 408, or to secondary control files, such as hypertext link 404. Fig. 6B sets forth how master file 400 (Fig. 6A) might appear through a browser program when presented on display 118. Note that hypertext links 402 are displayed in a different font format, as is the convention with browser programs. The system user may select a hypertext link by moving a pointing device such as a mouse over the text and pressing an activation button. The browser will automatically retrieve the file specified in the hypertext link 402 from the CD-ROM and present it.

If the system user selects the hypertext link 408 specifying discharge report in the master file 400, the system user will be presented with the patient's single discharge report 412 (Fig. 7A). Fig. 7A sets forth some of the HTML codes which may be used to format discharge report 412. Hypertext link 416 may be selected to retrieve the specified catheterization report from the CD-ROM. The hypertext link URL address has been modified as needed to make it compatible with the storage structure of the CD-ROM. Fig. 7B sets forth how discharge report 412 (Fig. 7A) might appear through a browser program when presented on display 118.

Fig. 8A sets forth the contents of a secondary control file 418 that a browser program would present if the system user, while viewing master file 400 (Fig. 6A), selected the hypertext link 404 specifying cardiology data. Secondary control file 418 presents the system user with various types of cardiology reports to choose from in the form of hypertext links 422. For those cardiology tests for which there is only one report available, the hypertext link specifies the URL address of that report. For those tests for which there are several reports available, the hypertext link 420 may specify the URL of a tertiary control file. Fig. 8B sets

forth how secondary control file 418 (Fig. 8A) might appear through a browser program when presented on display 118.

Fig. 9A sets forth the contents of tertiary control file 424 that a browser program would present if the system user, while viewing secondary control file 418 (Fig. 8B), selected the hypertext link 420 specifying electrocardiograph reports. This file presents the system user with a list of all the electrocardiograph reports, during the dates selected, to choose from in the form of hypertext links 426. For each electrocardiograph report the hypertext link specifies the URL address of the report. Fig. 9B sets forth how tertiary control file 424 (Fig. 9A) and its list of electrocardiograph reports might appear through a browser program when presented on display 118.

G. Parsing to Locate Data References

Fig. 15A illustrates how a data record is parsed. A data record is parsed to locate data references by searching it for text corresponding to a hypertext link or a multimedia data request. If one is found, the URL is located after the initial control sequence and will be saved (step 812) for use after the parsing is completed. If none are found, or when the record has been completely parsed, another pass can be made to search for data references in the form of key words or key phrases (step 820).

A key word or phrase is a recognized text string that is to be converted into a hypertext link. As an example, the data reference indicated by the phrase, "Admission ECG," can be converted (steps 828, 830) into the following hypertext link:

<a href="hww.st_mary.springfield/ecg/987654321/
03may1997/ecg/admission.html">Admission ECG.

The expression "03may1997" is the date the data record being parsed was created. The patient ID (987654321), the date, and other descriptors are available from steps 200 and 226, or from steps 544 or 560. A wide variety of medical expressions can be recognized as key words or phrases, and appropriate hypertext links created from them. The URL of the hyperlink is saved for later use (step 832). When the entire record has been searched (step 836), the URLs of the located data references are returned to the section of the flow chart that requested the record to be parsed (step 840).

H. Building Addresses Using Gleaned Information and Data References in Batch

It is contemplated that abbreviated database tables which includes at least some of the information included in database table 130, file format instruction tables 134 (including hypertext ciphers, URL ciphers and special instructions), tables 150 and 158 and data request

catalogue 500 can be used to "build" addresses of records referenced in a specific record during batch processing of the specific record (i.e. after the entire first record has been entered).

To this end, referring now to Fig. 21, a first abbreviated database table 1800 preferably includes only data references (DRs) 1802 and associated URL's or address formats 1804. The DRs 1802, like the keywords described above, are searchable terms or phrases which are commonly used to refer to specific occurrences which may be associated with stored records. For example, "admission ecg" may be a DR 1802. Other DRs may include "post-op x-ray", "PET image", a date, a patient's identification number, etc. The address formats 1804, like the URL's described above, specify required information needed to form an address of a record associated with the corresponding DR and also specify the sequence of address fields which have to be filled with the required information to form the address.

Referring to Fig. 22, the second abbreviated database table 1820 includes global instructions 1822 and a list of data types 1824 and corresponding record rule sets (RRSs) 1826. The global instructions 1822 include rules for identifying the data type 1824 of a record which is being searched (e.g. to identify key words or phrases for creating links to other records on the system databases 106, 2) in batch form. Different data types 1824 are associated with different record information configurations. For example, one five field record may include a date in the second field while another may include the date in the fourth field. Yet a third record may include the date in the second field but may also include a total of seven fields. These three records would be characterized by three different data types 1824, each different data type 1824 having a different information configuration.

The global instructions 1822 may take any form which, given the data types 1824 used with the wordprocessor 14, can be used to identify a data type 1824. For example, wordprocessor 14 may always provide a single field specifically reserved for a character or symbol which identifies the record data type 1824. For instance, an ecg report may always be entered into a HIS 111 (see Fig. 1) using a first data type template including specific fields for specific information (e.g. patient ID, date, time, physician, etc.). In this case, when an ecg data type template is opened to form an ecg record, wordprocessor 14 automatically enters a "1" in the data type field. Similarly, when a PET scan report is entered into the HIS, assuming PET scan reports are of a second record type, wordprocessor automatically enters a "2" in the data type field.

In the alternative, other global instructions 1822 may specify rules by which wordprocessor 14 can glean information entered by a user into record or template fields to determine the data type. For example, where a user enters a date into a fourth field instead of into a second field, wordprocessor 14 can distinguish a unique data type 1824 or at least a sub-group of data types. Similarly, wordprocessor 14 may recognize specific terms entered into certain fields to identify data type 1824. For example, when a user enters "ecg" into a first field, wordprocessor 14 may recognize a specific data type based on global instructions 1822.

While only two different examples of global instruction have been described herein, the examples are not meant to be exhaustive and other examples are contemplated.

Referring still to Fig. 22, the data types 1824 correspond to the different field configurations of the various record which are stored on the in the system databases 106, 2. The RRSs 1826 each include a set of rules for gleaning information from a record which is being searched (e.g. to identify key words or phrases for creating links to other records on the system databases 106, 2). For example, where DT-1 1828 indicates a first record type having five fields including a first patient ID field, a second date field, a third report type field, a fourth physician name field and a fifth text field, the corresponding RRS-1 1830 includes rules which specify that to glean the patient ID, date, report type and physician name, wordprocessor 14 must access the first, second, third and fourth fields, respectively.

Referring now to Fig. 16, a batch method performed by wordprocessor 14 (see also Fig. 1) is illustrated. The method of Fig. 16 will be described in the context of an exemplary process. To this end, it will be assumed that a plurality of records have already been stored at database addresses according to the methods described above and in accordance with the specifications of the tables of Figs. 21 and 22. Thus, each of the stored records is associated with a DR 1802 and is stored at a database address which has a format indicated by an associated address format 1804 in the table of Fig. 21. For example, a record associated with DR-3 1808 is stored at an address having a format consistent with Format 3 1810. Records which have already been stored will be referred to generally as first records.

In addition, referring also to Fig. 17, it will be assumed that the record illustrated therein (hereinafter a "second record") has been entered into the HIS and stored on one of the system databases 106, 2. The record in Fig. 17 is a report created using wordprocessor 14. The record includes DRs 1608 which reference a plurality of the first data records. Initially it is contemplated that DRs 1608 would not be highlighted but that, after wordprocessor 14 forms links between DRs 1608 and records corresponding to the DRs 1608, the DRs 1608 would be highlighted via bolding or a different color text. In Fig. 17 the DRs 1608 include "admission ecg," "previous ecg," "previous discharge cath," and "admission CK enzyme" referencing various stored first records.

For each of the DRs 1608 (i.e. keywords) in Fig. 17, wordprocessor 14 is capable of recognizing these DRs and correlating the DRs with address formats 1804 via table 1800. In addition, wordprocessor 14 is also capable of determining the data type 1828 of the record shown in Fig. 17 and an associated RRS 1826 using the global instructions 1822 from table 1820.

For the purposes of this explanation it will be assumed that the data type 1824 for the record illustrated in Fig. 17 includes five fields. In addition to a text field 1607, the four other fields include a patient ID field 1600, a date field 1602, a report type field 1604 and an author field 1606. It is also assumed that each of fields 1600, 1602, 1604, 1606 and 1607 already includes the information illustrated in Fig. 17.

In addition, the record of Fig. 17 also includes a data type field 1610. In the present example, it is assumed that, at an earlier time, when a physician accesses wordprocessor 14 to create a record, the physician indicates the data type to the wordprocessor in some manner. Data type may be indicated by selecting a data type from a list (e.g. ecg, PET report, post op X-ray, etc.) or it may be typed or it may be indicated in any other manner. When the physician indicates a data type 1824, word processor 14 places a character or a character string in data type field 1610 indicating the data type of the record being created. In addition, wordprocessor 14 thereafter can provide fields to be filled in which are consistent with the specified data type 1824. In this case, it is assumed that "DT-1" indicates an admission report having fields 1600, 1602, 1604, 1606, 1607 and 1610.

With the second record completed as illustrated in Fig. 17 and stored on one of the system databases 106, 2, a command is issued to wordprocessor 14 to search the second record to identify any references to first records which occur in the second record and, when a reference to a first record is located, to form a link between the reference and the referenced first record. To this end, wordprocessor 14 performs the process of Fig. 16. Wordprocessor 14 separately receives each phrase in field 1607 (where the phrases include groupings of N or less consecutive words where N is the maximum number of consecutive words which may be included in a DR 1802), and compares each phrase to DRs 1802 in table 1800 to identify DRs 1802. Where a phrase does not match a DR 1802, wordprocessor 14 jumps to the next entered phrase.

Referring to Figs. 16 and 17, when the phrase "admission ecg" is received as a second record segment at step 1500, at step 1502 wordprocessor 14 accesses table 1800 and compares the phrase "admission ecg" to the DRs 1802 until either a match is identified or until the phrase has been compared to all of the DRs 1802. In the alternative, table 1800 entries may be stored alphabetically and wordprocessor 14 may be equipped to recognize the first letter in a phrase. In this case, wordprocessor 14 may compare the phrase only to DRs 1802 which begin with the first letter of the phrase being compared to speed up the comparison process. At decision block 1504, where the phrase does not match a DR 1802, control passes back to processes step 1500 where the next record phrase or segment is received for comparison.

However, at block 1504, where the phrase matches a DR 1802, control passes to block 1508. In the present example it will be assumed that DR-3 1808 corresponds to the phrase "admission ecg". Thus, the phrase "admission ecg" matches a DR-3 1808 and control passes to block 1508. At block 1508 wordprocessor 14 uses table 1800 to identify address format (i.e. URL cipher) 1810 which corresponding to DRs 1808. As indicated above, the address format specifies a format of an address associated with DR-3 1808 and also specifies the required information needed to form the record address.

Next, at process block 1510, wordprocessor 14 accesses the global instructions 1822 in table 1820 and uses the rules therein to determine the data type 1824 of the second record. In the present example, the global instructions 1822 instruct the wordprocessor to access data

type field 1610 to identify the data type 1824. Accessing field 1610, wordprocessor 14 determines that the second record data type is DT-1. Accessing table 1824, wordprocessor 14 correlates data type DT-1 1828 with RRS-1 1830. As indicated above, RRS-1 1830 specifies rules for how to glean the required information from the record illustrated in Fig. 17. For example, address format 1810 may require, among other information, a patient ID number and a date used to locate reports for a particular patient related to the date. In the present example, RRS-1 1830 may specify that the information in field 1600 corresponds to a patient ID and that the information in field 1602 corresponds to the current date.

Now using RRS-1 1830, at process block 1512, wordprocessor 14 gleans the required information as specified by address format 1810 from the second record in the manner specified by RRS-1 1830. To this end, in the present example, wordprocessor 14 gleans the patient ID number and the date from fields 1600 and 1602.

Next, at block 1512, wordprocessor 14 forms an address for the record referenced by DR-3 1808. At block 1514 wordprocessor 14 automatically highlights the DR 1608 "admission ecg" in text field 1607 thus providing the DR "admission ecg" as a selectable segment. In addition, wordprocessor 14 links the DR "admission ecg" to the formed address such that, when the DR "admission ecg" is selected (e.g. via a mouse controlled cursor or the like), the wordprocessor 14 automatically accesses the record stored at the formed address and provides the record to a user for review.

In addition to information facially included in the second record, wordprocessor 14 may also glean other information associated with the second record for building a record address. For example, wordprocessor 14 may be associated with a specific facility and therefore may associate every record generated by the wordprocessor 14 with the specific facility (i.e. St. Mary's Springfield). In this case, when gleaning information, wordprocessor 14 may also glean information specifying the specific facility which is required to form an address. This information may be gleaned from any of a variety of locations including admit system 108, hospital information system 111 or some other system linked to a hospital database 106, 2.

This process of comparing record segments to DRs 1802 and forming links between DRs 1802 and records referenced by the DRs 1802 is continued on the second record text. In the present example links are formed between phrases (i.e. DRs 1802) "previous discharge cath" and "admission CK enzyme" and records referenced thereby. For example, the address linked to the phrase "admission ecg" is identified by number 1700 while the address linked to the phrase "previous discharge ecg" is identified by number 1702.

In addition, information in fields 1600 through 1606 may also be recognizable DRs 1802. In this case wordprocessor 14 also forms links between information in those fields and corresponding records by first determining if the information in a field is a DR 1802, identifying the address format 1804 corresponding to the DR 1802, identifying required information for forming an address, identifying a data type 1824 corresponding to the record being searched, identifying the RRS 1826 associated with the data type 1824, gleaning the required

information as indicated by the RRS 1826, forming an address for the record referenced by the DR 1802 by combining the required information, providing the DR 1802 as a selectable segment and linking the selectable segment to the record via the formed address. For example, the address linked to the patient ID number and associated patient demographics is identified by number 1704 and the address linked to the staff directory information for Dr. Markelson is identified by number 1706.

Referring now to Fig 18 therein is illustrated an HTML document corresponding to the document of Fig. 17 including hypertext addresses formed according to the process of Fig. 16. It can be seen that each of the DRs 1608 of Fig. 17 now includes a linking address.

Referring also to Fig. 19, the document of Fig. 18 is illustrated as the document would be viewed via wordprocessor 14 or a standard network browser (not illustrated). As illustrated, six selectable segments have been highlighted (i.e. underlined), a separate selectable segment for each recognized DR.

It should be noted that while the above-referenced batch processing to build record addresses has been described in the context of the wordprocessor 14, some other data translation and collection system (e.g. 110 in Fig. 1) may be provided to perform exactly the same functions in batch format.

It should also be noted that while specific fields may be provided in a record template for entering specific types of information, in another embodiment, a single field may be able to receive more than one type of information from which a wordprocessor or other type of device could glean the separate types of information for building a record address. For example, in one embodiment, a patient's ID number, a date, a time and perhaps other information may be provided in a single field, the wordprocessor 14 able to parse information in the single field to identify specific types of information. In this case, it is contemplated that there would have to be rules to avoid duplication of specific types of information in the single field. For example, there would have to be a rule that no more than one patient ID could be provided in the field or that, if more than a single patient ID were provided in the field, one of the ID numbers would have to be selected for generating an address.

In yet another embodiment of the present invention, referring to Figs. 21, 22 and 23, portions of the data in tables 21 and 22 are combined to form even a more abbreviated database table 1900 which can be used to form database addresses for DRs which reference a first record in a second record. To this end, table 1900 includes a data reference (DR) list 1902, and both address formats 1904 and record rule sets (RRSs) 1906 which are correlated with the DRs 1902. In this embodiment the DR list 1902 and address formats 1904 have the same purposes and forms as the list in Fig. 21.

However, instead of being linked to a second record type, a separate RRS 1906 is linked to each of the unique DRs 1902. In this case, each RRS 1906 includes a set of rules which, independent of a second record's data type, indicate how to glean required information for forming a record address form the second record information. For example, a particular medical facility may require a patient ID be identified as "ID:__" followed by a 9 digit number.

In this case, the RRSs 1906 would specify that the term "ID:__" followed by a 9 digit number is a patient identification which can be used to populate a patient ID field in an address format. Similarly, the RRSs 1906 in this example include other rules which can be used to glean information from a second record for forming an address.

Referring to Fig. 16, the method illustrated therein would be essentially the same using the table of Fig. 23 instead of the tables of Figs. 21 and 22 and therefore, the method using table 1900 will not be explained here in detail. The only difference in the method of Fig. 16 when using table 1900 is that, at process block 1510, the wordprocessor does not have to identify the data type and identifies the RRS by correlating the RRS with the DR in table 1900.

I. Real Time Operation Of Address Building Using Ciphers

U.S. Patent Application Number 08/727,293 which was filed on October 9, 1996 and is entitled "Method and System for Automated Data Storage and Retrieval with Uniform Addressing Scheme" is incorporated herein by reference. That application teaches a system whereby, as record information is entered into a record, a wordprocessor analyzes the information in real time to identify keywords, roots or data references (DRs) which are references to other records which are stored or which may be subsequently stored on one of the system databases 106, 2. It is contemplated that the abbreviated database tables illustrated in Figs. 21 and 22 can be used to "build" addresses of records referenced in a specific record as the record is entered into a wordprocessor database 2. Thus, the process of forming record addresses described above in the context of batch processing (i.e. processing after the records have already been entered and stored) can be practiced in real time.

To this end, referring again to Fig. 16, the wordprocessor method illustrated therein can also be performed in real time. To describe real time operation of the wordprocessor 14 to generate links between records, once again, it will be assumed that a plurality of records have already been stored at database addresses according to the methods described above and in accordance with the specifications of the tables of Figs. 21 and 22. Thus, each of the stored records is associated with a DR 1802 and a corresponding data type 1828 and address format 140 1804. Records which have already been stored will be referred to generally as first records. In addition, referring also to Fig. 17, it will be assumed that the second record illustrated therein is being entered into the database 2 in real time.

For each of the DRs 1802 (i.e. keywords), wordprocessor 14 is capable of recognizing these DRs 1802 in the document illustrated in Fig. 17 and correlating the DRs 1802 with address formats 1804 via table 1800. In addition, wordprocessor 14 is also capable of determining the data type 1828 of the record shown in Fig. 17 as that record is being input in real time and an associated RRS 1826 using the global instructions 1822 from table 1820.

As in the batch example above, the record of Fig. 17 includes a data type field 1610. In the present example, it is assumed that, when a physician initially accesses wordprocessor 14 to create the record of Fig. 17, the physician indicates the data type to the wordprocessor in some manner. For example, data type may be indicated by selecting a data type from a list (e.g. ecg, PET report, post op X-ray, etc.). When the physician indicates a data type 1824, word processor 14 places a character or a character string in data type field 1610 indicating the data type of the record being created. In addition, wordprocessor 14 thereafter can provide fields to be filled which are consistent with the specified data type 1824. Once again, it is assumed that "DT-1" indicates an admission report having fields 1600, 1602, 1604, 1606, 1607 and 1610.

After fields 1600 through 1606 are filled, the physician is prompted to enter report text into field 1607. During text entry, wordprocessor 14 performs the process of Fig. 16. To this end, processor 14 receives each phrase entered into field 1607 (where the phrases include groupings of N or less consecutive words where N is the maximum number of consecutive words which may be included in a DR 1802), and compares each phrase to DRs 1802 in table 1800 to identify DRs 1802. Where a phrase does not match a DR 1802, wordprocessor 14 jumps to the next entered phrase.

Referring to Figs. 16 and 17, when the phrase "admission ecg" is received as a second record segment at step 1500, at step 1502 wordprocessor 14 accesses table 1800 and compares the phrase "admission ecg" to the DRs 1802 until either a match is identified or until the phrase has been compared to all of the DRs 1802. At decision block 1504, where the phrase does not match a DR 1802, control passes back to processes step 1500 where the next record phrase or segment is received for comparison.

However, at block 1504, where the phrase matches a DR 1802, control passes to block 1508. In the present example it will again be assumed that DR-3 1808 corresponds to the phrase "admission ecg". Thus, the phrase "admission ecg" matches a DR-3 1808 and control passes to block 1508. At block 1508 wordprocessor 14 uses table 1800 to identify address format (i.e. URL cipher) 1810 which corresponding to DRs 1808. As indicated above, the address format specifies a format of an address associated with DR-3 1808 and also specifies the required information needed to form the record address.

Next, at process block 1510, wordprocessor 14 accesses the global instructions 1822 in table 1820 and uses the rules therein to determine the data type 1824 of the second record. In the present example, the global instructions 1822 instruct the wordprocessor to access data type field 1610 to identify the data type 1824. Accessing field 1610, wordprocessor 14 determines that the second record data type is DT-1. Accessing table 1824, wordprocessor 14 correlates data type DT-1 1828 with RRS-1 1830. As indicated above, RRS-1 1830 specifies rules for how to glean the required information from the record illustrated in Fig. 17. For example, address format 1810 may require, among other information, a patient ID number and a date used to locate reports for a particular patient related to the date. In the present

example, RRS-1 1830 may specify that the information in field 1600 corresponds to a patient ID and that the information in field 1602 corresponds to the current date.

Now using RRS-1 1830, at process block 1512, wordprocessor 14 gleans the required information as specified by address format 1810 from the second record in the manner specified by RRS-1 1830. To this end, in the present example, wordprocessor 14 gleans the patient ID number and the date from fields 1600 and 1602.

Next, at block 1512, wordprocessor 14 forms an address for the record referenced by DR-3 1808. At block 1514 wordprocessor 14 automatically highlights the DR 1608 "admission ecg" in text field 1607 thus providing the DR "admission ecg" as a selectable segment. In addition, wordprocessor 14 links the DR "admission ecg" to the formed address such that, when the DR "admission ecg" is selected (e.g. via a mouse controlled cursor or the like), the wordprocessor 14 automatically accesses the record stored at the formed address and provides the record to a user for review.

This process of comparing record segments to DRs 1802 and forming links between DRs 1802 and records referenced by the DRs 1802 is continued as text is entered into the record text field 1607. In the present example links are formed between phrases (i.e. DRs 1802) "previous discharge cath" and "admission CK enzyme" and records referenced thereby.

In addition, as in the batch example above, information in fields 1600 through 1606 may also be recognizable DRs 1802. Here, the only difference in wordprocessor 14 operation is that the DRs in fields 1600 - 1606 are recognized as the fields are filled and, if all of the required information is not yet entered for building an address, the wordprocessor 14 waits to build the address until all of the required information has been provided. For example, referring still to Figs. 17 and 21, where the patient ID field 1600 is recognized as a DR 1802 and the date in field 1602 is required to form an address to a record corresponding to the patient and the date, if the date has not yet been entered, the wordprocessor 14 must wait for the date to be entered prior to forming the address. In the alternative, the wordprocessor 14 may be programmed to wait until all fields 1600 - 1606 are filled prior to building any addresses for DRs (i.e. processing of information in the preliminary, non-text fields may be done in batch).

It should be understood that while the real time addressing method described above is described in the context of the tables of Figs. 21 and 22, the table of Fig. 23 may be used instead, the only differences being that in step 1510 wordprocessor 14 does not identify the second record data type and that the wordprocessor 14 determines the RRS by correlating an RRS 1906 in table 1900 with an identified DR 1902.

While a particular embodiment of the invention has been illustrated and described, it will be obvious to those skilled in the art that various changes and modifications may be made without sacrificing the advantages provided by the principle of construction disclosed herein. For example, while the present invention has been described above in the context of an interface wherein a physician enters record information via a keyboard, clearly the invention is not to be so limited. For example, data may be dictated into a system, the inventive

wordprocessor including voice recognition software and identifying DRS as they are dictated and forming links between the DRS and records referenced thereby.

In the alternative, referring to Fig. 20, the system may be used with a data specifying device such as a hand held information gathering device (HHD) 1200 which downloads record information via a receiver (e.g. rf) 1202 to a wordprocessor 14, wordprocessor 14 generating address links either during the downloading process or in batch form after information transfer. One such HHD is described in U.S. Patent Application Number 09/170,169 which is entitled "Data Collection Device and System", which is commonly owned with the present application and which is incorporated herein by reference. In this type of system, the HHD may in fact specify a DR which can be used by processor to identify a DB address in any of several different ways. For example, processor 14 may simply correlate the DR with a DB address. In the alternative, processor 14 may build an address using the DR and an associated address format. In addition to specifying the DR, it is also contemplated that the HHD specifies other information for forming the DB address (e.g. background or general data set information such as patient ID, time, date, etc.). Also, it should be recognized that any data specifying device may be used such as a database which indicates a DR or the like. To this end, see the DB 106 linked to processor 14 via a network router 1206 or other network device (i.e. a receiver) in Fig. 20.

Furthermore, it should be recognized that the present invention may also be used in the case of a proxy server which routes address commands and requests among network devices. For example, the invention may build record addresses which do not actually match record addresses on a database but which, when provided to a proxy server, enable the proxy server to identify the actual record addresses and form a link.

To apprize the public of the scope of this invention, we make the following claims.

CLAIMS

What is claimed is:

1. A method to be used with a processor linked to at least one database, the at least one database including at least one second record, a first record including at least one first record phrases, at least one first record phrase associated with a keyword phrase which references the at least one second record, the method for rendering second records accessible from within the first record and comprising the steps of:

receiving at least one first record phrase;

determining if the first record phrase is associated with a keyword phrase;

if the first record phrase is associated with a keyword phrase, identifying the at least one second record which is referenced by the keyword phrase; and

rendering the identified record accessible.

2. The method of claim 1 wherein the processor is linked to an interface and wherein the step of receiving includes receiving an indication via the interface that the processor should attempt to recognize a first record phrase as a keyword phrase.

3. The method of claim 2 wherein the indicated first record phrase corresponds to more than one keyword phrase, the step of determining includes identifying each keyword phrase corresponding to the first record phrase and the step of identifying at least one second record includes identifying a separate second record for each identified keyword phrase.

4. The method of claim 3 wherein the step of identifying each keyword phrase corresponding to the first record phrase includes the step of comparing the first record phrase to a list of defined keyword phrases and, when the referencing phrase includes a threshold number of characters in a defined keyword phrase, identifying the defined keyword phrase as a keyword phrase.

5. The method of claim 3 wherein the interface includes a display, the step of rendering records accessible includes displaying the keyword phrases on the display in a selectable format and facilitating selection of at least one of the displayed keyword phrases.

6. The method of claim 2 wherein the interface includes a display and the step of receiving an indication includes the step of receiving an indication via the display which identifies the first record phrase.

7. The method of claim 2 wherein the step of receiving an indication includes the step of receiving a leading character which designates a first record phrase.

8. The method of claim 7 wherein the processor facilitates entry of first record text and the method further includes the steps of, as first record text is entered, monitoring the text for the leading character and, after a leading character is recognized, identifying the phrase thereafter as a first record phrase.

9. The method of claim 2 wherein the processor facilitates entry of first record text and wherein the step of receiving includes, as first record text is entered, receiving each newly entered phrase in the first record as a first record phrase.

10. The method of claim 1 wherein the keyword phrase is a keyword.

11. The method of claim 1 wherein the step of determining includes comparing first record phrases to keyword phrases.

12. The method of claim 1 wherein the processor is linked to a display, the step of rendering includes visually presenting the keyword phrase in a format which is distinguishable from other first record phrases and linking the record associated therewith to the presented keyword phrase such that, when the presented keyword phrase is selected, the linked record is accessed.

13. The method of claim 1 wherein the step of identifying the at least one second record includes the step of identifying an address for the at least one second record.

14. The method of claim 13 wherein the step of identifying an address includes forming an address for the at least one second record.

15. The method of claim 1 wherein the first record is associated with other information which characterizes the keyword phrase and the step of identifying the at least one second record further includes the steps of identifying other information which further characterizes the keyword phrase and, based on the keyword phrase and other information, identifying the at least one second record associated therewith.

16. The method of claim 15 wherein the other information is within the first record text.

17. The method of claim 15 wherein the first record is stored at a first record address and the other information is associated with the first record address.

18. The method of claim 15 wherein the processor facilitates entry of first record text and the method is performed as first record text is entered.

19. The method of claim 15 wherein the keyword is a second keyword and the other information is a first keyword.

20. The method of claim 12 wherein the processor is also linked to an interface which facilitates modification of first record phrases and, wherein the method further includes the steps of, monitoring modifications to the first record phrases and, when a first record phrase associated with a second record is modified such that the first record phrase no longer corresponds to a keyword phrase, de-linking the keyword phrase and the previously associated second record.

21. The method of claim 2 wherein the interface includes at least one input device and the step of receiving includes receiving an indication via the input device.

22. A method to be used with a processor linked to at least one database, the database including at least one second record, a first record including phrases, at least one first record phrase associated with a keyword phrase which references the at least one second record, the method for indicating that the first record phrase in the first record references another record and comprising the steps of:

identifying at least one phrase in a first record which references a second record; and

indicating the identified first record phrase in the first record.

23. The method of claim 22 wherein each second record is associated with at least one keyword phrase and the processor accesses a list of defined keyword phrases and the step of identifying includes determining if a phrases in the first record is associated with a defined keyword phrase and, if a phrase in the first record is associated with a defined keyword phrase, identifying the phrase as a first record phrase.

24. The method of claim 22 wherein the processor is also linked to an interface which facilitates modification of first record text including modification of indicated first record phrases and, when an indicated first record phrase is modified, the method further includes the step of eliminating indication of the indicated first record phrase in the first record. first record for instances of each keyword phrase in the list.

25. The method of claim 22 wherein the step of indicating includes the step of highlighting the first record phrase in the first record.

26. The method of claim 22 wherein the processor is linked to a display for displaying at least the first record and wherein the step of indicating includes highlighting the first record phrase within the first record on the display.

27. The method of claim 23 wherein the processor is linked to a display for displaying at least the first record and wherein the step of indicating includes the step of providing the keyword phrase in a separate window on the display.

28. The method of claim 22 wherein the processor facilitates entry of first record text and the step of identifying includes, as text is entered, determining if newly entered characters are associated with second records.

29. The method of claim 28 wherein the step of determining includes monitoring entered text a leading character which designates a phrase and, after a leading character is recognized, determining if the characters thereafter are associated with a second record.

30. The method of claim 22 for indicating a plurality of first record phrases in a first record which reference second records and wherein the steps of identifying and indicating are performed for each first record phrase which references a second record.

31. The method of claim 22 further including the steps of, after identifying, forming a link between the indicated first record phrase and the referenced record.

32. The method of claim 22 wherein the step of indicating includes the step of rendering the first record phrase selectable, the method further including the steps of, after rendering the first record phrase selectable, monitoring for selection of the first record phrase and, when the first record phrase is selected, providing the second record.

33. An apparatus to be used with a processor linked to at least one database, the at least one database including at least one second record, a first record including at least one first record phrases, at least one first record phrase associated with a keyword phrase which references the at least one second record, the apparatus for rendering second records accessible from within the first record and comprising:

a processor running a pulse sequencing program to perform the steps of:

receiving at least one first record phrase;

determining if the first record phrase is associated with a keyword phrase;

if the first record phrase is associated with a keyword phrase, identifying the at least one second record which is referenced by the keyword phrase; and

rendering the identified record accessible.

34. The apparatus of claim 33 wherein the processor is linked to an interface and wherein the processor performs the step of receiving by receiving an indication via the interface that the processor should attempt to recognize a first record phrase as a keyword phrase and wherein the identified record is rendered accessible via the interface.

35. The apparatus of claim 34 wherein the indicated first record phrase corresponds to more than one keyword phrase, the processor performs the step of determining by identifying each keyword phrase corresponding to the first record phrase and performs the step of identifying at least one second record by identifying a separate second record for each identified keyword phrase.

36. The apparatus of claim 35 wherein the processor performs the step of identifying each keyword phrase corresponding to the first record phrase by comparing the first record phrase to a list of defined keyword phrases and, when the referencing phrase includes a threshold number of characters in a defined keyword phrase, identifying the defined keyword phrase as a keyword phrase.

37. The apparatus of claim 35 wherein the interface includes a display and the processor performs the step of rendering records accessible by displaying the keyword phrases on the display in a selectable format and facilitating selection of at least one of the displayed keyword phrases.

38. The apparatus of claim 34 wherein the interface includes a display and the processor performs the step of receiving an indication by receiving an indication via the display which identifies the first record phrase.

39. The apparatus of claim 34 wherein the processor performs the step of receiving an indication by receiving a leading character which designates a first record phrase.

40. The apparatus of claim 39 wherein the processor facilitates entry of first record text and the processor further runs the pulse sequencing program to perform the steps of, as first record text is entered, monitoring the text for the leading character and, after a leading character is recognized, identifying the phrase thereafter as a first record phrase.

41. The apparatus of claim 34 wherein the processor facilitates entry of first record text and wherein the processor performs the step of receiving by, as first record text is entered, receiving each newly entered phrase in the first record as a first record phrase.

42. The apparatus of claim 33 wherein the keyword phrase is a keyword.

43. The apparatus of claim 33 wherein the processor performs the step of determining by comparing first record phrases to keyword phrases.

44. The apparatus of claim 33 wherein the processor is linked to a display, the processor perform the step of rendering by visually presenting the keyword phrase in a format which is distinguishable from other first record phrases and linking the record associated therewith to the presented keyword phrase such that, when the presented keyword phrase is selected, the linked record is accessed.

45. The apparatus of claim 33 wherein the processor performs the step of identifying the at least one second record by identifying an address for the at least one second record.

46. The apparatus of claim 45 wherein the processor performs the step of identifying an address by forming an address for the at least one second record.

47. The apparatus of claim 33 wherein the first record is associated with other information which characterizes the keyword phrase and the processor performs the step of identifying the at least one second record by further causing the program to perform the steps of identifying other information which further characterizes the keyword phrase and, based on the keyword phrase and other information, identifying the at least one second record associated therewith.

48. The apparatus of claim 47 wherein the other information is within the first record text.

49. The apparatus of claim 47 wherein the first record is stored at a first record address and the other information is associated with the first record address.

50. The apparatus of claim 47 wherein the processor facilitates entry of first record text and the processor performs the steps as first record text is entered.

51. The method of claim 47 wherein the keyword is a second keyword and the other information is a first keyword.

52. The apparatus of claim 44 wherein the processor is also linked to an interface which facilitates modification of first record phrases and, wherein the processor runs the program to further perform the steps of, monitoring modifications to the first record phrases and, when a first record phrase associated with a second record is modified such that the first record phrase no longer corresponds to a keyword phrase, de-linking the keyword phrase and the previously associated second record.

53. The apparatus of claim 44 wherein the interface includes at least one input device and the processor performs the step of receiving by receiving an indication via the input device.

54. An apparatus to be used with a processor linked to at least one database, the database including at least one second record, a first record including phrases, at least one first record phrase associated with a keyword phrase which references the at least one second record, the apparatus for indicating that the first record phrase in the first record references another record and comprising:

a processor running a pulse sequencing program to perform the steps of:

identifying at least one phrase in a first record which references a second record; and

indicating the identified first record phrase in the first record.

55. The apparatus of claim 54 wherein each second record is associated with at least one keyword phrase and the processor accesses a list of defined keyword phrases and the processor performs the step of identifying by determining if a phrases in the first record is associated with a defined keyword phrase and, if a phrase in the first record is associated with a defined keyword phrase, identifying the phrase as a first record phrase.

56. The apparatus of claim 54 wherein the processor is also linked to an interface which facilitates modification of first record text including modification of indicated first record phrases and, when an indicated first record phrase is modified, the processor further causes the program to perform the step of eliminating indication of the indicated first record phrase in the first record.

57. The apparatus of claim 54 wherein the processor performs the step of indicating by highlighting the first record phrase in the first record.

58. The apparatus of claim 54 wherein the processor is linked to a display for displaying at least the first record and wherein the processor performs the step of indicating by highlighting the first record phrase within the first record on the display.

59. The apparatus of claim 55 wherein the processor is linked to a display for displaying at least the first record and wherein the processor performs the step of indicating by the step of providing the keyword phrase in a separate window on the display.

60. The apparatus of claim 54 wherein the processor facilitates entry of first record text and the processor performs the step of identifying by, as text is entered, determining if newly entered characters are associated with second records.

61. The apparatus of claim 60 wherein the processor performs the step of determining by monitoring entered text for a leading character which designates a phrase and, after a leading character is recognized, determining if the characters thereafter are associated with a second record.

62. The apparatus of claim 54 for indicating a plurality of first record phrases in a first record which reference second records and wherein the processor performs the steps of identifying and indicating for each first record phrase which references a second record.

63. The apparatus of claim 54 wherein the processor further runs the program to perform the steps of, after identifying, forming a link between the indicated first record phrase and the referenced record.

64. The apparatus of claim 54 wherein the processor performs the step of indicating by rendering the first record phrase selectable, the processor further running the program to perform the steps of, after rendering the first record phrase selectable, monitoring for selection of the first record phrase and, when the first record phrase is selected, providing the second record.

09512125

ABSTRACT OF THE DISCLOSURE

An apparatus and method for linking a keyword phrase in a first record to a second record referenced by the keyword phrase including receiving the keyword phrase, identifying the second record and rendering the second record accessible and an apparatus and method for identifying a reference in a first record to a second record and indicating existence of the second record in the first record.

4454522

1

EXPRESS MAIL LABEL N 636 882 900 US

PTO/SB/01 (6-95)
Approved for use through 9/30/98. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Please type a plus sign (+) inside this box +

0010/PTO Rev. 6/95 — U.S. Department of Commerce Patent and Trademark Office

DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION

[X] Declaration Submitted with Initial Filing OR [ ] Declaration Submitted after Initial Filing

Attorney Docket Number	250591.90163
First Named Inventor	Carlos De La Huerga
COMPLETE IF KNOWN	
Application Number	
Filing Date	Herewith
Group Art Unit	
Examiner Name	

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe that I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD AND SYSTEM FOR AUTOMATED DATA STORAGE AND RETRIEVAL

(Title of the Invention)

the specification of which

[X] is attached hereto

OR

[ ] was filed on (MM/DD/YYYY) _____ as United States Application Number or PCT International Application Number _____ and was amended on (MM/DD/YYYY) _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code §119(a)-(d) or §365(b) of any foreign application(s) for patent or inventor's certificate or §365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached? YES	NO
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]
			[ ]	[ ]	[ ]

[ ] Additional foreign applications numbers are listed on a supplemental priority sheet attached hereto:

I hereby claim the benefit under Title 35, United States Code §119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Date (MM/DD/YYYY)

[ ] Additional provisional application numbers are listed on a supplemental priority sheet attached hereto.

Burden Hour Statement: This form is estimated to take .4 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

Please type a plus sign (+) inside this box →

DECLARATION

I hereby claim benefit under Title 35, United States Code §120 of any United States application(s), or §365(C) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application or PCT international application in the manner provided in the first paragraph of Title 35, United States Code §112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations §1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. Parent Application Number	PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number *(if applicable)*
08/727,293 08/871,818 09/247,349		10/09/96 06/09/97 02/10/99	5,895,461 5,903,889

☐ Additional U.S. or PCT international application numbers are listed on a supplemental priority sheet attached hereto

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and all continuation and divisional applications based thereon, and to transact all business in the Patent and Trademark Office connected therewith:

☐ Firm Name _____ Customer Number or label _____

OR

☒ List attorney(s) and/or agent(s) name and registration number below

Name	Registration Number	Name	Registration Number
Neil E. Hamilton	19,869	Jean C. Baker	35,433
Thomas W. Ehrmann	20,374	David G. Ryser	36,407
Barry E. Sammons	25,608	Bennett J. Berson	37,094
J. Rodman Steele	25,931	Michael A. Jaskolski	37,551
Nicholas J. Seay	27,386	Allen J. Moss	38,567
George E. Haas	27,642	Richard T. Roche	38,599
Michael J. McGovern	28,326	Sherry Whitney	39,422
Carl R. Schwartz	29,437	Mark D. Passler	40,764
Gregory A. Nelson	30,577	Jill A. Fahrlander	42,518
Keith M. Baxter	31,233	Scott D. Paul	42,984
John D. Franzini	31,356	Daniel G. Radler	43,028
Joseph W. Bain	34,290	Steven J. Wietrzny	P44,402
Robert J. Sacco	35,667		

☐ Additional attorney(s) and/or agents named on a supplemental priority sheet attached hereto

Please direct all correspondence to ☐ Customer Number or label _____ OR ☒ Fill in correspondence address below

Name	Michael A. Jaskolski				
Address	Quarles & Brady LLP				
Address	411 East Wisconsin Ave. Suite 2550				
City	Milwaukee	**State**	WI	**Zip**	53202-4497
Country	USA	**Telephone**	(414) 277-5000	**Fax**	(414) 271-3552

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Name of Sole or First Inventor: ☐ A petition has been filed for this unsigned inventor

Given	Carlos	Middle		Family	De La Huerga	Suffix	
Inventor's Signature						Date	Feb 22, 2000
Residence:	Milwaukee	State	WI	Country	USA	Citizenship	US
Post Office	9190 North Upper River Road						
Post Office							
City	Milwaukee	State	WI	Zip	53217	Country	Applicant Authority

☐ Additional inventors are being named on supplemental sheet(s) attached hereto

.///c:/APPS/preexam/corresponde

Bib Data Sheet

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

SERIAL NUMBER	FILING DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
09/512,125	02/24/2000 RULE _	707	2771	250591.90163

APPLICANTS
Carlos De La Huerga, Milwaukee, WI ;

** CONTINUING DATA ********** yes **********
THIS APPLICATION IS A CIP OF 09/247,349 02/10/1999
WHICH IS A CIP OF 08/727,293 10/09/1996 PAT 5,895,461
AND CLAIMS BENEFIT OF 60/023,126 07/30/1996
WHICH IS A CIP OF 08/871,818 06/09/1997 PAT 5,903,889

** FOREIGN APPLICATIONS ********** none **********

IF REQUIRED, FOREIGN FILING LICENSE GRANTED **
** 04/19/2000

Foreign Priority claimed	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
☐ yes ☑ no 35 USC 119 (a-d) conditions met ☐ yes ☑ no ☐ Met after Allowance Verified and Acknowledged Examiner's Signature: Frantz Coby Initials: FC	WI	30	64	4

ADDRESS

Michael A Jaskolski
Quarles And Brady LLP
411 East Wisconsin Avenue
Suite 2550
Milwaukee ,WI 53202

TITLE

Method and system for automated data storage and retrieval

FILING FEE RECEIVED	FEES: Authority has been given in Paper	
1560	No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

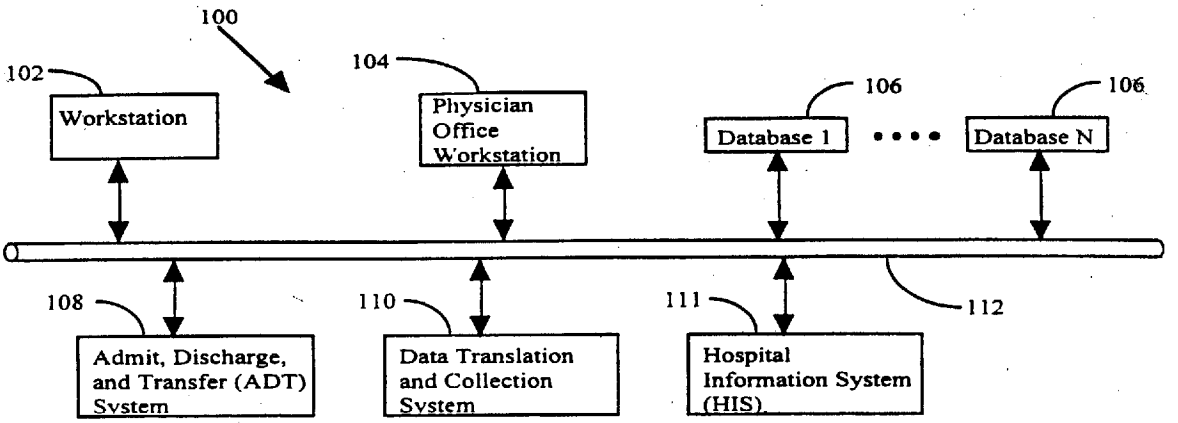


Figure 1

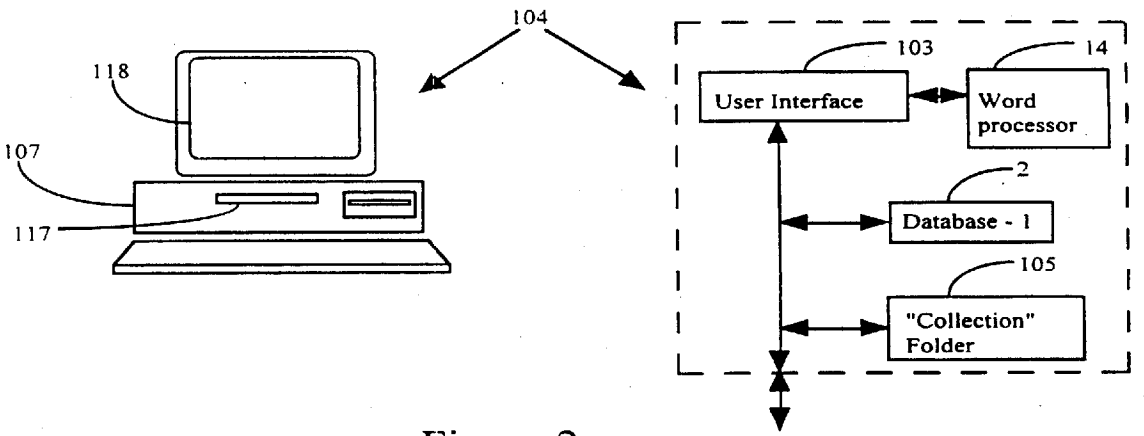


Figure 2

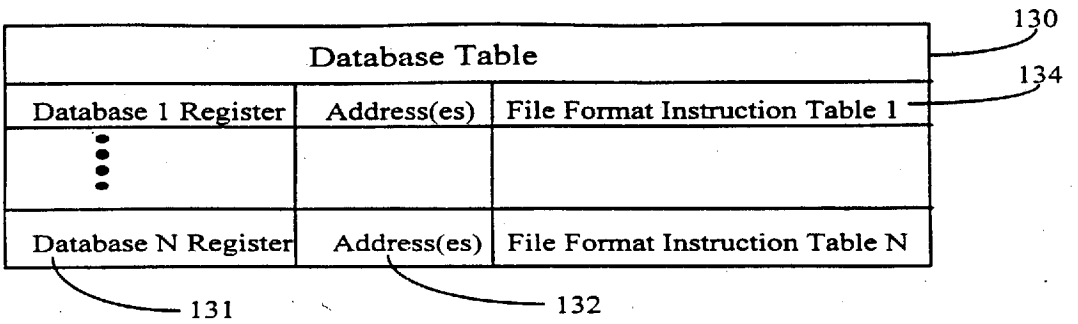


Figure 3A

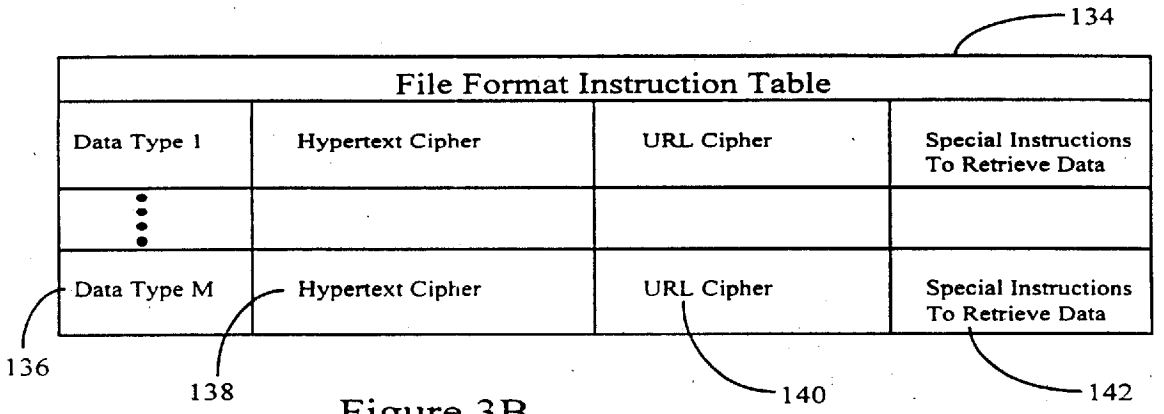


Figure 3B

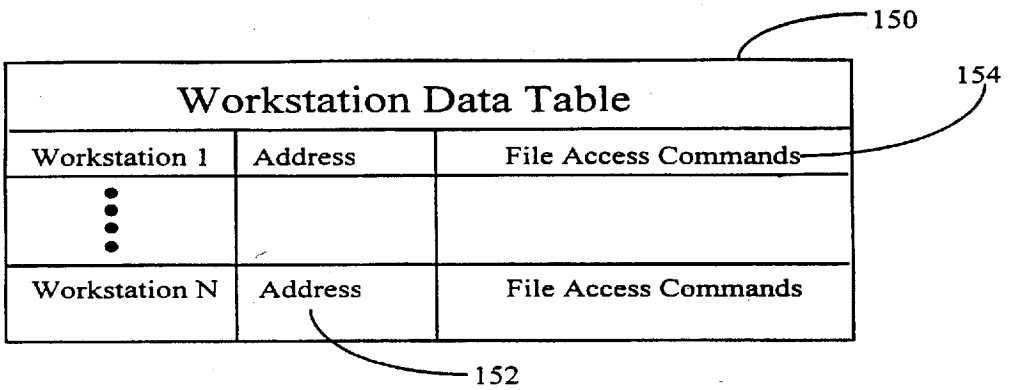


Figure 4A

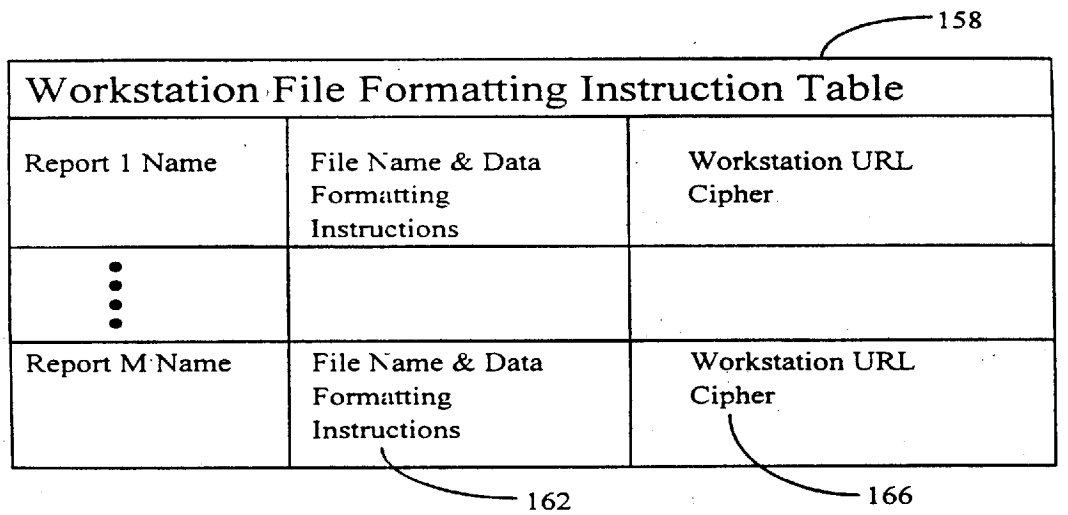


Figure 4B

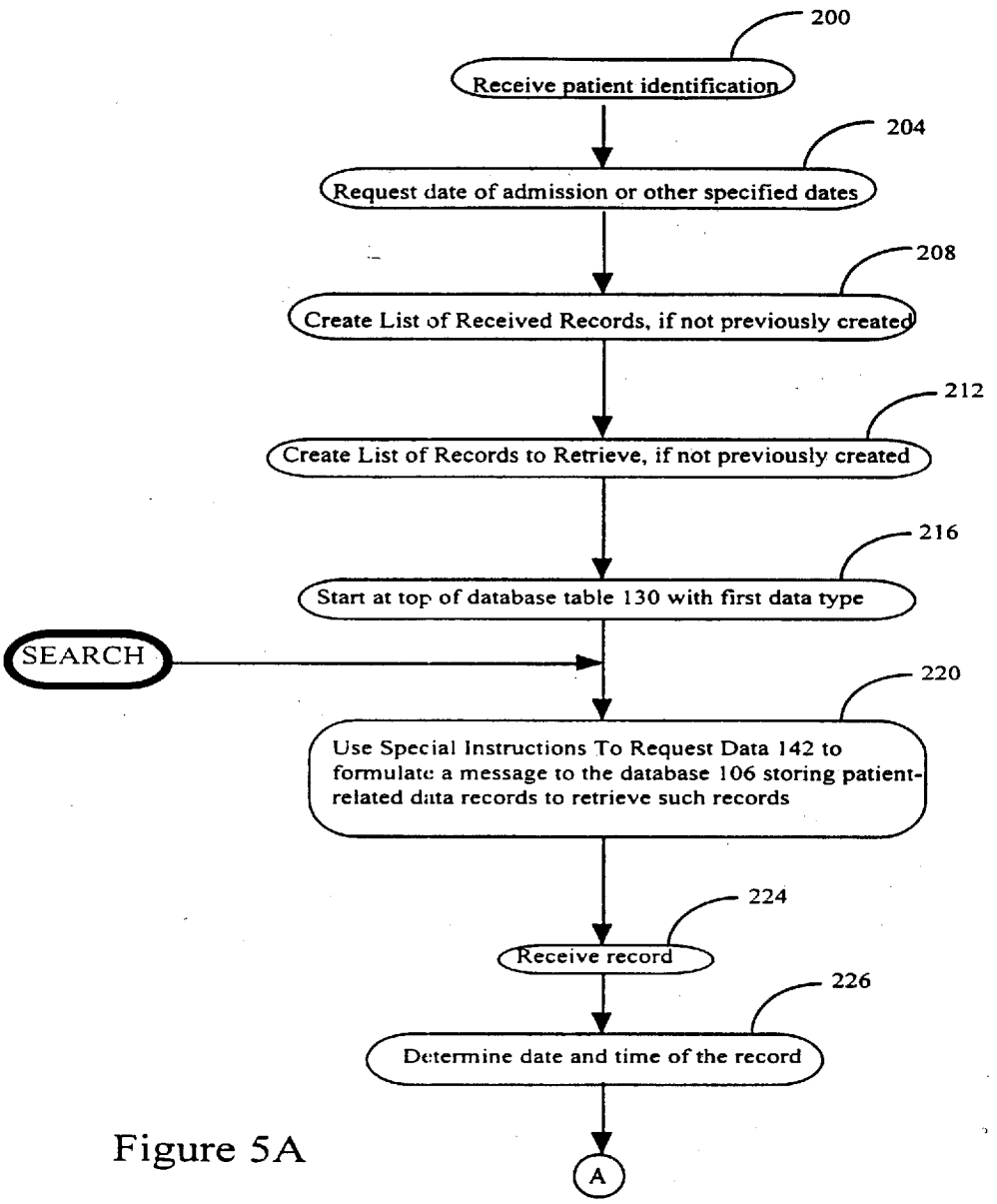


Figure 5A

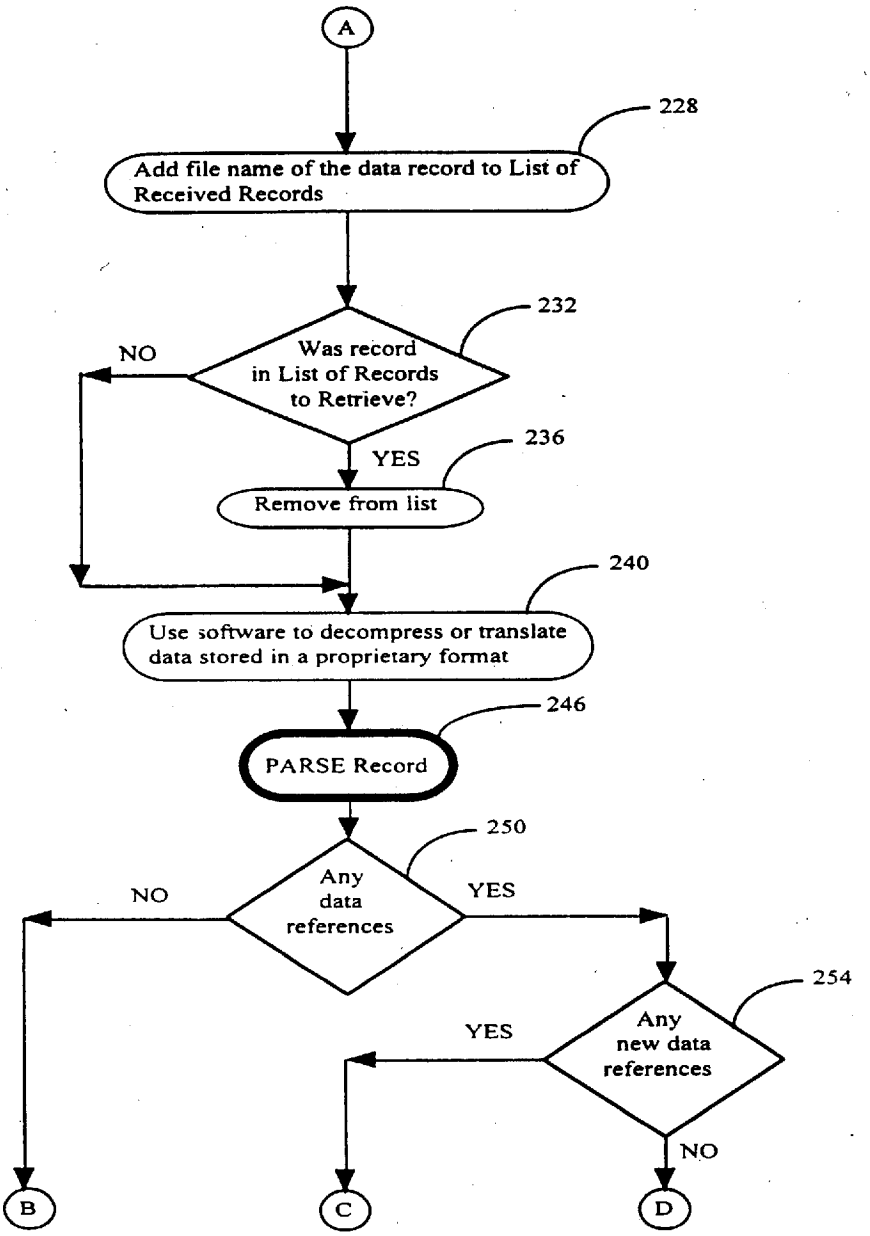

A
228
Add file name of the data record to List of Received Records
232
Was record in List of Records to Retrieve?
NO
YES
236
Remove from list
240
Use software to decompress or translate data stored in a proprietary format
246
PARSE Record
250
Any data references
NO
YES
254
Any new data references
YES
NO
B
C
D

Figure 5B

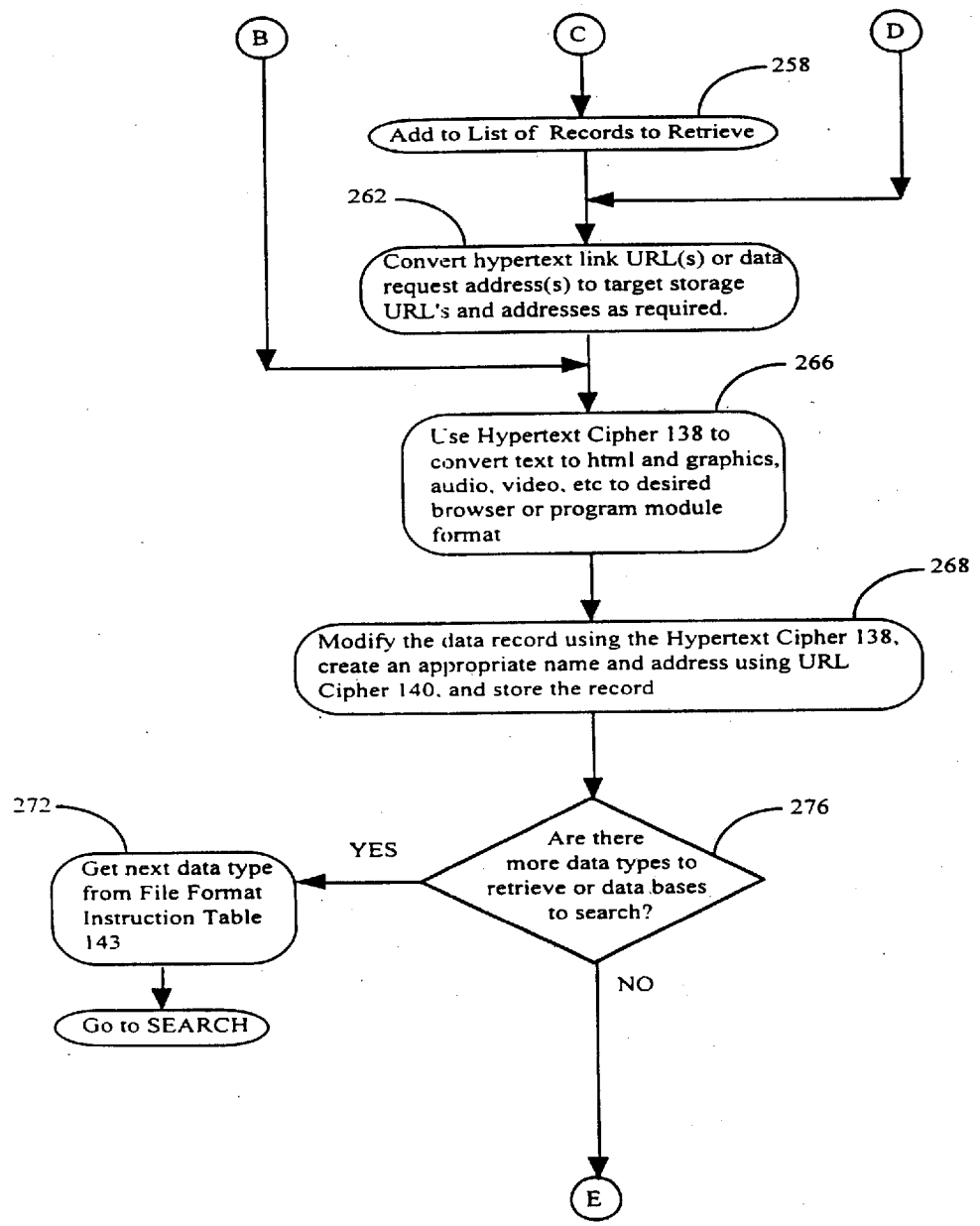

B
C
D
258
Add to List of Records to Retrieve
262
Convert hypertext link URL(s) or data request address(s) to target storage URL's and addresses as required.
266
Use Hypertext Cipher 138 to convert text to html and graphics, audio, video, etc to desired browser or program module format
268
Modify the data record using the Hypertext Cipher 138, create an appropriate name and address using URL Cipher 140, and store the record
272
Get next data type from File Format Instruction Table 143
YES
Are there more data types to retrieve or data bases to search?
276
NO
Go to SEARCH
E

Figure 5C

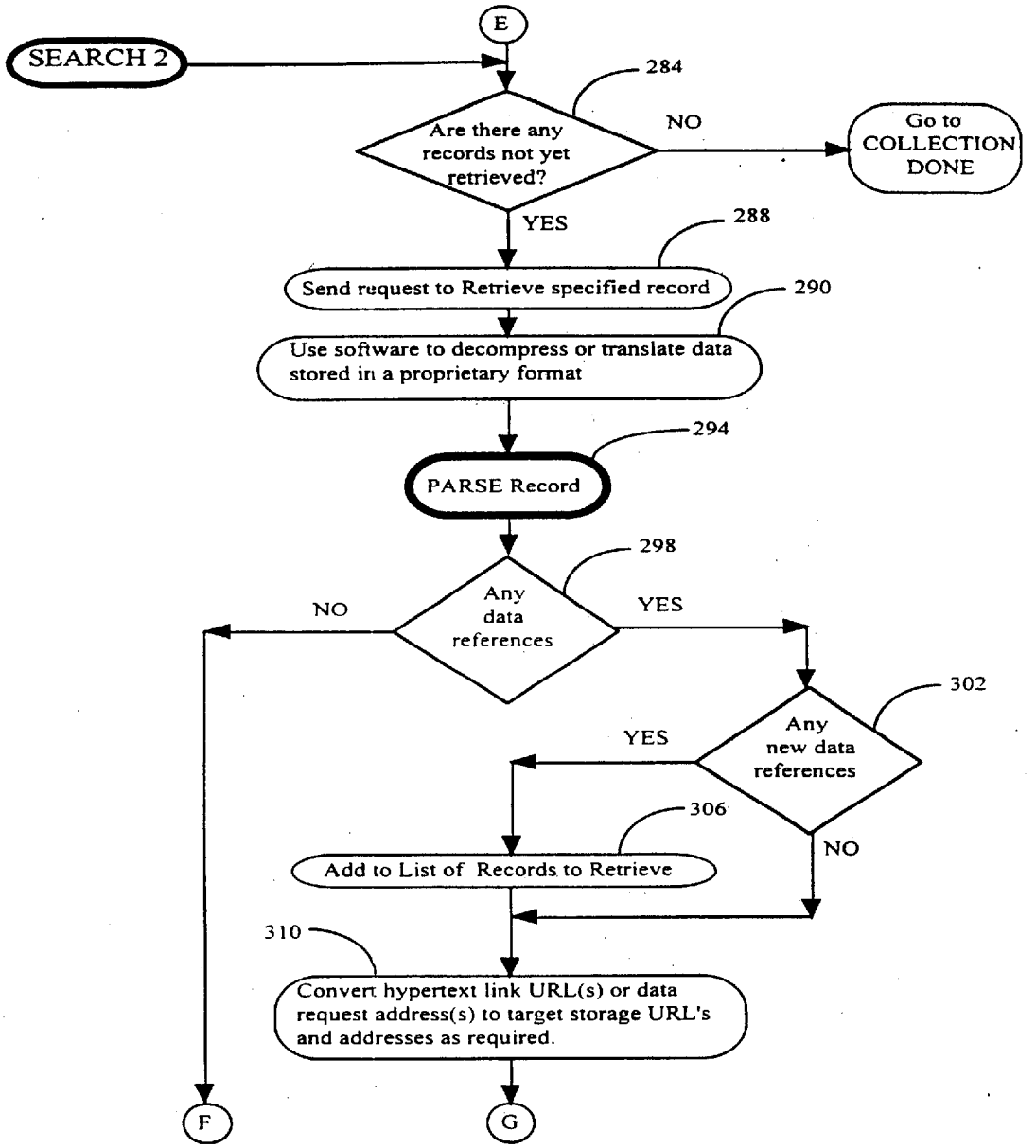

SEARCH 2
E
284
Are there any records not yet retrieved?
NO
Go to COLLECTION DONE
YES
288
Send request to Retrieve specified record
290
Use software to decompress or translate data stored in a proprietary format
294
PARSE Record
298
Any data references
NO
YES
302
Any new data references
YES
306
Add to List of Records to Retrieve
NO
310
Convert hypertext link URL(s) or data request address(s) to target storage URL's and addresses as required.
F
G

Figure 5D

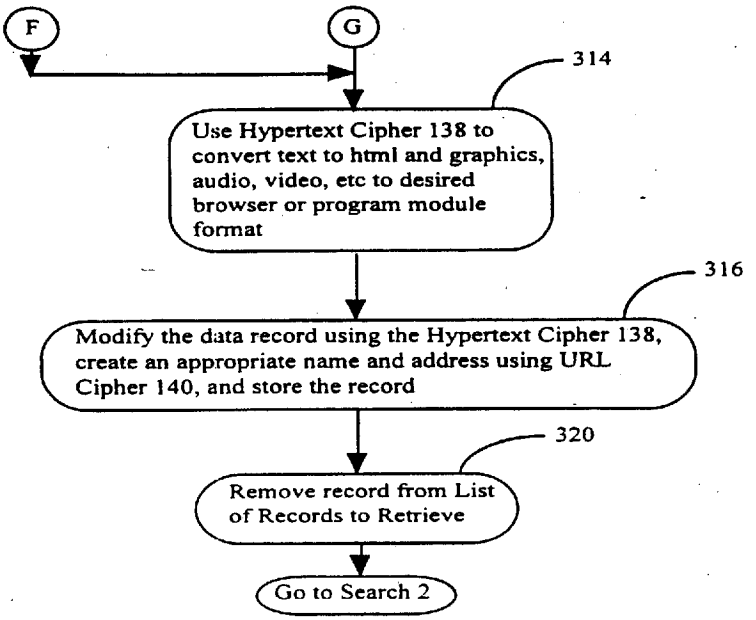


Figure 5E

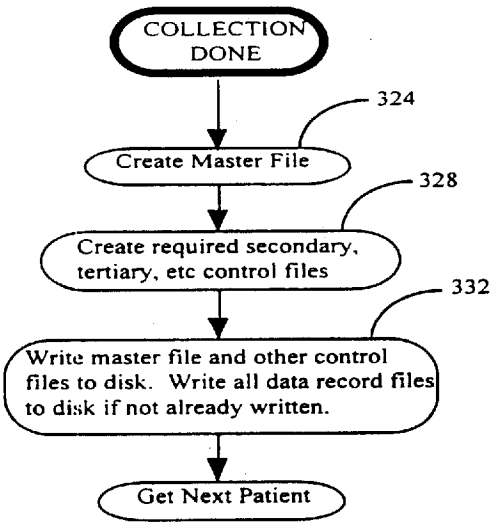


Figure 5F

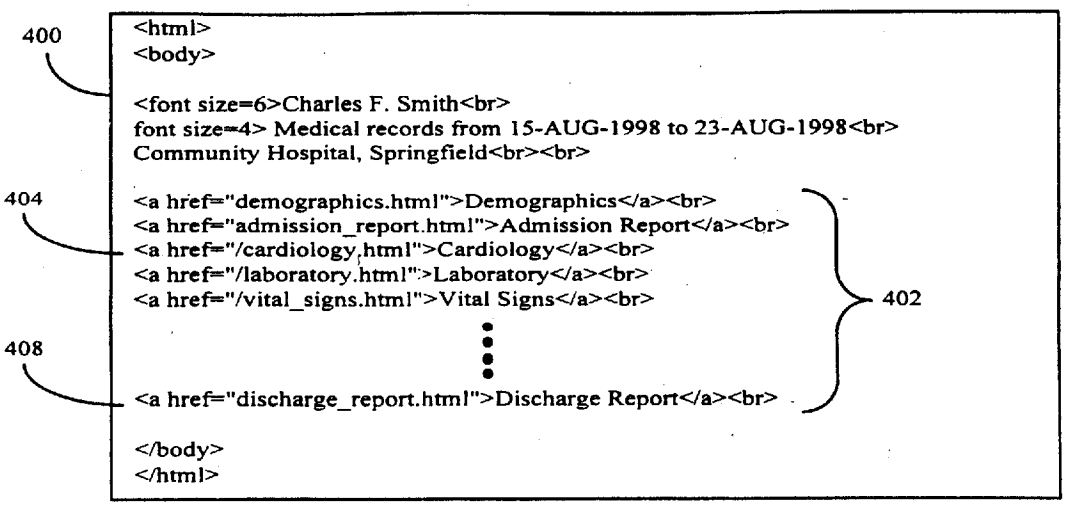


Figure 6A

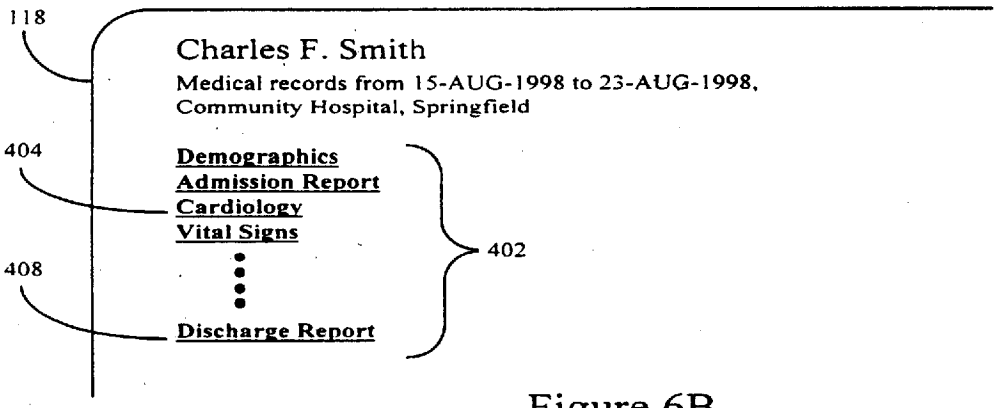


Figure 6B

412

```
<html>
<body>
DISCHARGE SUMMARY for Charles F. Smith<br><br>

Date of Admission: 15-AUG-98<br>
Date of Discharge: 17-AUG-98<br><br>

HISTORY OF PRESENT ILLNESS:<br>
Patient #1 is a 47-year-old male admitted for further evaluation of recent exertional
angina and abnormal exercise test.<br><br>

416
HOSPITAL COURSE:<br>
The patient was admitted to the Telemetry unit and underwent <a
href="charles_f_smith/cardiology/cath/1998-08-15/10:25/report.html">cardiac
catheterization on August 15, 1998</a>. Catheterization demonstrated normal
ventricular function without evidence for prior infarction. The coronary arteriogram
showed moderate stenosis throughout the mid and distal portions of the left anterior
descending artery and diagonal branch, as well as ....
```

Figure 7A

416

118

DISCHARGE SUMMARY for Charles F. Smith

Date of Admission: 15-AUG-98
Date of Discharge: 17-AUG-98

HISTORY OF PRESENT ILLNESS:
Patient #1 is a 47-year-old male admitted for further evaluation of recent exertional angina and abnormal exercise test.

HOSPITAL COURSE:
The patient was admitted to the Telemetry unit and underwent **catheterization on August 15, 1998**. Catheterization demonstrated normal ventricular function without evidence for prior infarction. The coronary arteriogram showed moderate stenosis throughout the mid and distal portions of the left anterior descending artery and diagonal branch, as well as ....

Figure 7B

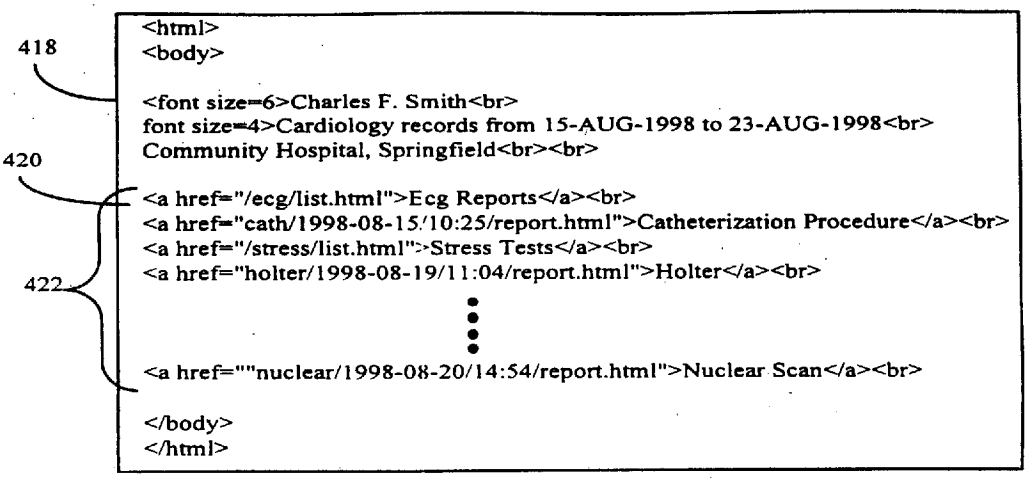


Figure 8A

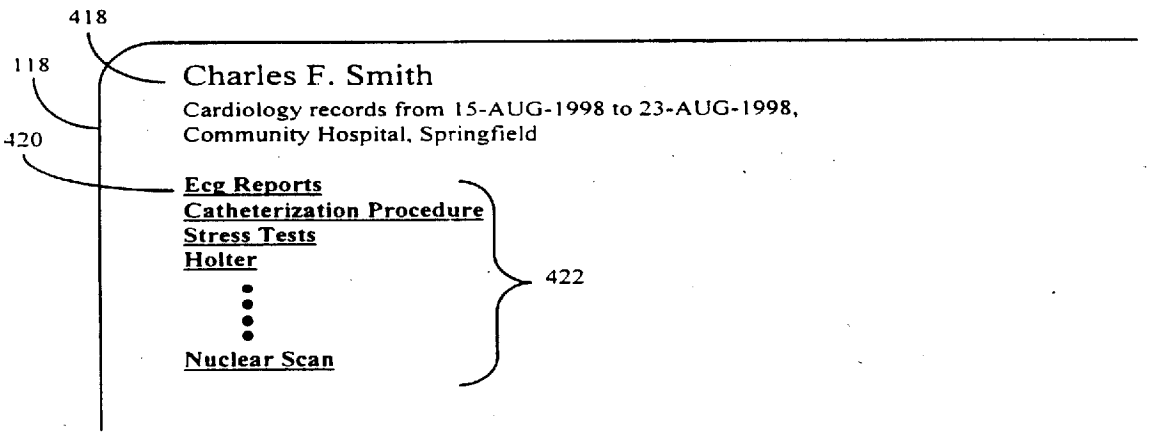


Figure 8B

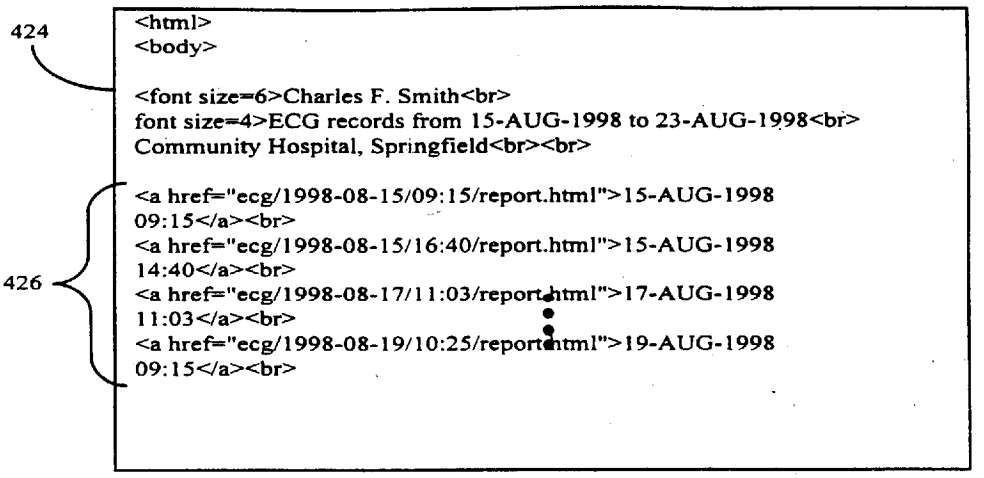


Figure 9A

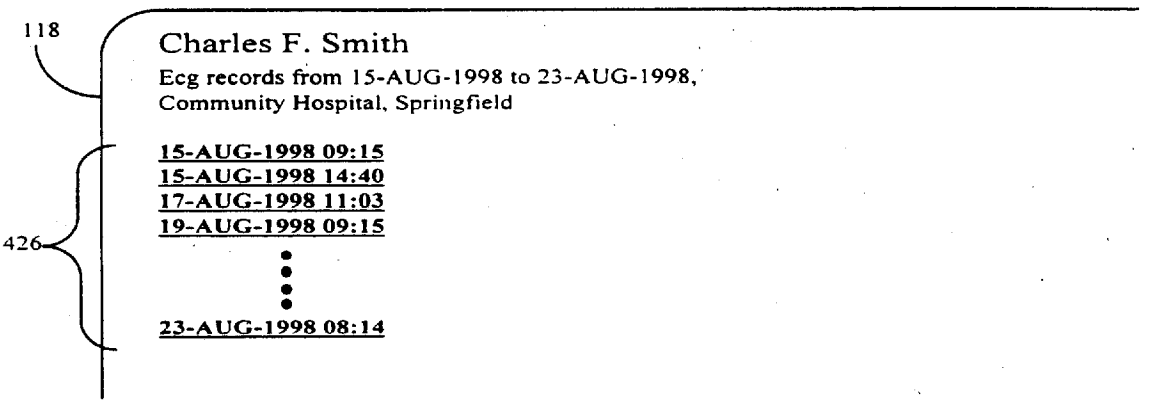


Figure 9B

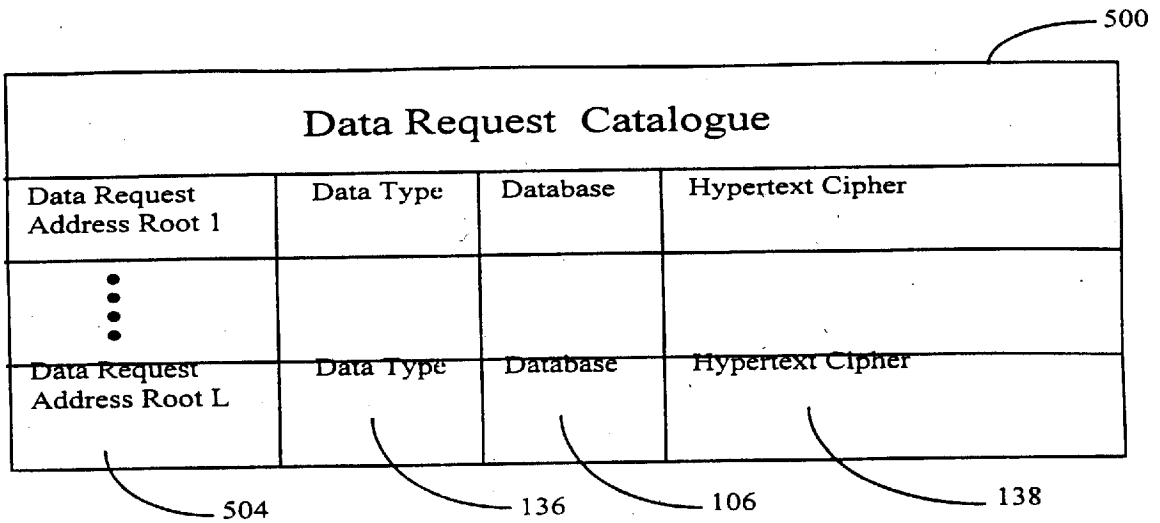


Figure 10

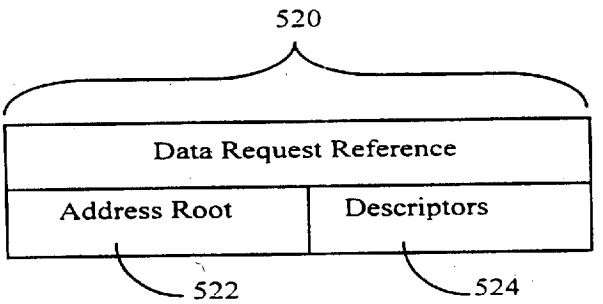


Figure 11

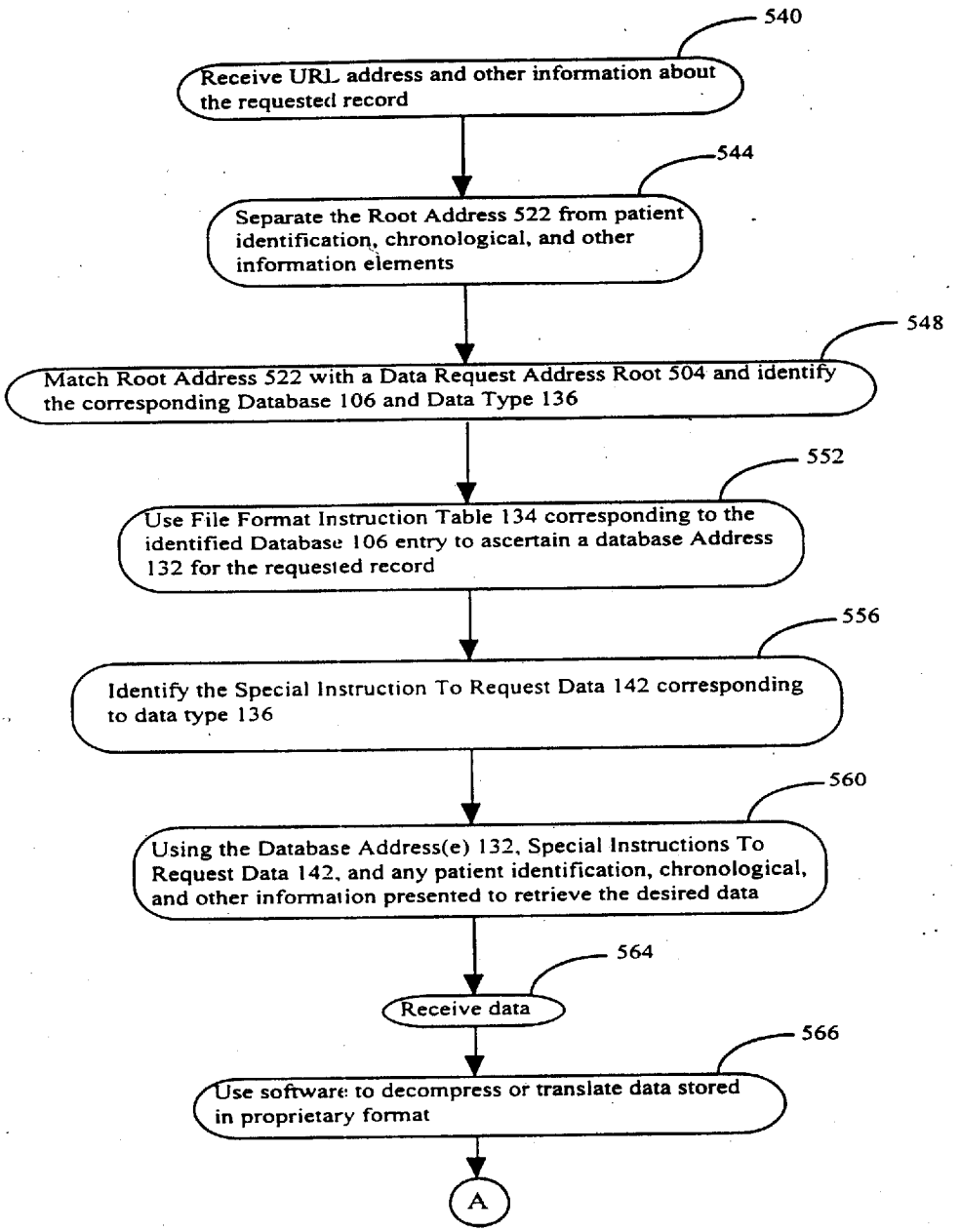

540
Receive URL address and other information about the requested record
544
Separate the Root Address 522 from patient identification, chronological, and other information elements
548
Match Root Address 522 with a Data Request Address Root 504 and identify the corresponding Database 106 and Data Type 136
552
Use File Format Instruction Table 134 corresponding to the identified Database 106 entry to ascertain a database Address 132 for the requested record
556
Identify the Special Instruction To Request Data 142 corresponding to data type 136
560
Using the Database Address(e) 132, Special Instructions To Request Data 142, and any patient identification, chronological, and other information presented to retrieve the desired data
564
Receive data
566
Use software to decompress or translate data stored in proprietary format
A

Figure 12A

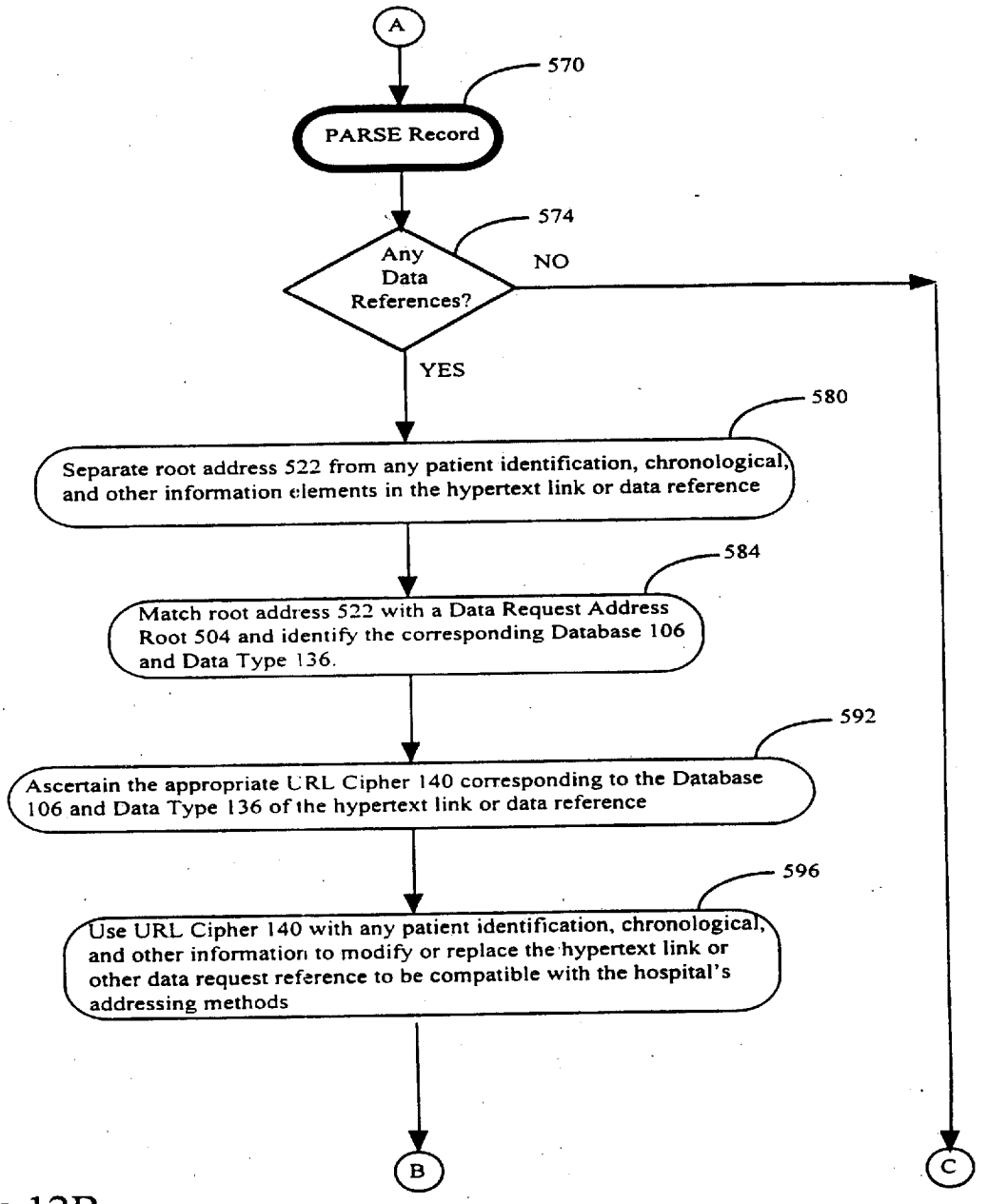

A
570
PARSE Record
574
Any Data References?
NO
YES
580
Separate root address 522 from any patient identification, chronological, and other information elements in the hypertext link or data reference
584
Match root address 522 with a Data Request Address Root 504 and identify the corresponding Database 106 and Data Type 136.
592
Ascertain the appropriate URL Cipher 140 corresponding to the Database 106 and Data Type 136 of the hypertext link or data reference
596
Use URL Cipher 140 with any patient identification, chronological, and other information to modify or replace the hypertext link or other data request reference to be compatible with the hospital's addressing methods
B
C

Figure 12B

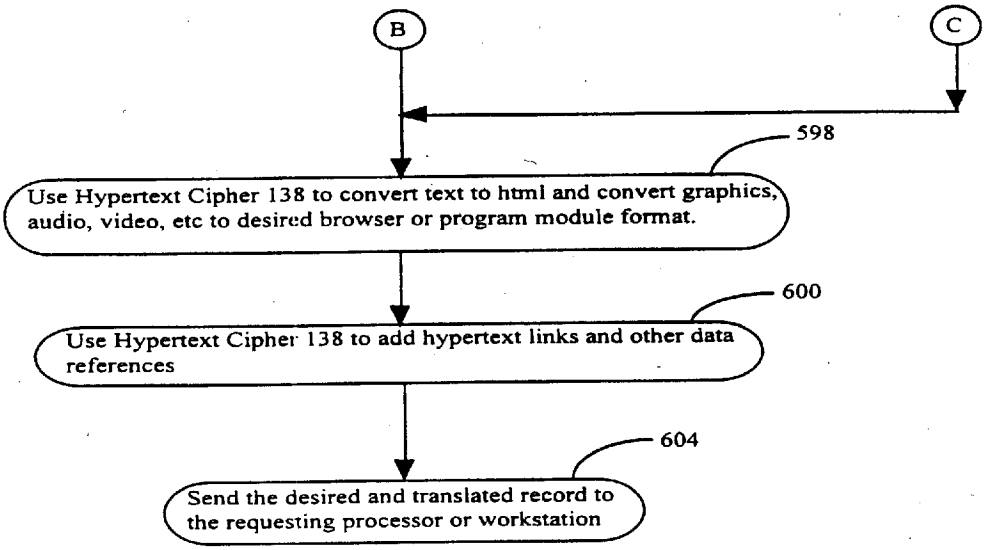

B
C
598
Use Hypertext Cipher 138 to convert text to html and convert graphics, audio, video, etc to desired browser or program module format.
600
Use Hypertext Cipher 138 to add hypertext links and other data references
604
Send the desired and translated record to the requesting processor or workstation

Figure 12C

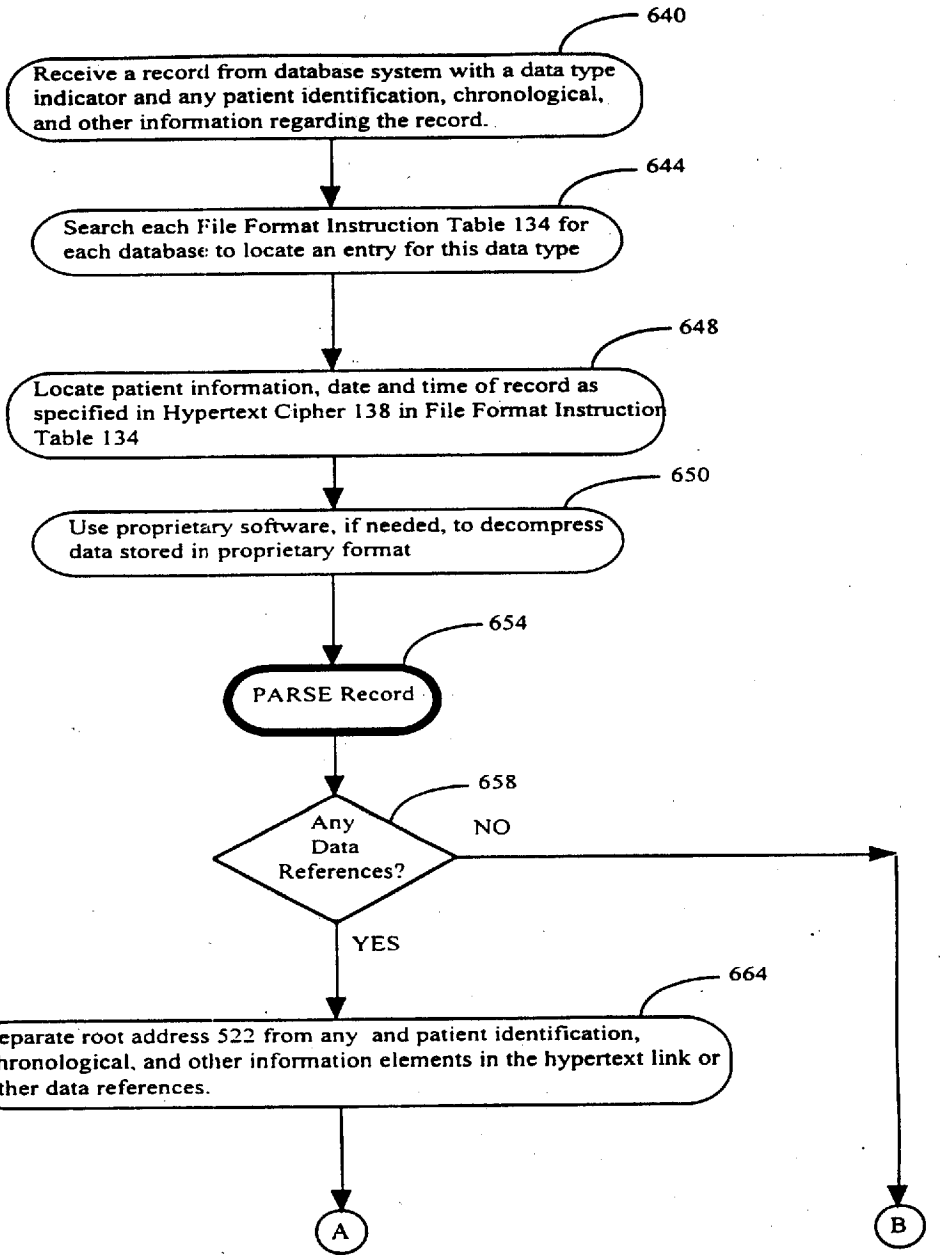


Figure 13A

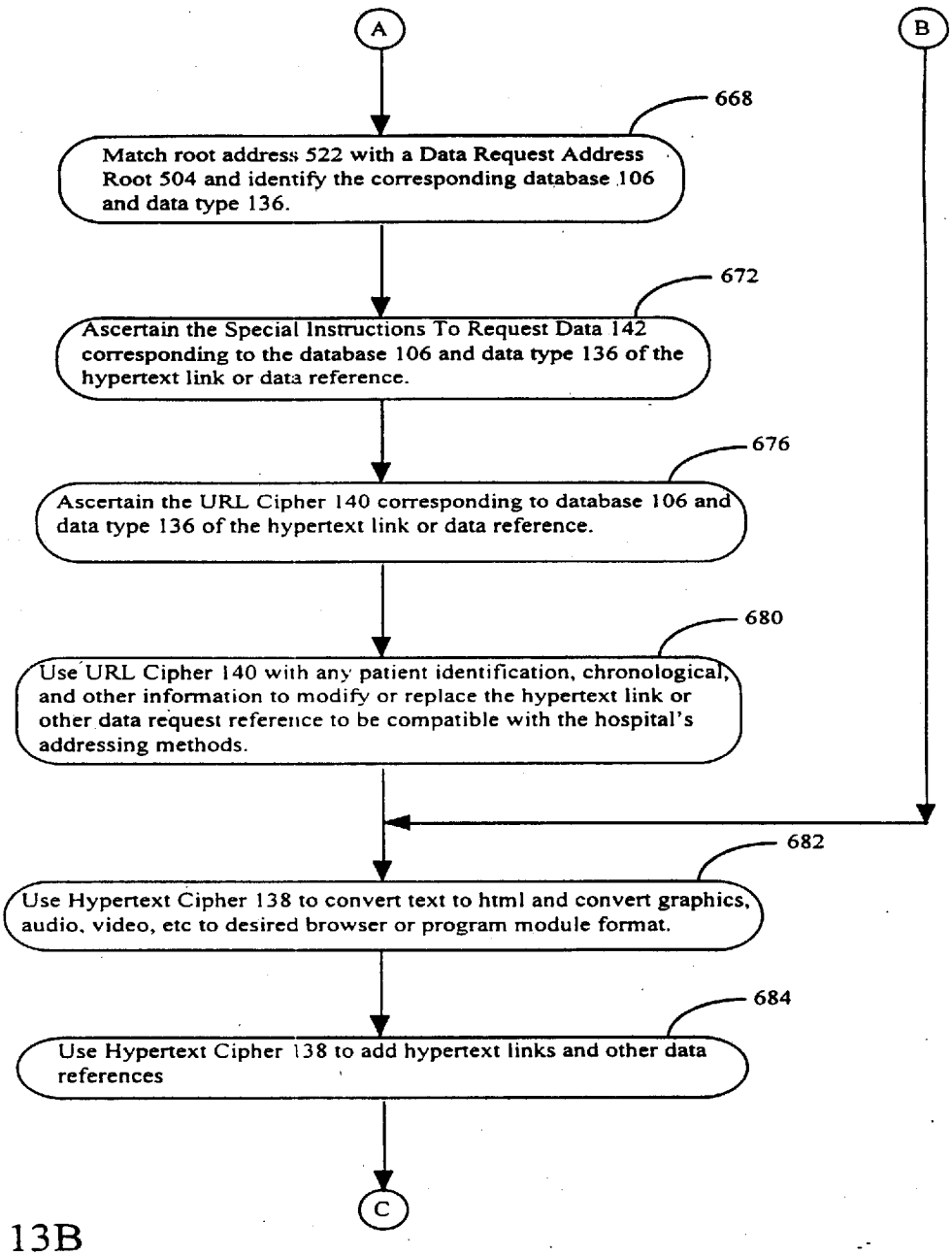

A
B
668
Match root address 522 with a Data Request Address Root 504 and identify the corresponding database 106 and data type 136.
672
Ascertain the Special Instructions To Request Data 142 corresponding to the database 106 and data type 136 of the hypertext link or data reference.
676
Ascertain the URL Cipher 140 corresponding to database 106 and data type 136 of the hypertext link or data reference.
680
Use URL Cipher 140 with any patient identification, chronological, and other information to modify or replace the hypertext link or other data request reference to be compatible with the hospital's addressing methods.
682
Use Hypertext Cipher 138 to convert text to html and convert graphics, audio, video, etc to desired browser or program module format.
684
Use Hypertext Cipher 138 to add hypertext links and other data references
C

Figure 13B

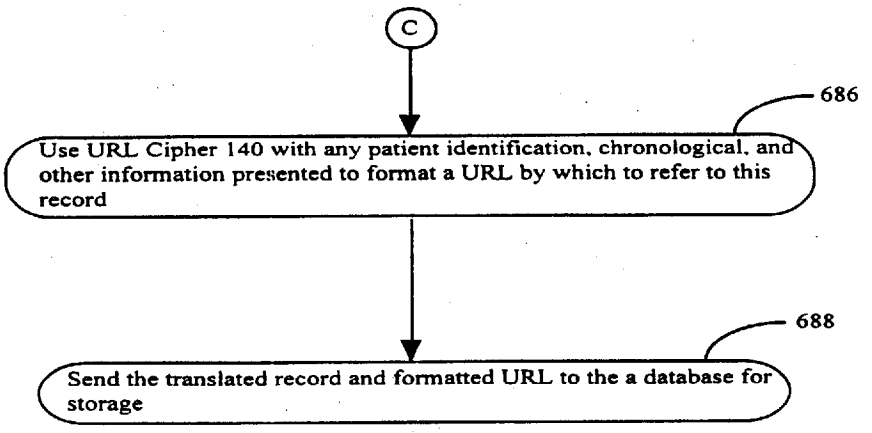


Figure 13C

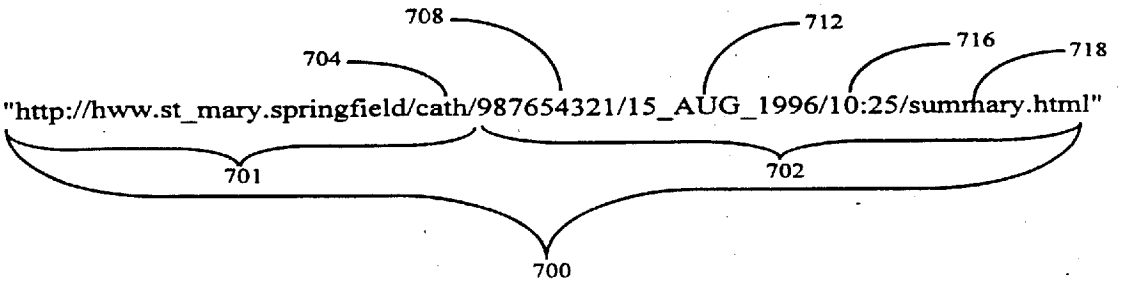


Figure 14A

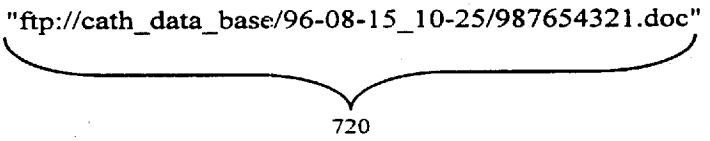


Figure 14B

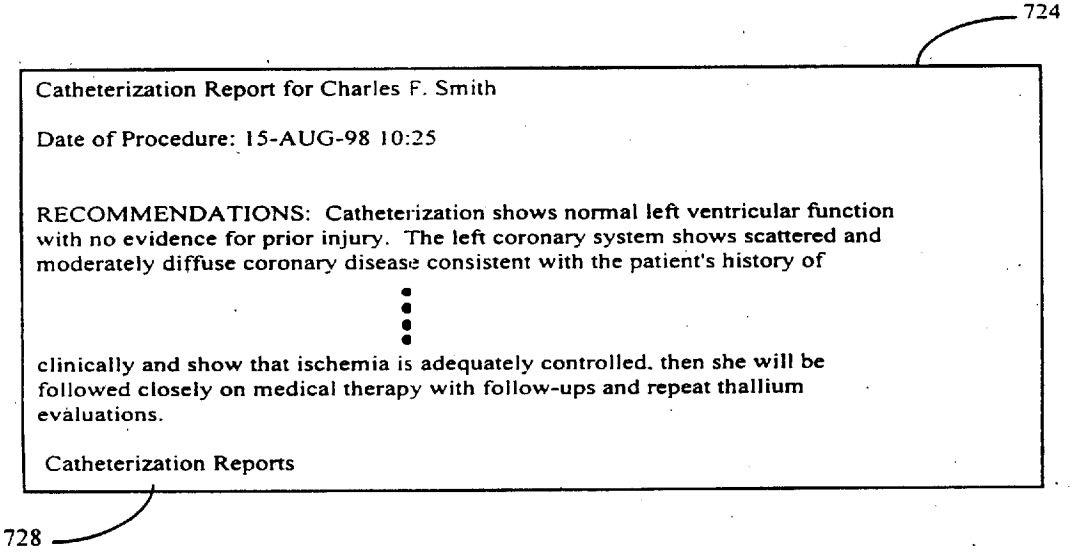


Figure 14C

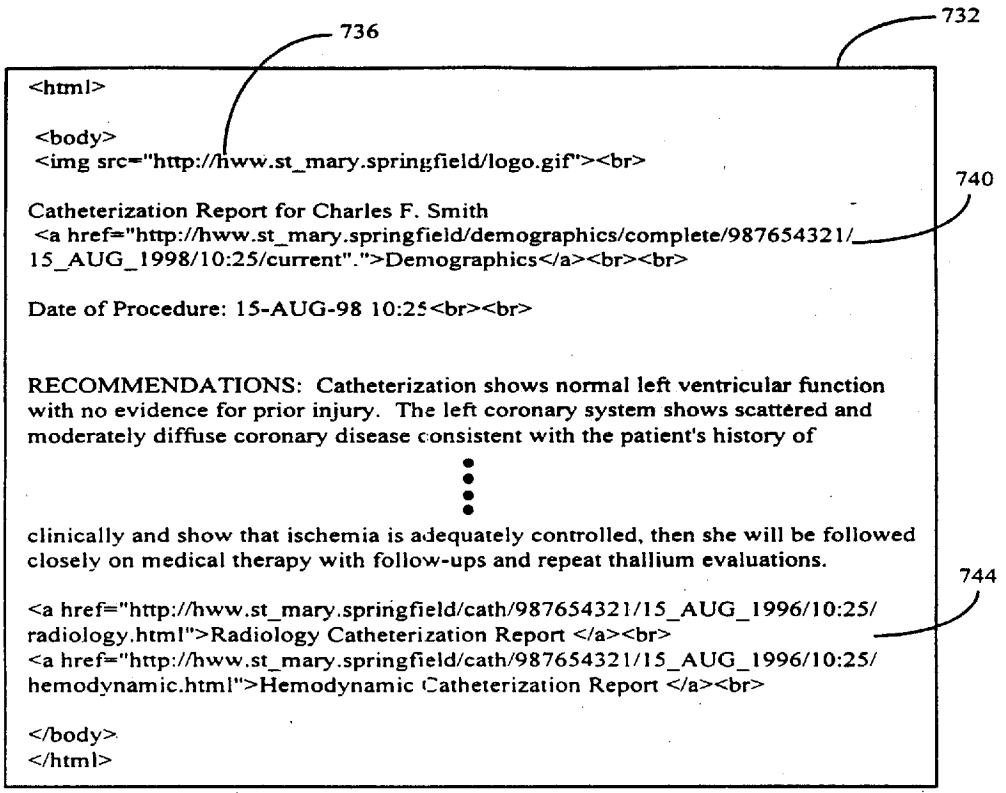


Figure 14D

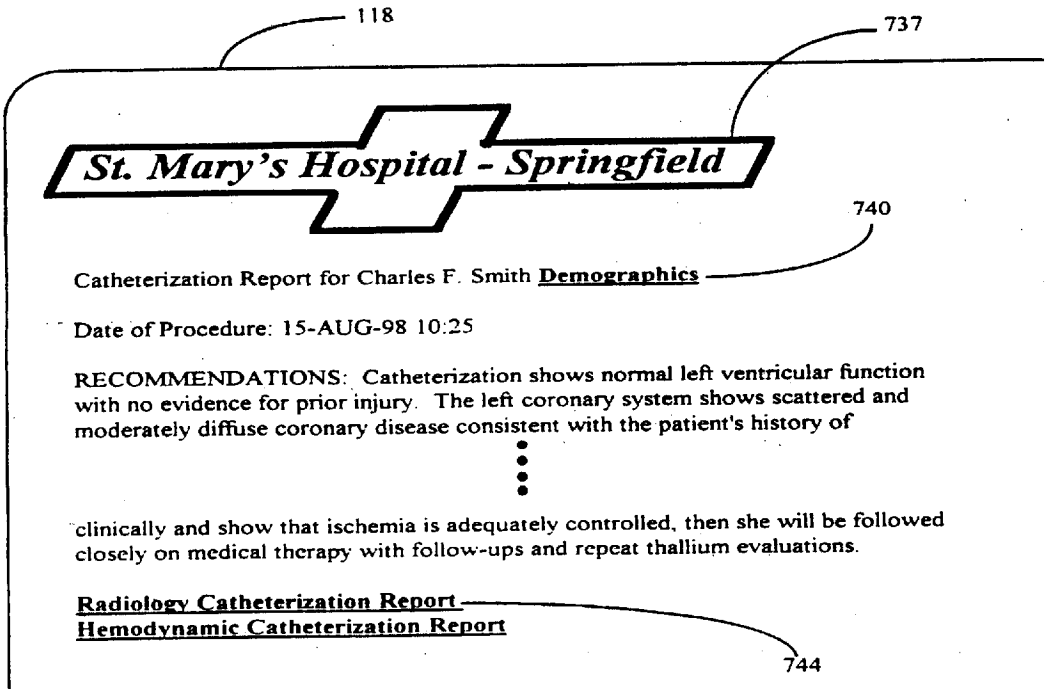


Figure 14E

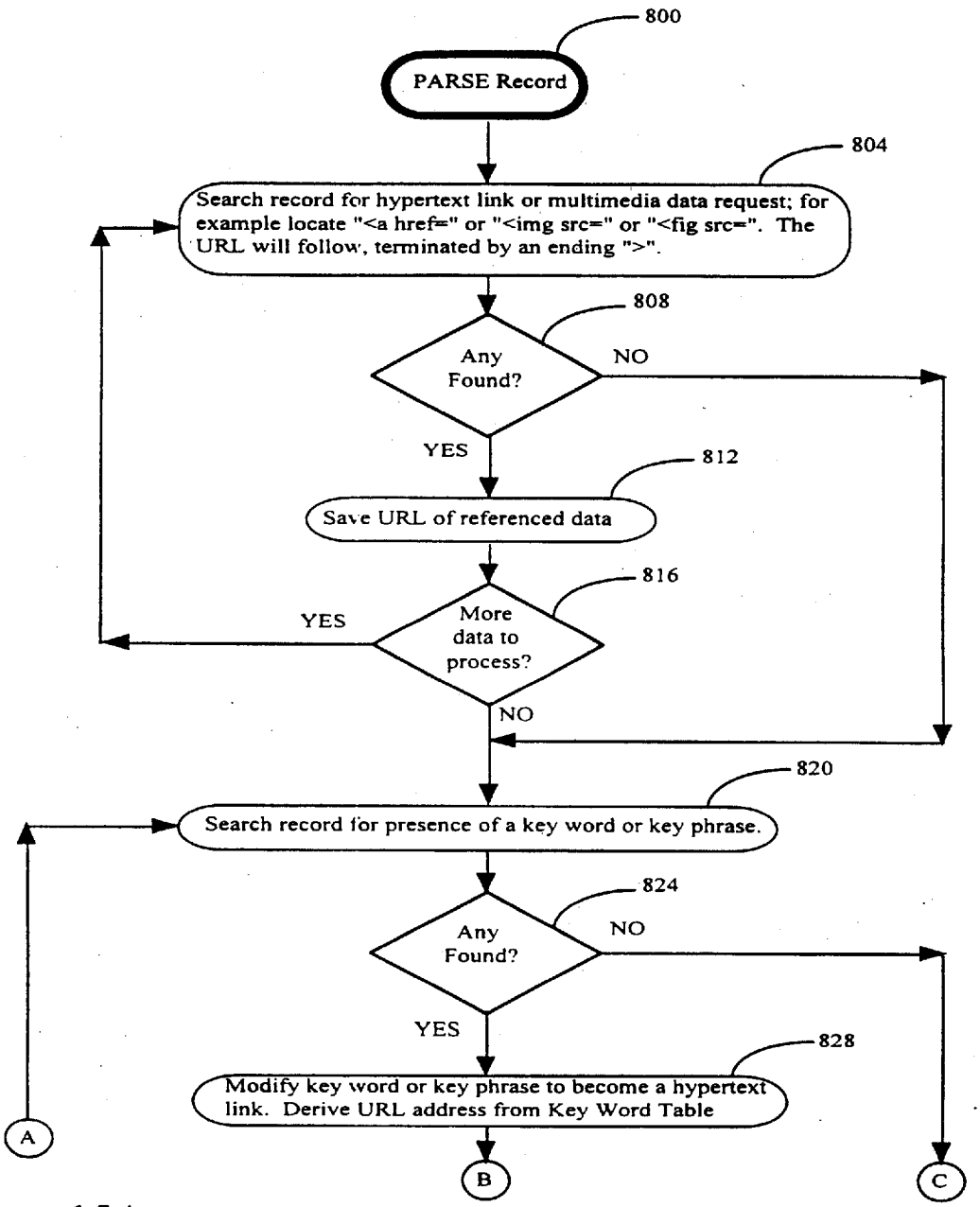


Figure 15A

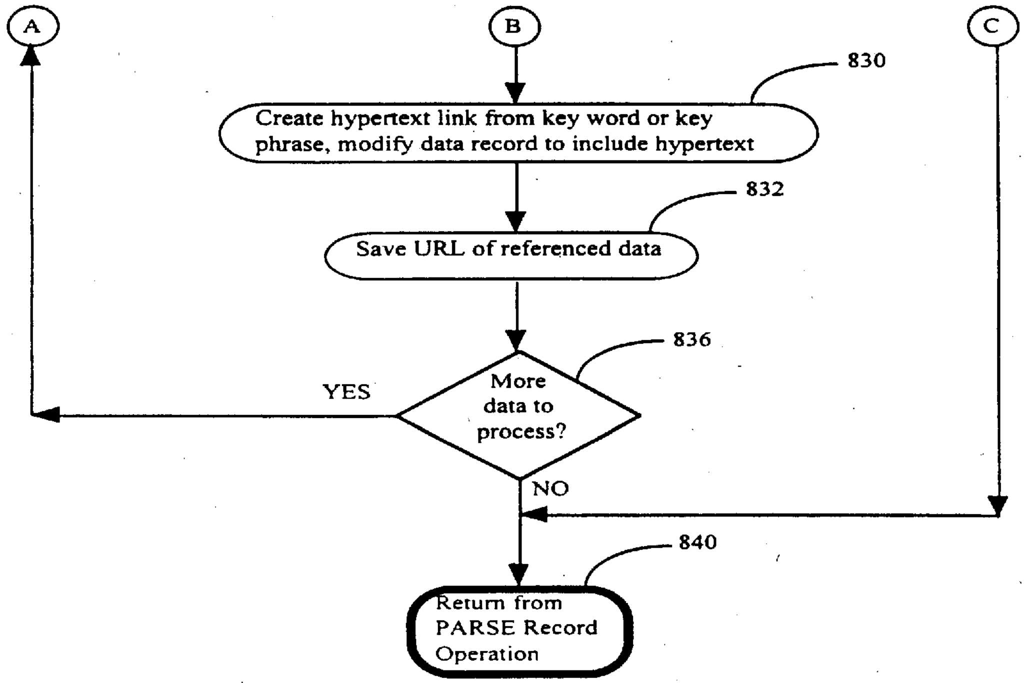


Figure 15B

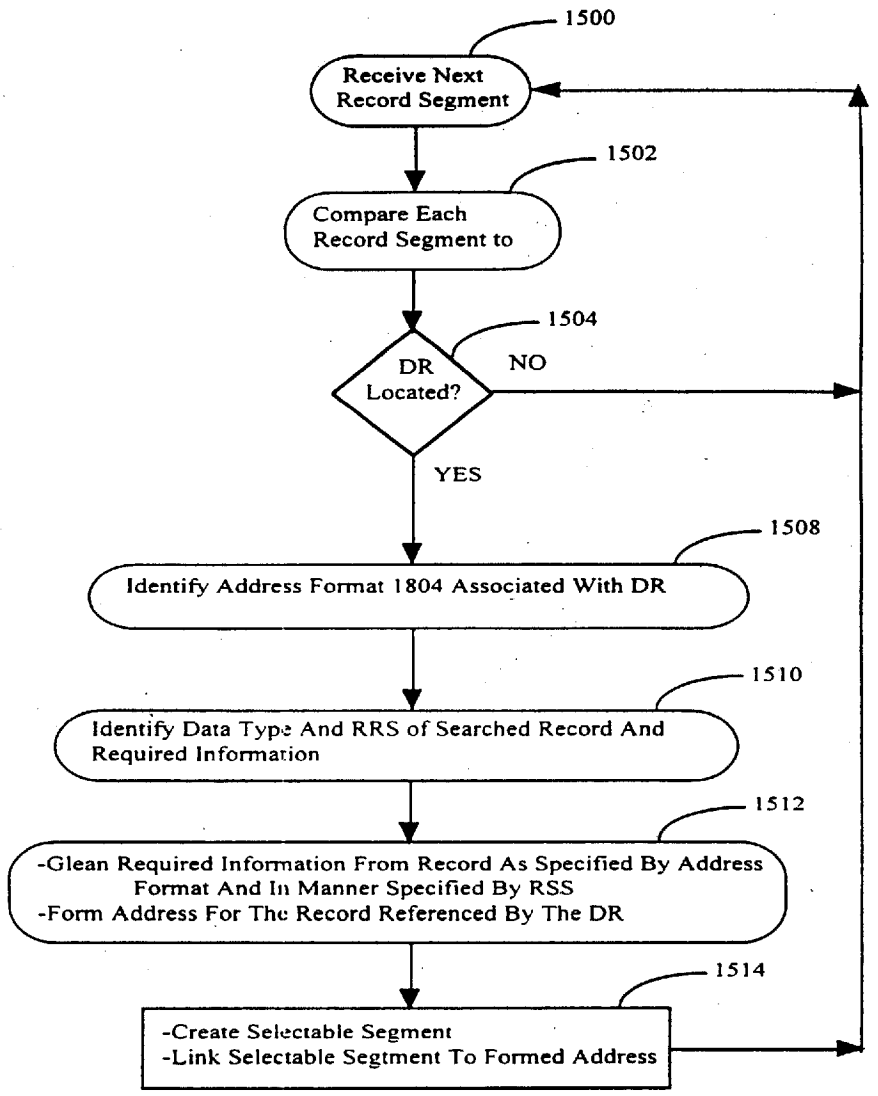

1500
Receive Next Record Segment
1502
Compare Each Record Segment to
1504
DR Located?
NO
YES
1508
Identify Address Format 1804 Associated With DR
1510
Identify Data Type And RRS of Searched Record And Required Information
1512
-Glean Required Information From Record As Specified By Address Format And In Manner Specified By RSS
-Form Address For The Record Referenced By The DR
1514
-Create Selectable Segment
-Link Selectable Segtment To Formed Address

Figure 16

Text As It Appears in the Word Processor

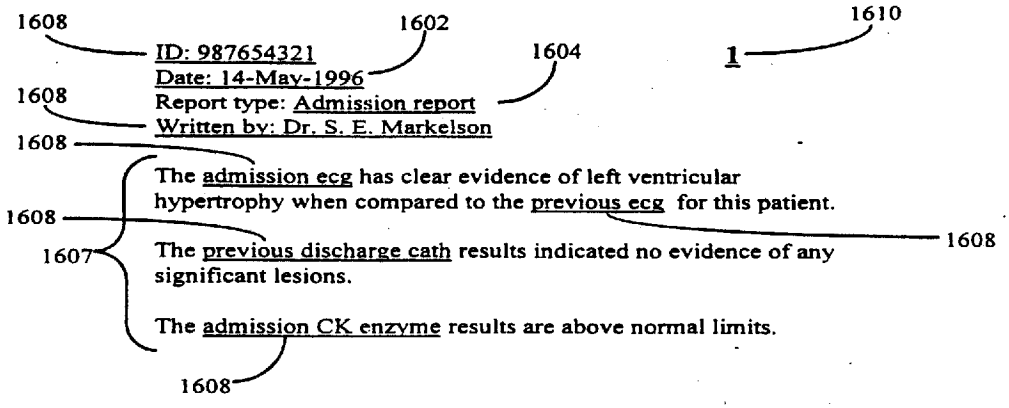


Figure 17

Text after Being Converted to HTML with Hypertext Links Added

```
<html>
<body>                                                                    1702
<a href= "http://hww.st_mary.springfield/demographics/987654321/14_May_1996">
ID: 987654321</a><br>
Date: 14-May-1996<br>                                                     1702
Report type: Admission report
Written by:  <a href="http://hww.st_mary.springfield/staff_directory/S._E._Markelson">
Dr. S. E. Markelson</a><br>                                               1700
<br>
1608
The <a href="http://hww.st_mary.springfield/ecg/987654321/14_May_1996/00:00/admit">
admission ecg  </a> has clear evidence of left ventricular hypertrophy when compared to the
1608
 <a href="http://hww.st_mary.springfield/ecg/987654321/14_May_1996/00:00/current">
previous ecg  </a> for this patient.<br>
<br>                                                                      1702
The
<a
href="http://hww.st_mary.springfield/cath/987654321/14_May_1996/00:00/prev_discharge">
previous discharge cath </a> results indicated no evidence of any significant occlusions.<br>
<br>
The
<a href="http://hww.st_mary.springfield/lab_CK_enz/987654321/14_May_1996/00:00/admit">
admission CK enzyme </a>results are above normal limits.<br>

</body>
</html>
```

Figure 18

Text As Viewed via Word Processor 14 or Browser

ID: 987654321
Date: 14-May-1996
Report type: Admission report
Written by: Dr. S. E. Markelson

The admission ecg has clear evidence of left ventricular hypertrophy when compared to the current ecg for this patient.

The previous discharge cath results indicated no evidence of any significant lesions.

The admission CK enzyme results are above normal limits.

Figure 19

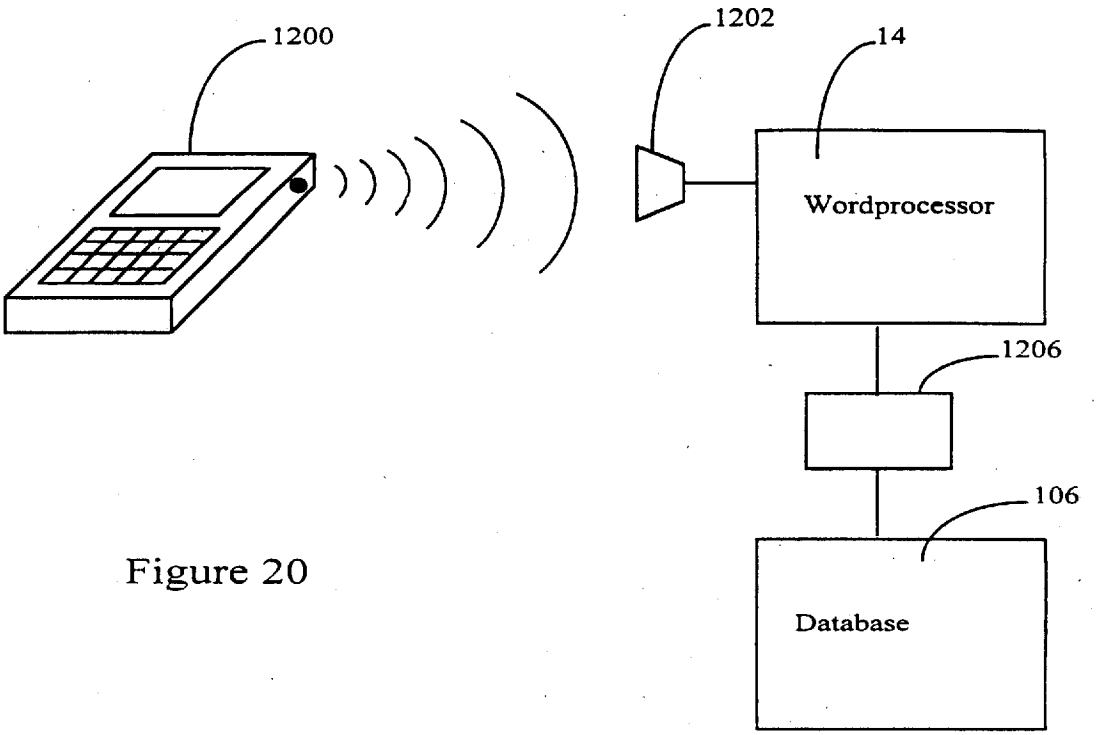


Figure 20

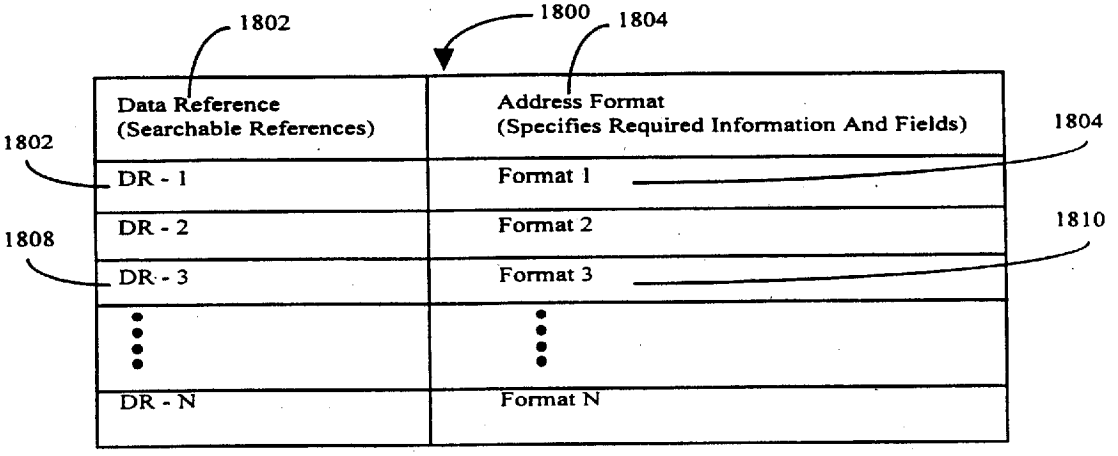


Figure 21

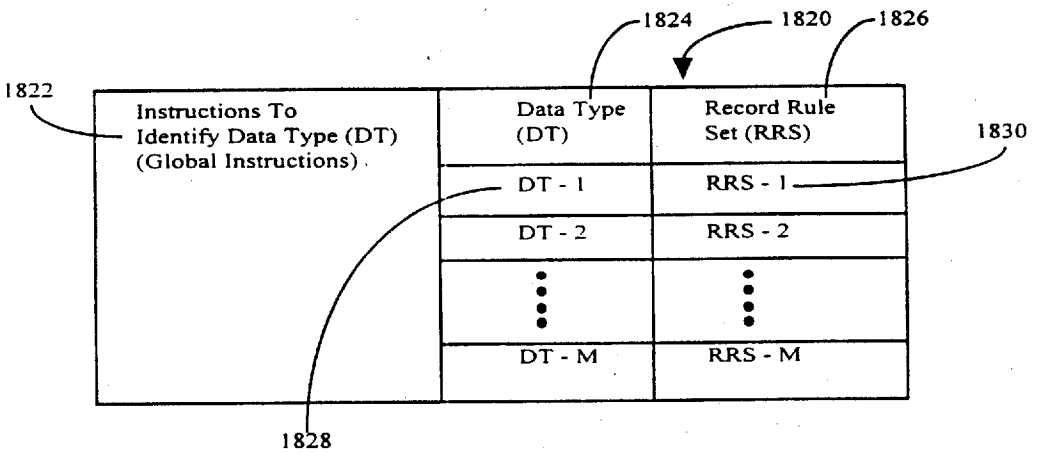


Figure 22

1902 1900 1904 1906

Data Reference (Searchable References)	Address Format	Record Rule Set (RRS)
DR - 1	Format 1	RRS - 1
DR - 2	Format 2	RRS- 2
DR - 3	Format 3	RRS -3
⋮	⋮	
DR - N	Format N	RRS - N

Figure 23

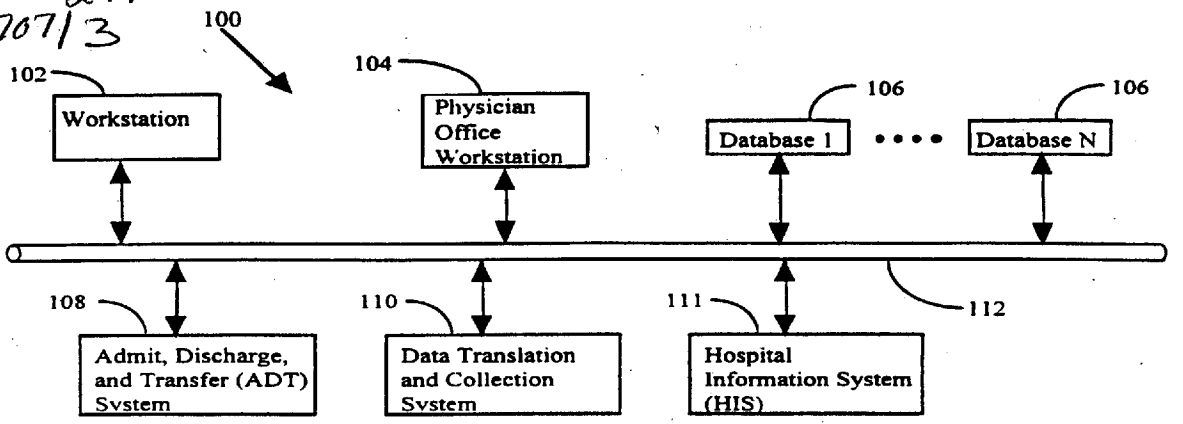


Figure 1

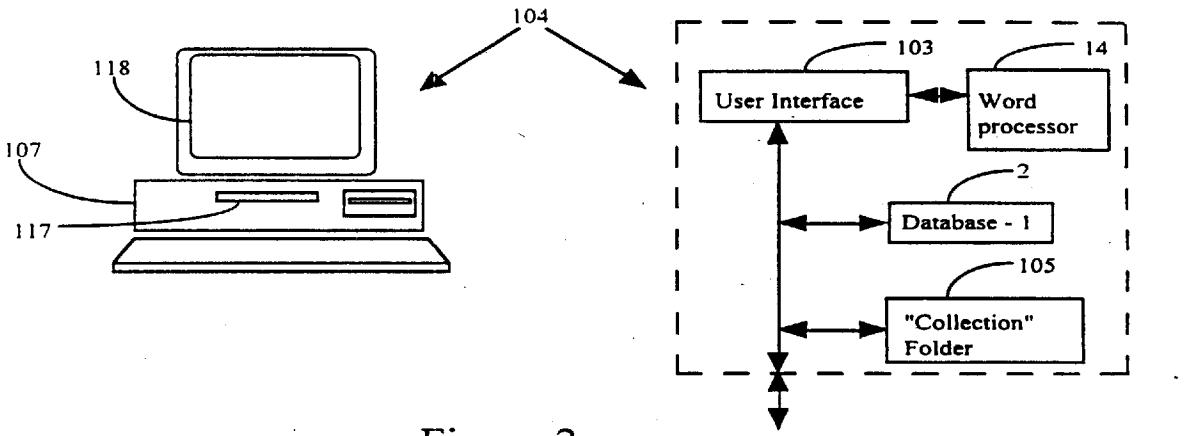


Figure 2

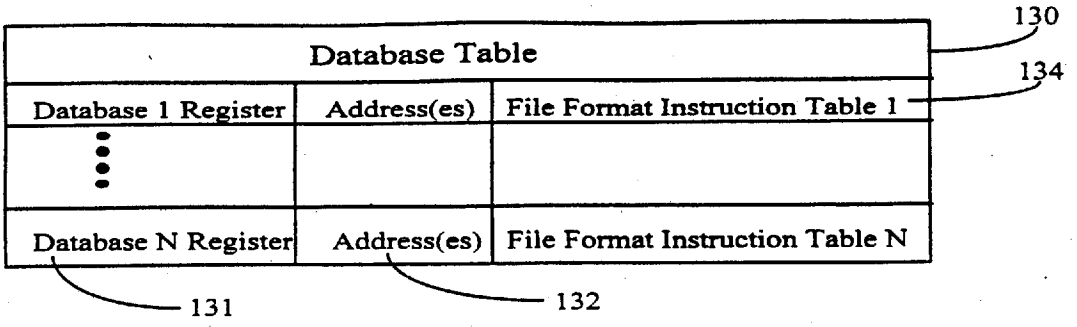


Figure 3A

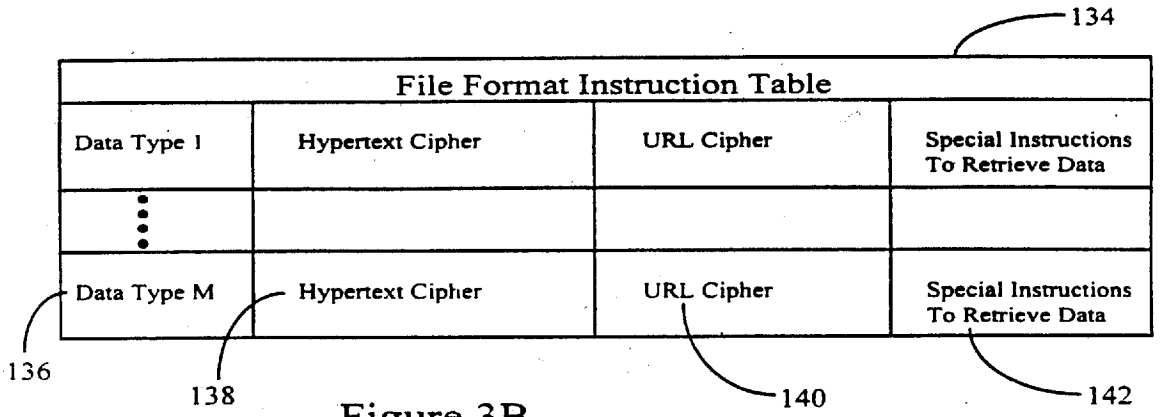


Figure 3B

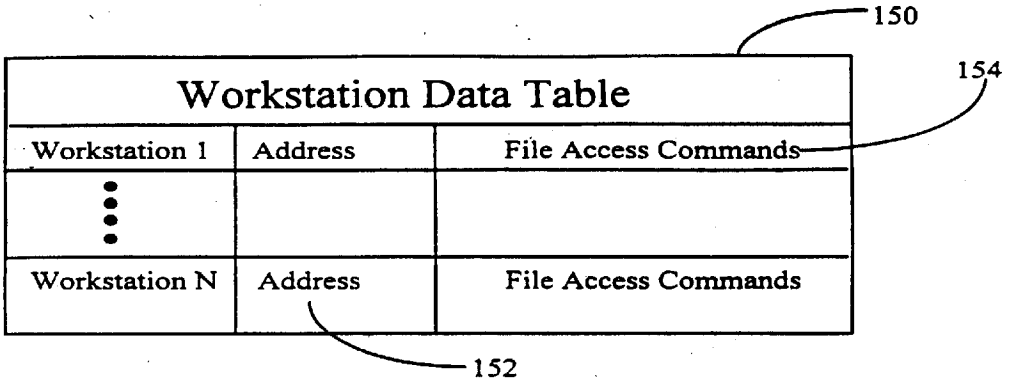


Figure 4A

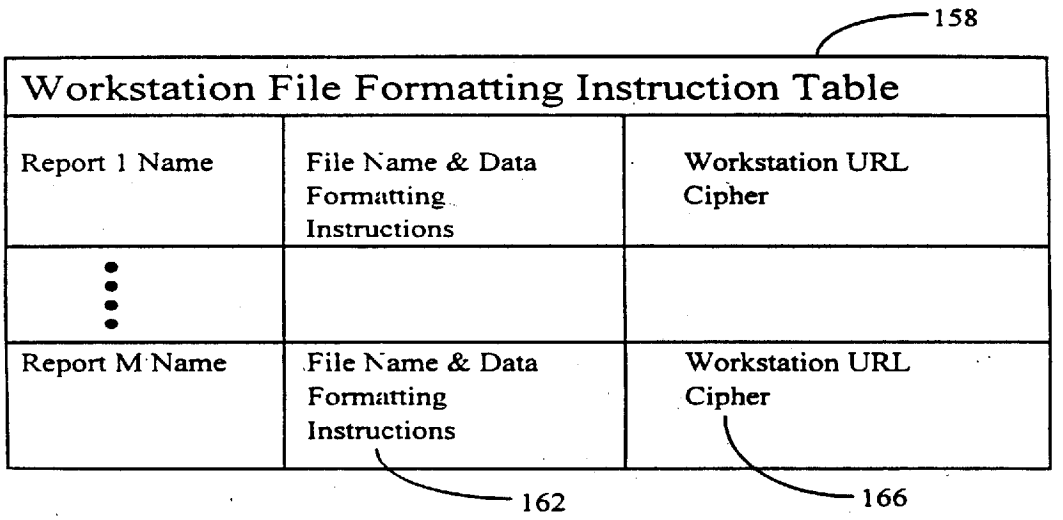


Figure 4B

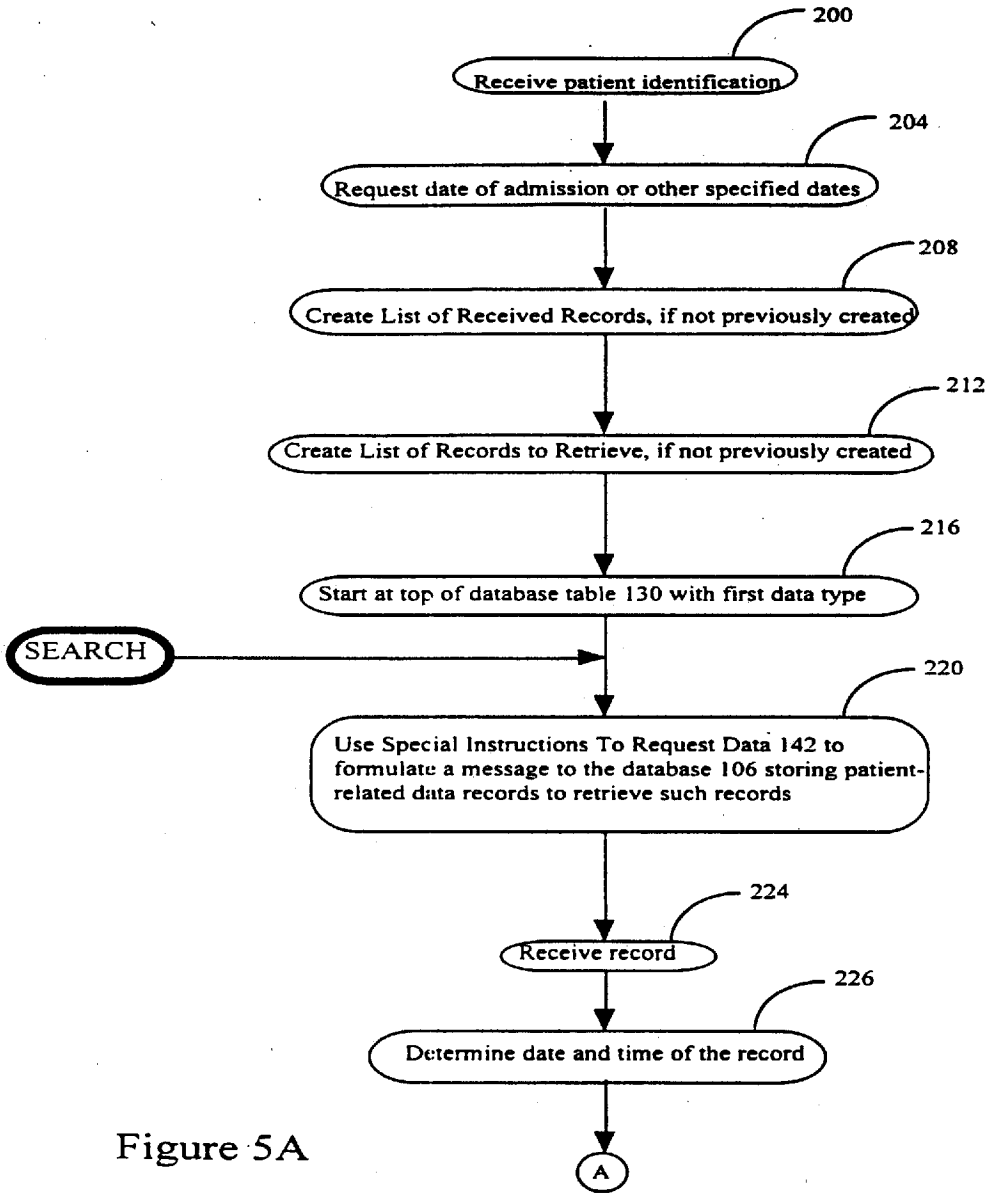

200
Receive patient identification
204
Request date of admission or other specified dates
208
Create List of Received Records, if not previously created
212
Create List of Records to Retrieve, if not previously created
216
Start at top of database table 130 with first data type
SEARCH
220
Use Special Instructions To Request Data 142 to formulate a message to the database 106 storing patient-related data records to retrieve such records
224
Receive record
226
Determine date and time of the record
A

Figure 5A

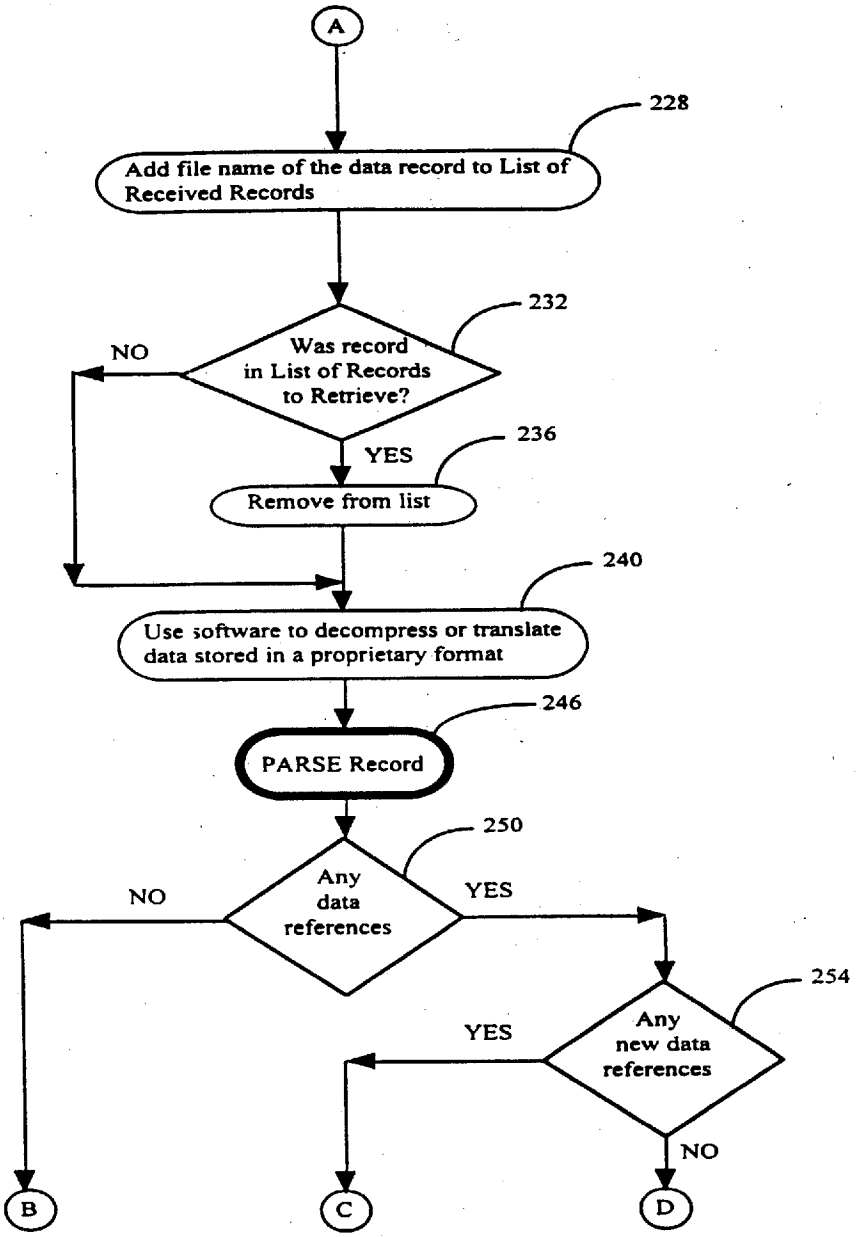

A
228
Add file name of the data record to List of Received Records
232
Was record in List of Records to Retrieve?
NO
YES
236
Remove from list
240
Use software to decompress or translate data stored in a proprietary format
246
PARSE Record
250
Any data references
NO
YES
254
Any new data references
YES
NO
B
C
D

Figure 5B

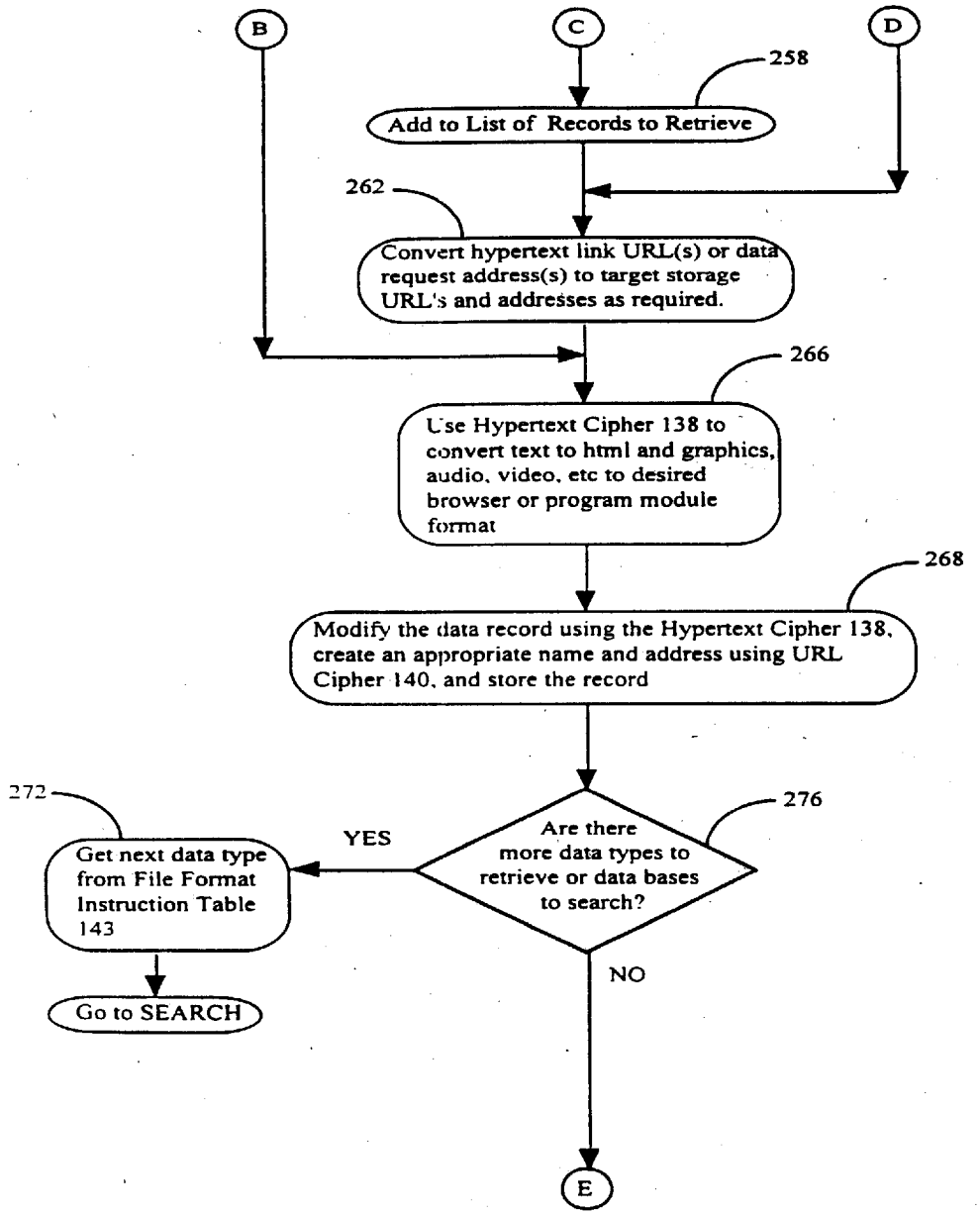


Figure 5C

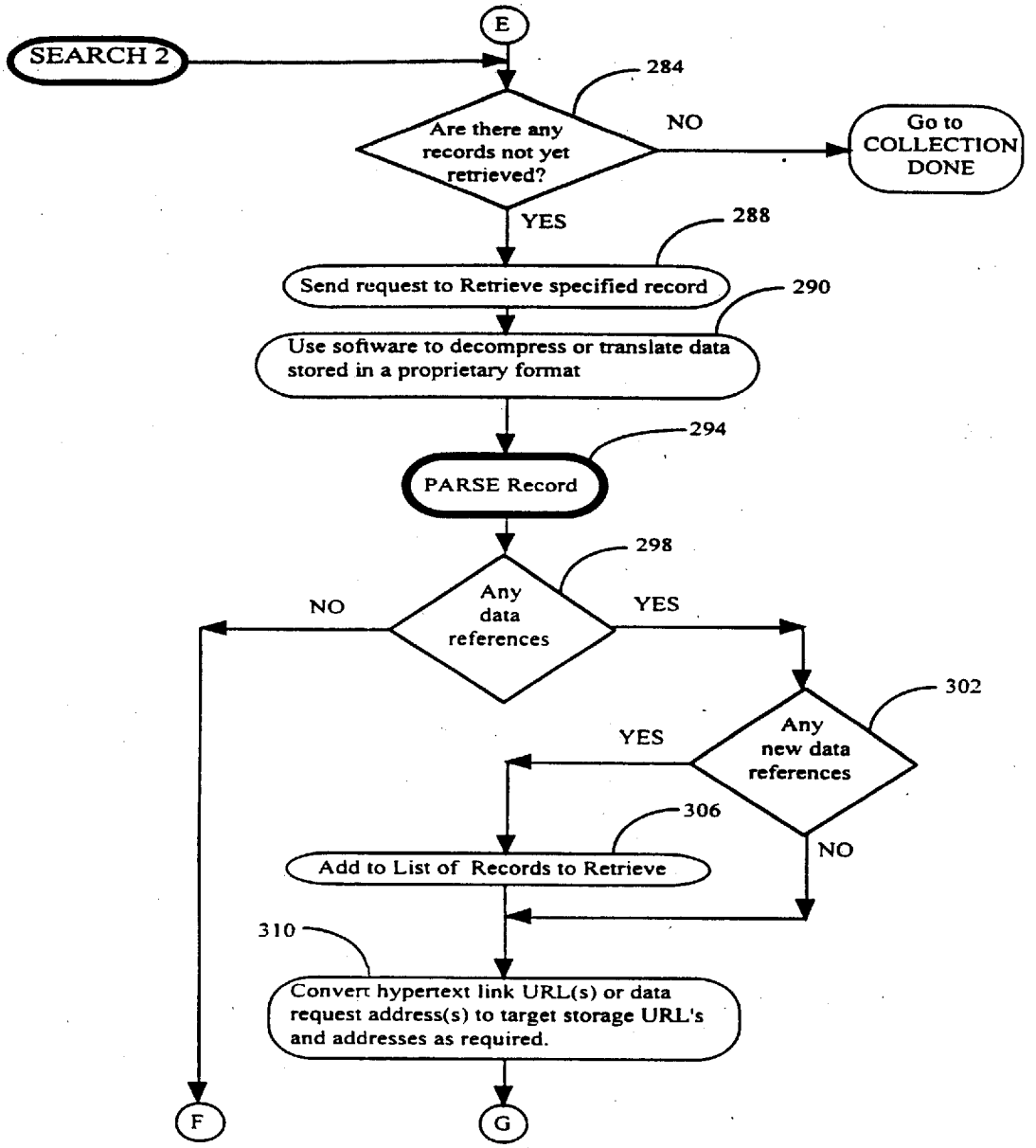


Figure 5D

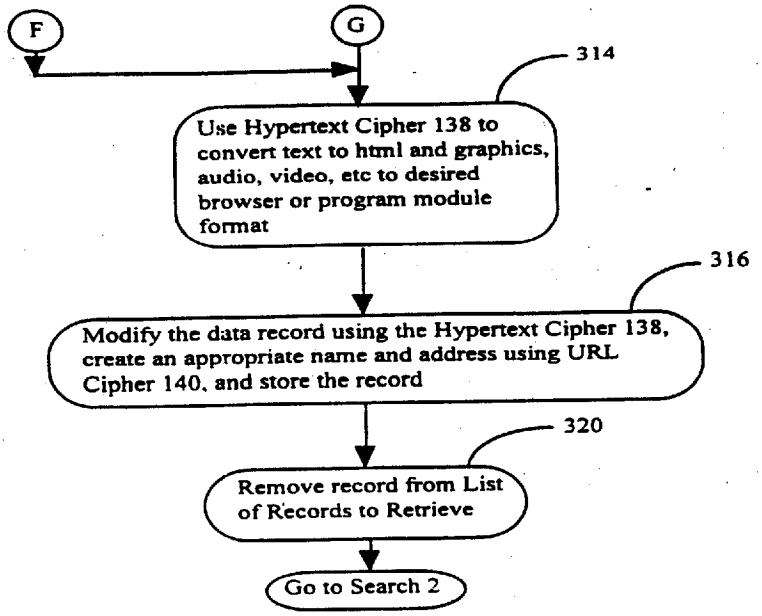


Figure 5E

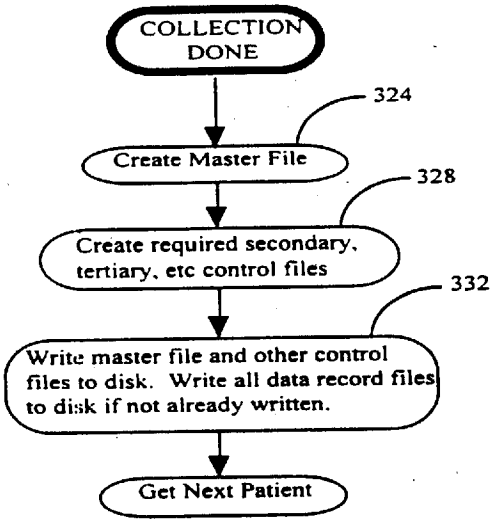


Figure 5F

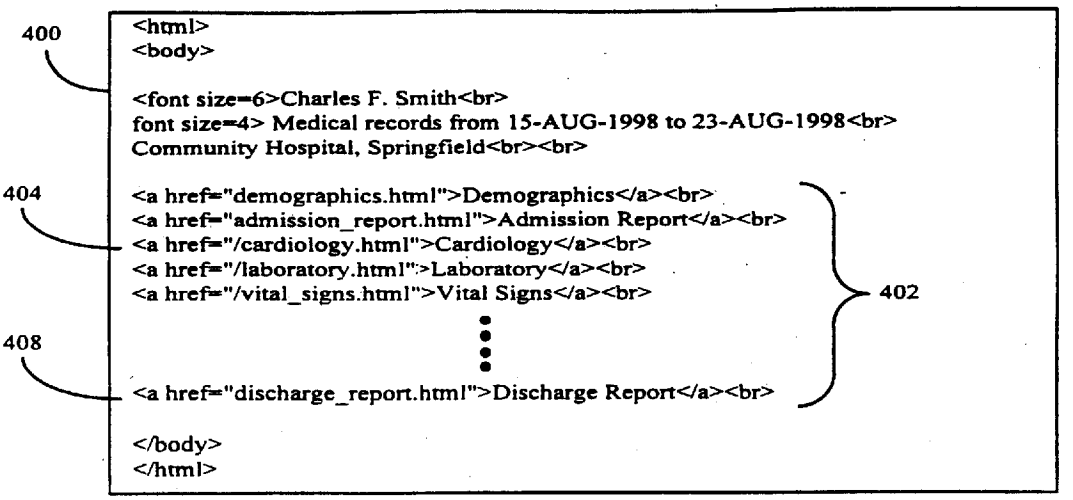


Figure 6A

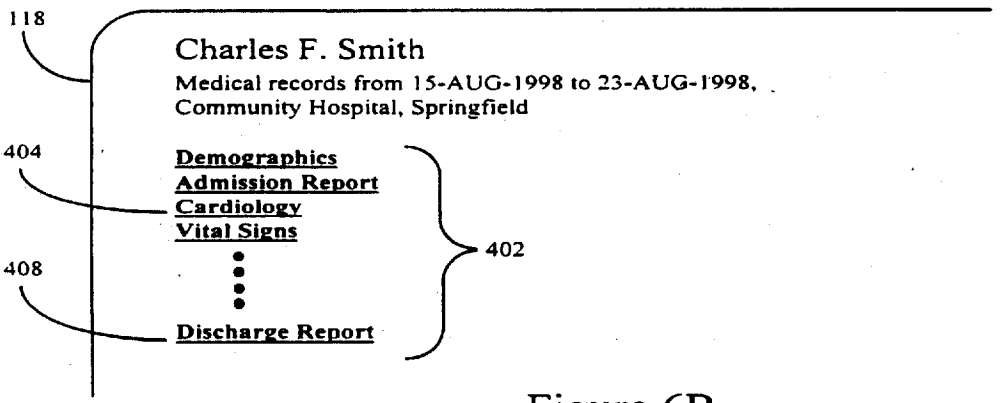


Figure 6B

412

```
<html>
<body>
DISCHARGE SUMMARY for Charles F. Smith<br><br>

Date of Admission: 15-AUG-98<br>
Date of Discharge: 17-AUG-98<br><br>

HISTORY OF PRESENT ILLNESS:<br>
Patient #1 is a 47-year-old male admitted for further evaluation of recent exertional
angina and abnormal exercise test.<br><br>

HOSPITAL COURSE:<br>
The patient was admitted to the Telemetry unit and underwent <a
href="charles_f_smith/cardiology/cath/1998-08-15/10:25/report.html">cardiac
catheterization on August 15, 1998</a>. Catheterization demonstrated normal
ventricular function without evidence for prior infarction. The coronary arteriogram
showed moderate stenosis throughout the mid and distal portions of the left anterior
descending artery and diagonal branch, as well as ••••
```

416

Figure 7A

416

118

DISCHARGE SUMMARY for Charles F. Smith

Date of Admission: 15-AUG-98
Date of Discharge: 17-AUG-98

HISTORY OF PRESENT ILLNESS:
Patient #1 is a 47-year-old male admitted for further evaluation of recent exertional angina and abnormal exercise test.

HOSPITAL COURSE:
The patient was admitted to the Telemetry unit and underwent **catheterization on August 15, 1998**. Catheterization demonstrated normal ventricular function without evidence for prior infarction. The coronary arteriogram showed moderate stenosis throughout the mid and distal portions of the left anterior descending artery and diagonal branch, as well as ••••

Figure 7B

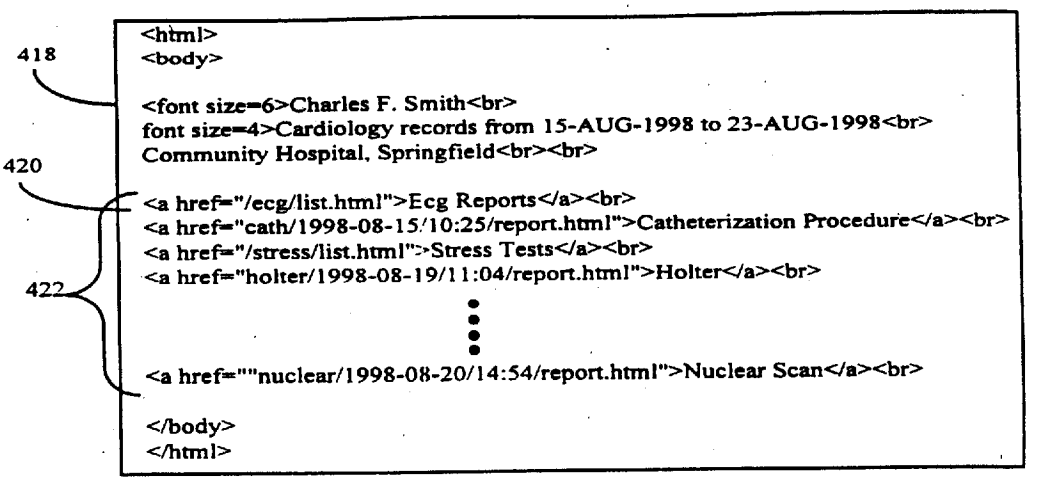


Figure 8A

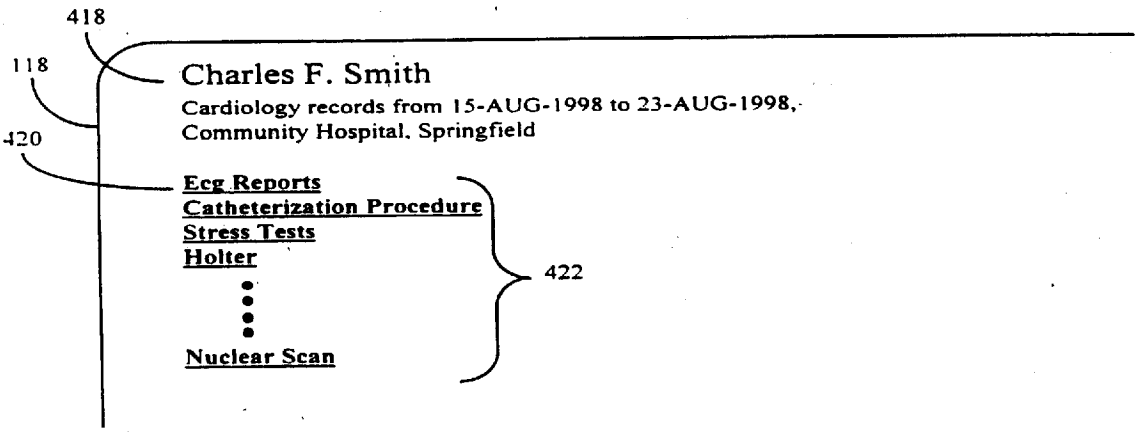


Figure 8B

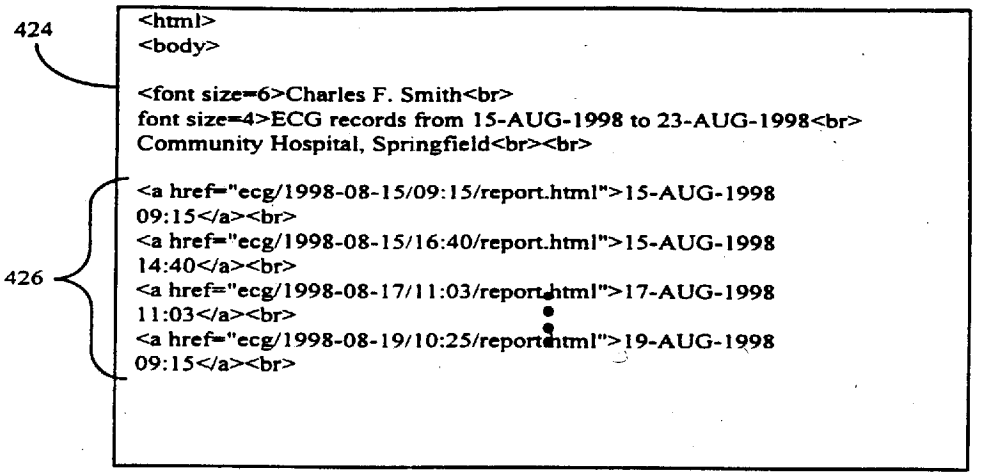


Figure 9A

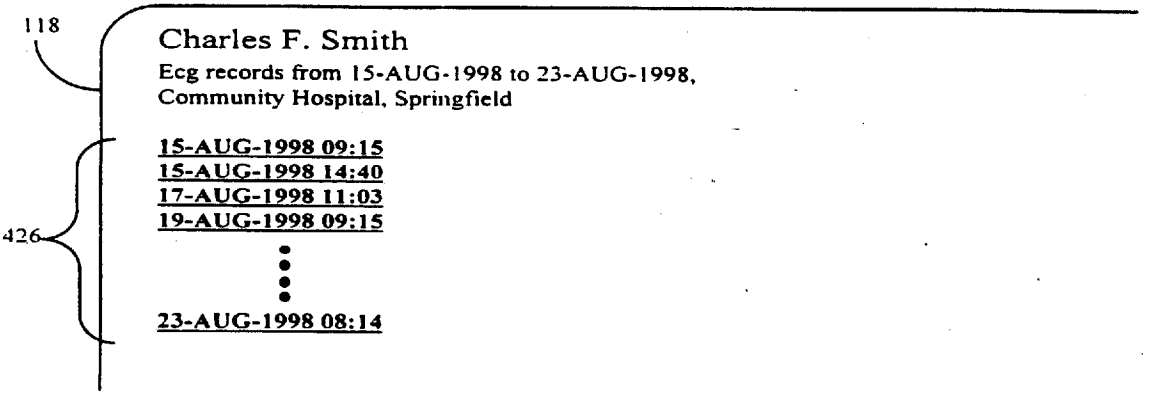


Figure 9B

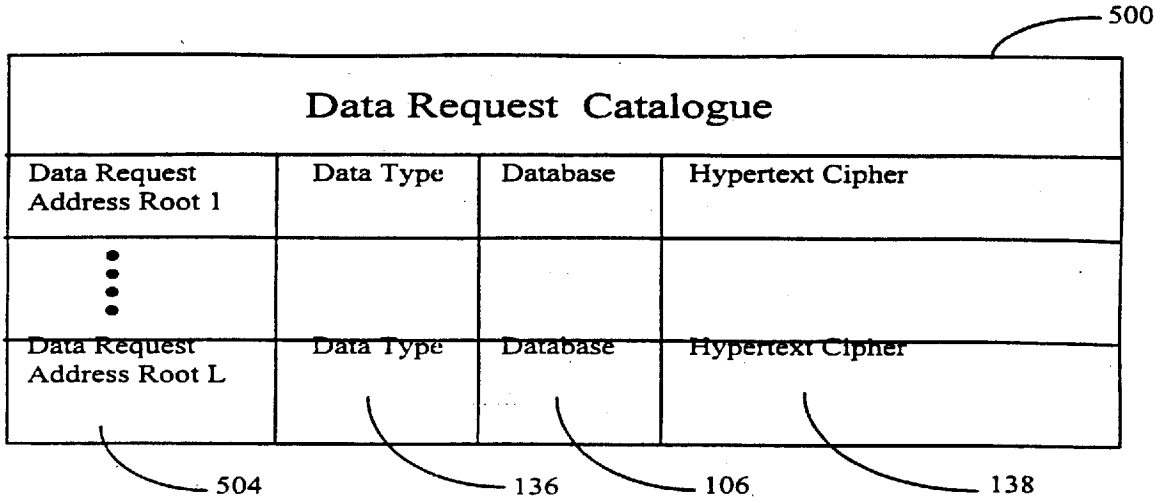


Figure 10

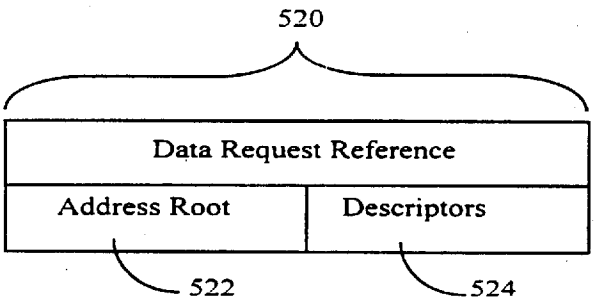


Figure 11

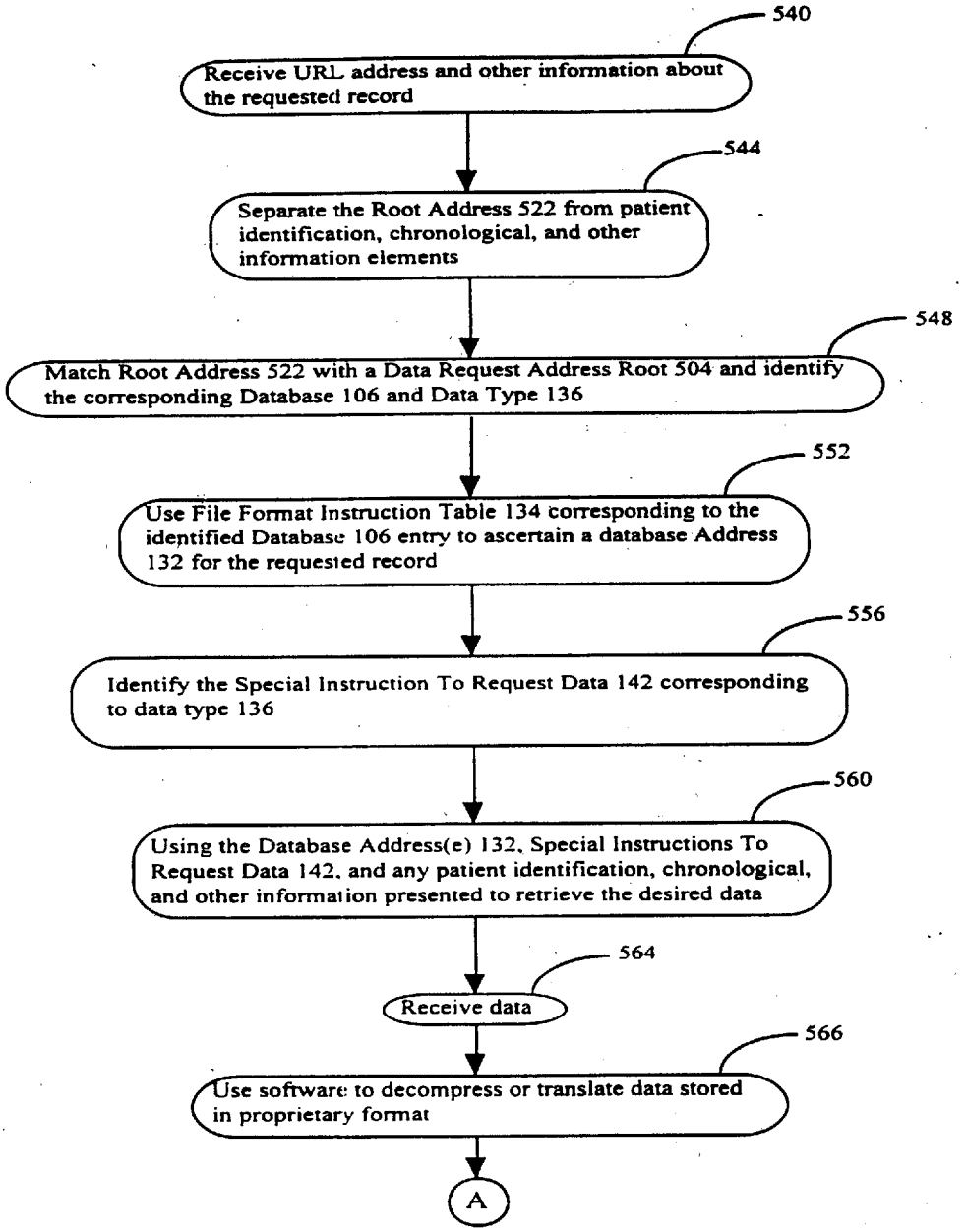

540
Receive URL address and other information about the requested record
544
Separate the Root Address 522 from patient identification, chronological, and other information elements
548
Match Root Address 522 with a Data Request Address Root 504 and identify the corresponding Database 106 and Data Type 136
552
Use File Format Instruction Table 134 corresponding to the identified Database 106 entry to ascertain a database Address 132 for the requested record
556
Identify the Special Instruction To Request Data 142 corresponding to data type 136
560
Using the Database Address(e) 132, Special Instructions To Request Data 142, and any patient identification, chronological, and other information presented to retrieve the desired data
564
Receive data
566
Use software to decompress or translate data stored in proprietary format
A

Figure 12A

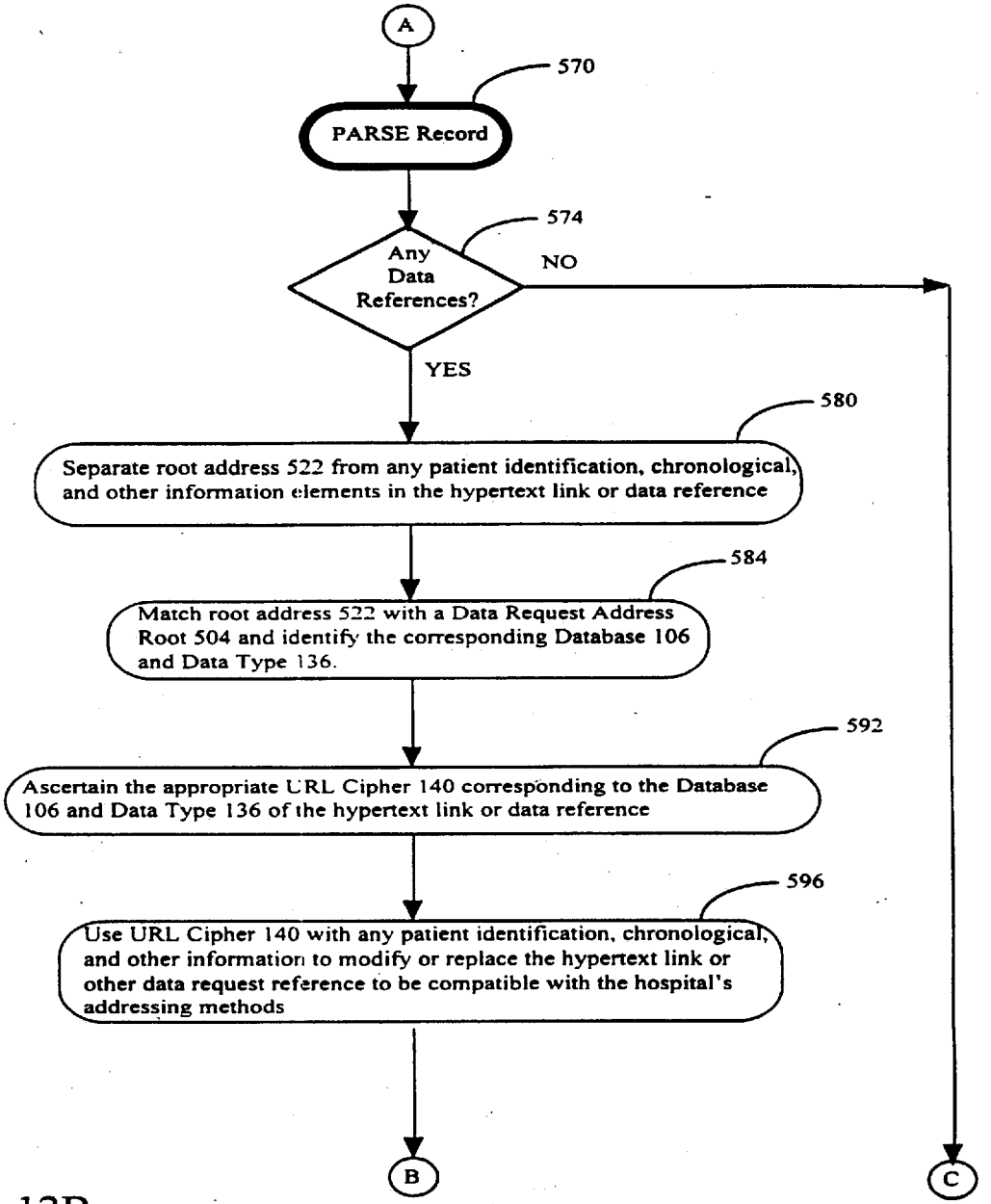


Figure 12B

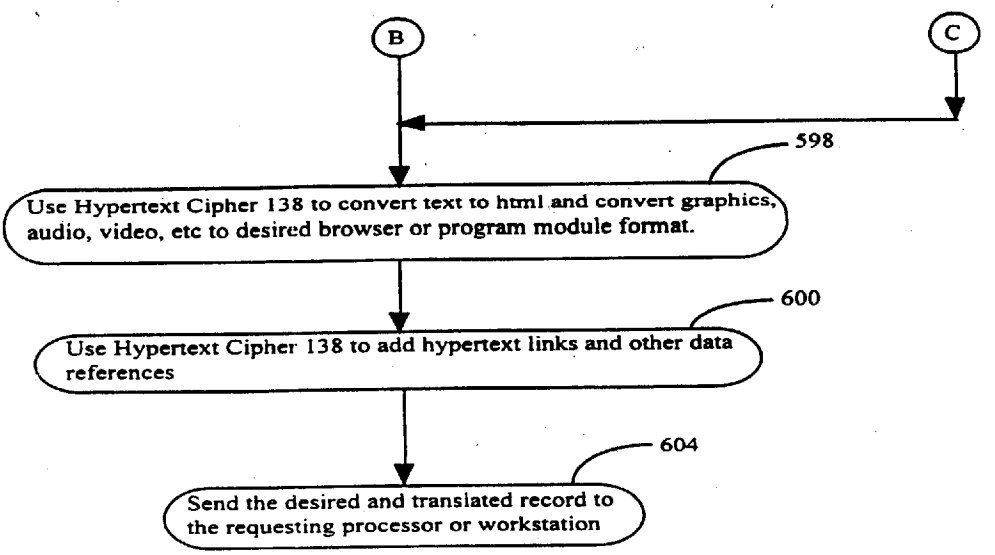

B
C
598
Use Hypertext Cipher 138 to convert text to html and convert graphics, audio, video, etc to desired browser or program module format.
600
Use Hypertext Cipher 138 to add hypertext links and other data references
604
Send the desired and translated record to the requesting processor or workstation

Figure 12C

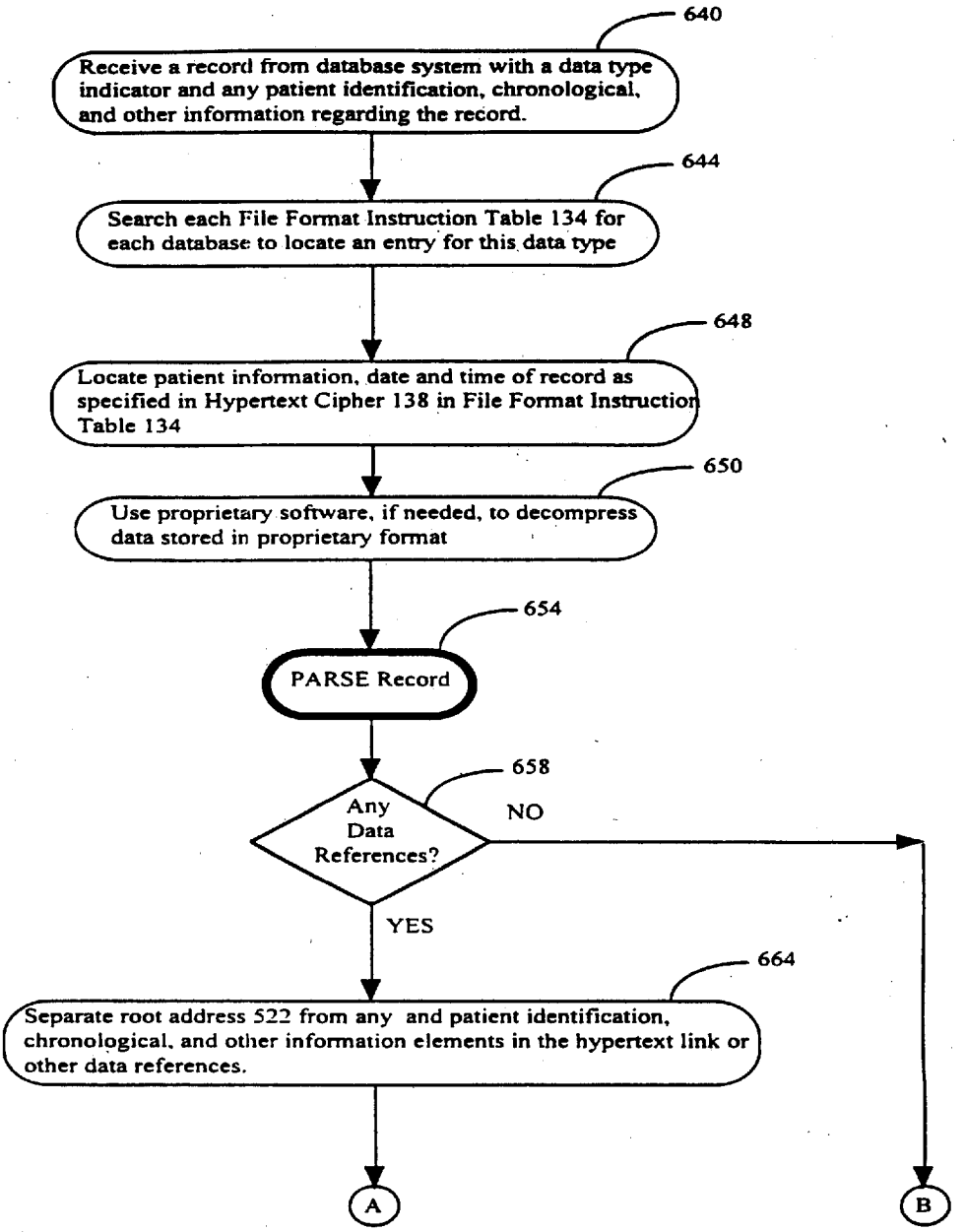

640
Receive a record from database system with a data type indicator and any patient identification, chronological, and other information regarding the record.
644
Search each File Format Instruction Table 134 for each database to locate an entry for this data type
648
Locate patient information, date and time of record as specified in Hypertext Cipher 138 in File Format Instruction Table 134
650
Use proprietary software, if needed, to decompress data stored in proprietary format
654
PARSE Record
658
Any Data References?
NO
YES
664
Separate root address 522 from any and patient identification, chronological, and other information elements in the hypertext link or other data references.
A
B

Figure 13A

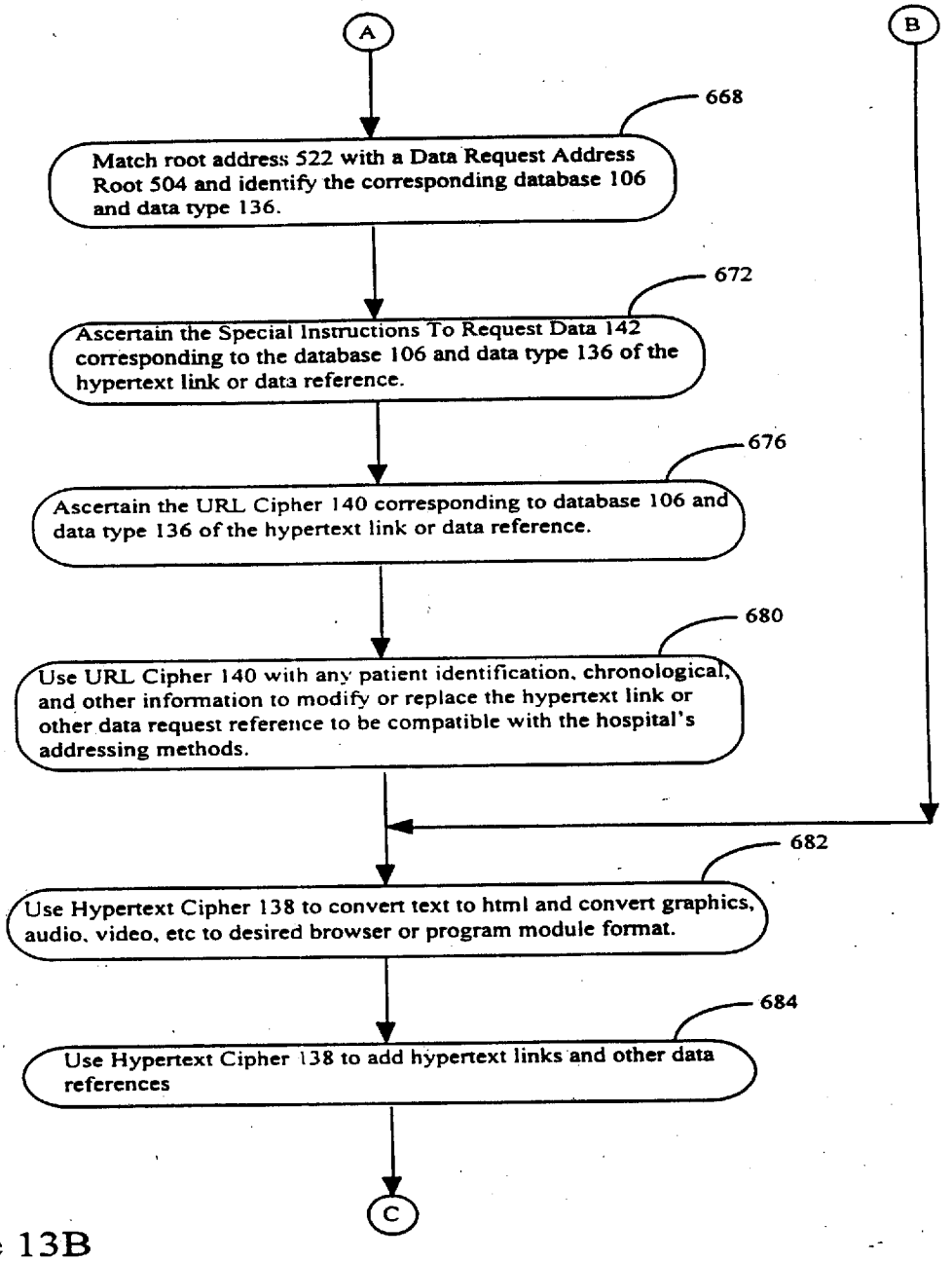


Figure 13B

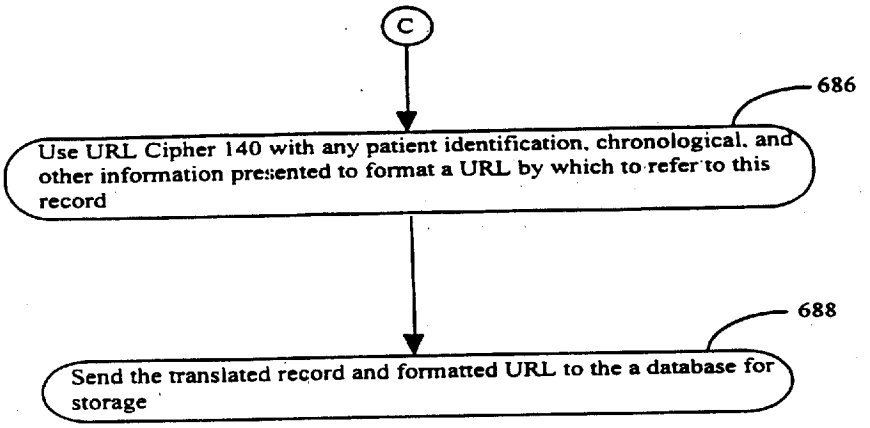


Figure 13C

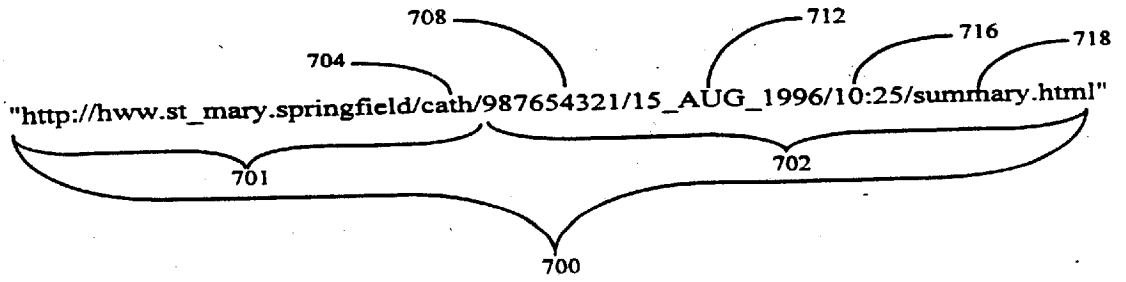


Figure 14A

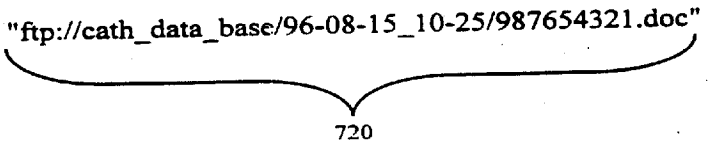


Figure 14B

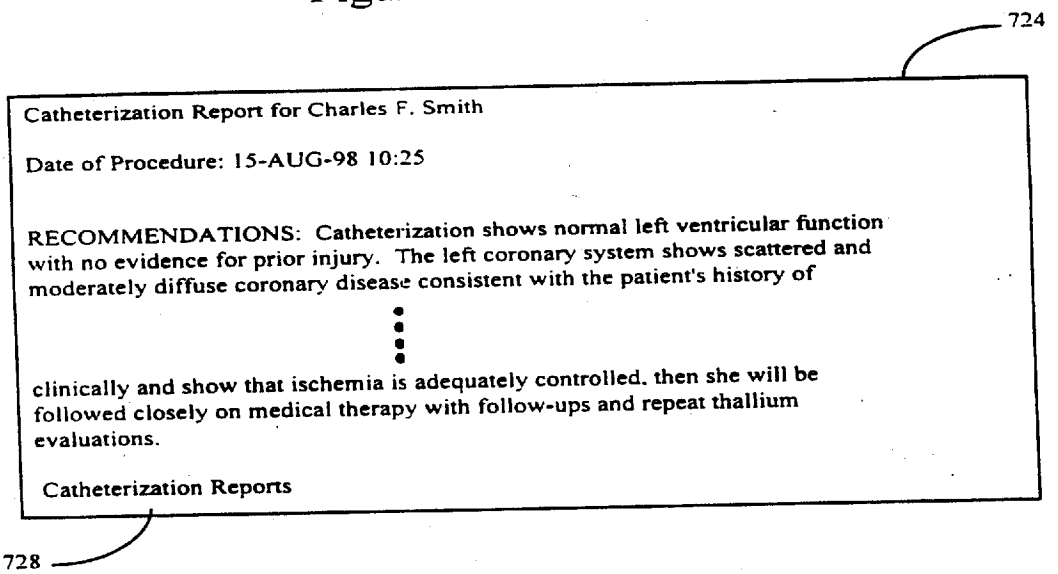


Figure 14C

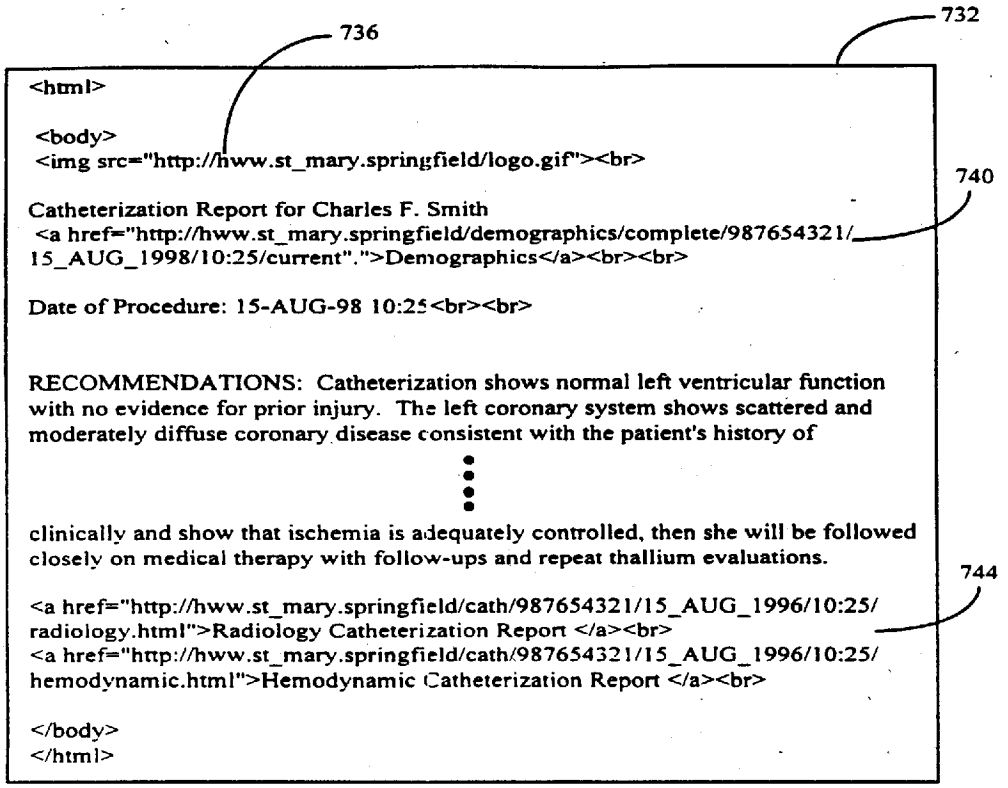


Figure 14D

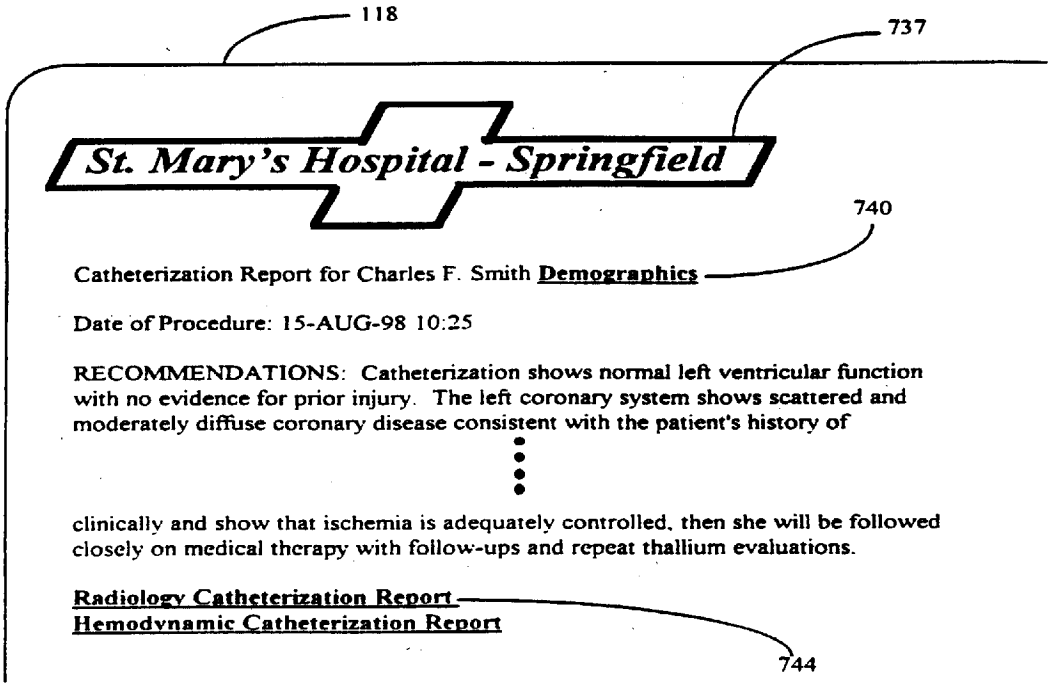

118
737
St. Mary's Hospital - Springfield
740
Catheterization Report for Charles F. Smith Demographics
Date of Procedure: 15-AUG-98 10:25
RECOMMENDATIONS: Catheterization shows normal left ventricular function with no evidence for prior injury. The left coronary system shows scattered and moderately diffuse coronary disease consistent with the patient's history of
clinically and show that ischemia is adequately controlled, then she will be followed closely on medical therapy with follow-ups and repeat thallium evaluations.
Radiology Catheterization Report
Hemodynamic Catheterization Report
744

Figure 14E

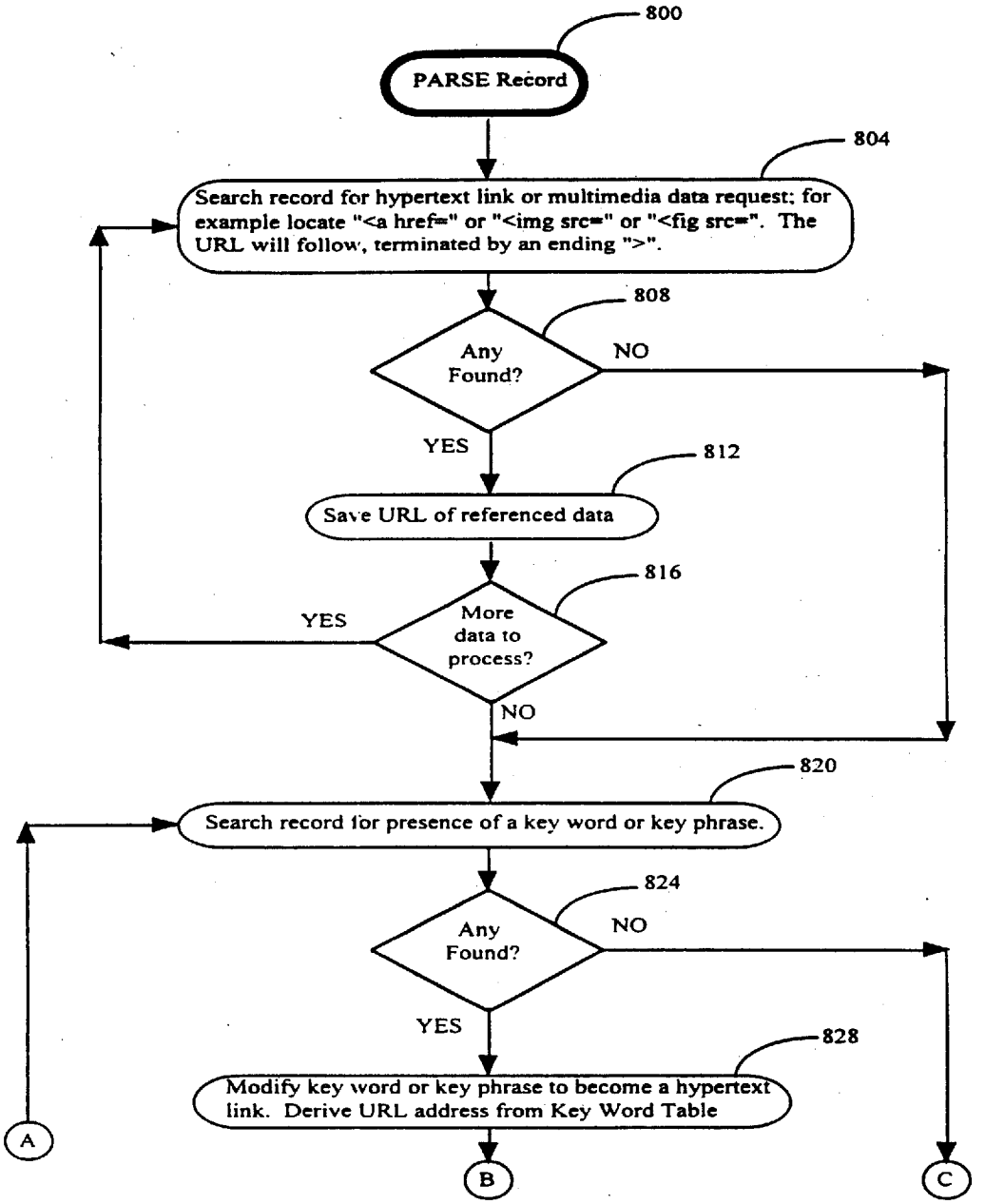

800
PARSE Record
804
Search record for hypertext link or multimedia data request; for example locate "<a href=" or "<img src=" or "<fig src=". The URL will follow, terminated by an ending ">".
808
Any Found?
NO
YES
812
Save URL of referenced data
816
More data to process?
YES
NO
820
Search record for presence of a key word or key phrase.
824
Any Found?
NO
YES
828
Modify key word or key phrase to become a hypertext link. Derive URL address from Key Word Table
A
B
C

Figure 15A

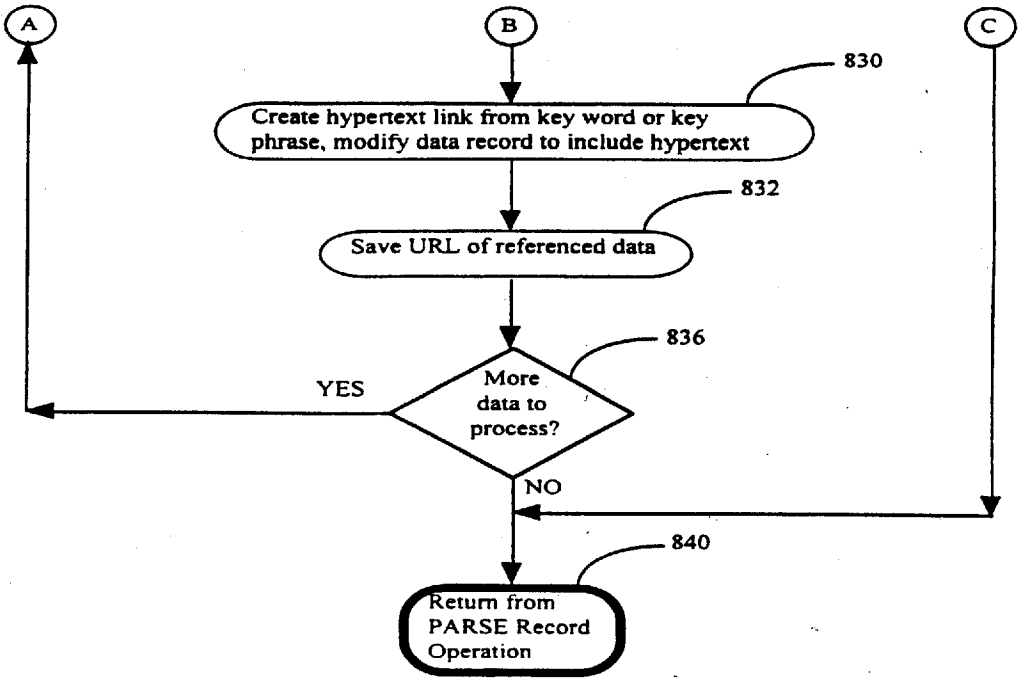


Figure 15B

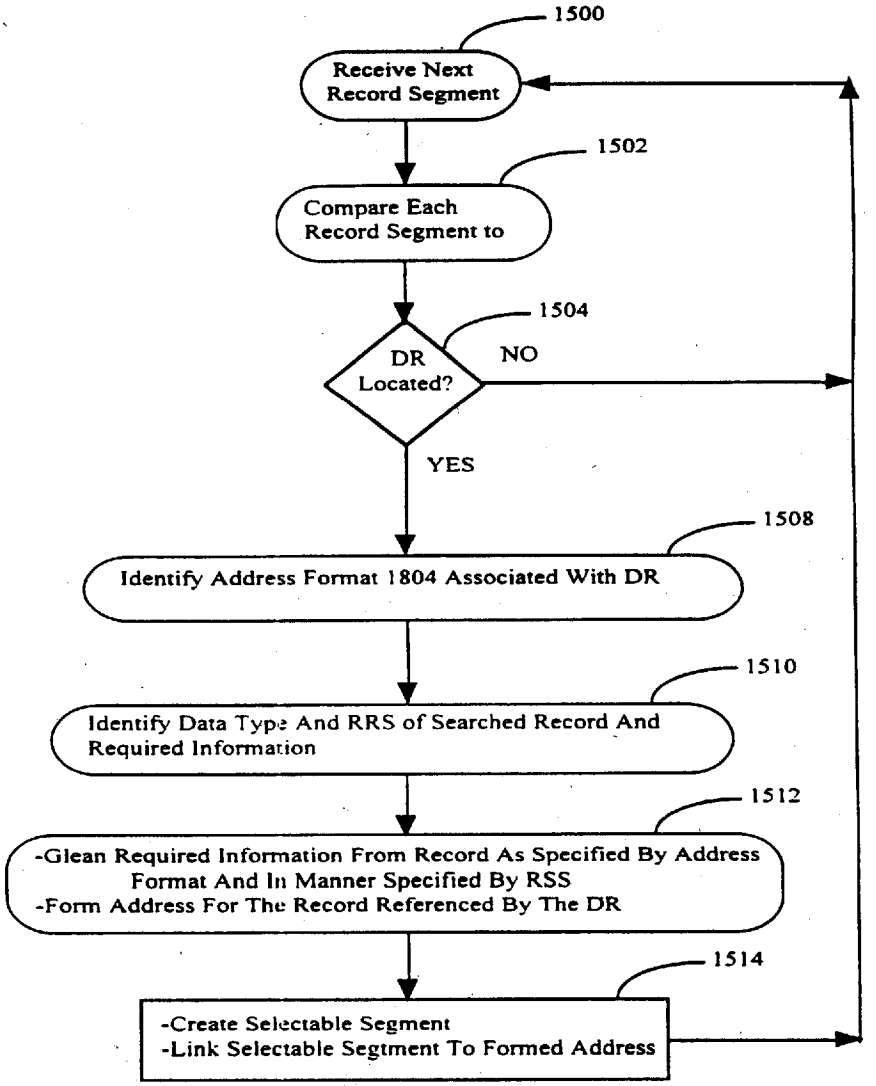

1500
Receive Next Record Segment
1502
Compare Each Record Segment to
1504
DR Located?
NO
YES
1508
Identify Address Format 1804 Associated With DR
1510
Identify Data Type And RRS of Searched Record And Required Information
1512
-Glean Required Information From Record As Specified By Address Format And In Manner Specified By RSS
-Form Address For The Record Referenced By The DR
1514
-Create Selectable Segment
-Link Selectable Segment To Formed Address

Figure 16

Text As It Appears in the Word Processor

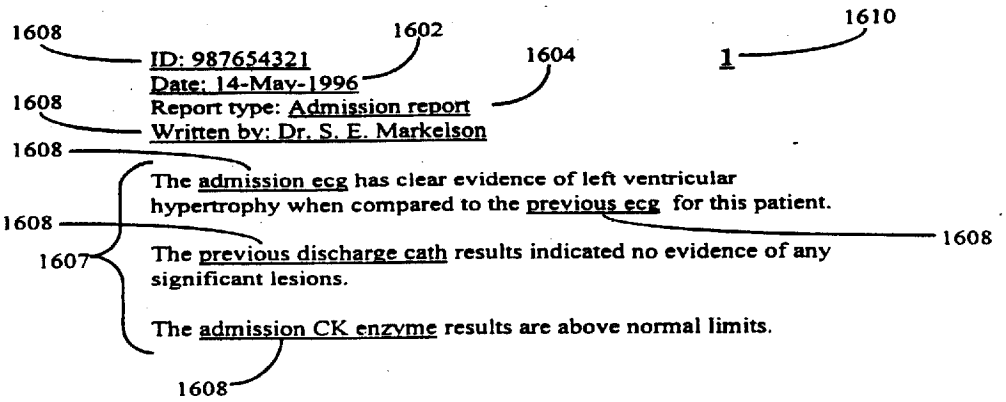


Figure 17

Text after Being Converted to HTML with Hypertext Links Added

```
<html>                                                                          1702
<body>
<a href= "http://hww.st_mary.springfield/demographics/987654321/14_May_1996">
ID: 987654321</a><br>
Date: 14-May-1996<br>                                                           1702
Report type: Admission report
Written by:  <a href="http://hww.st_mary.springfield/staff_directory/S._E._Markelson">
Dr. S. E. Markelson</a><br>                                                     1700
<br>
1608
The <a href="http://hww.st_mary.springfield/ecg/987654321/14_May_1996/00:00/admit">
admission ecg  </a> has clear evidence of left ventricular hypertrophy when compared to the
1608
 <a href="http://hww.st_mary.springfield/ecg/987654321/14_May_1996/00:00/current">
previous ecg  </a> for this patient.<br>
<br>                                                                            1702
The
<a
href="http://hww.st_mary.springfield/cath/987654321/14_May_1996/00:00/prev_discharge">
previous discharge cath </a> results indicated no evidence of any significant occlusions.<br>
<br>
The
<a href="http://hww.st_mary.springfield/lab_CK_enz/987654321/14_May_1996/00:00/admit">
admission CK enzyme </a>results are above normal limits.<br>

</body>
</html>
```

Figure 18

Text As Viewed via Word Processor 14 or Browser

ID: 987654321
Date: 14-May-1996
Report type: Admission report
Written by: Dr. S. E. Markelson

The admission ecg has clear evidence of left ventricular hypertrophy when compared to the current ecg for this patient.

The previous discharge cath results indicated no evidence of any significant lesions.

The admission CK enzyme results are above normal limits.

Figure 19

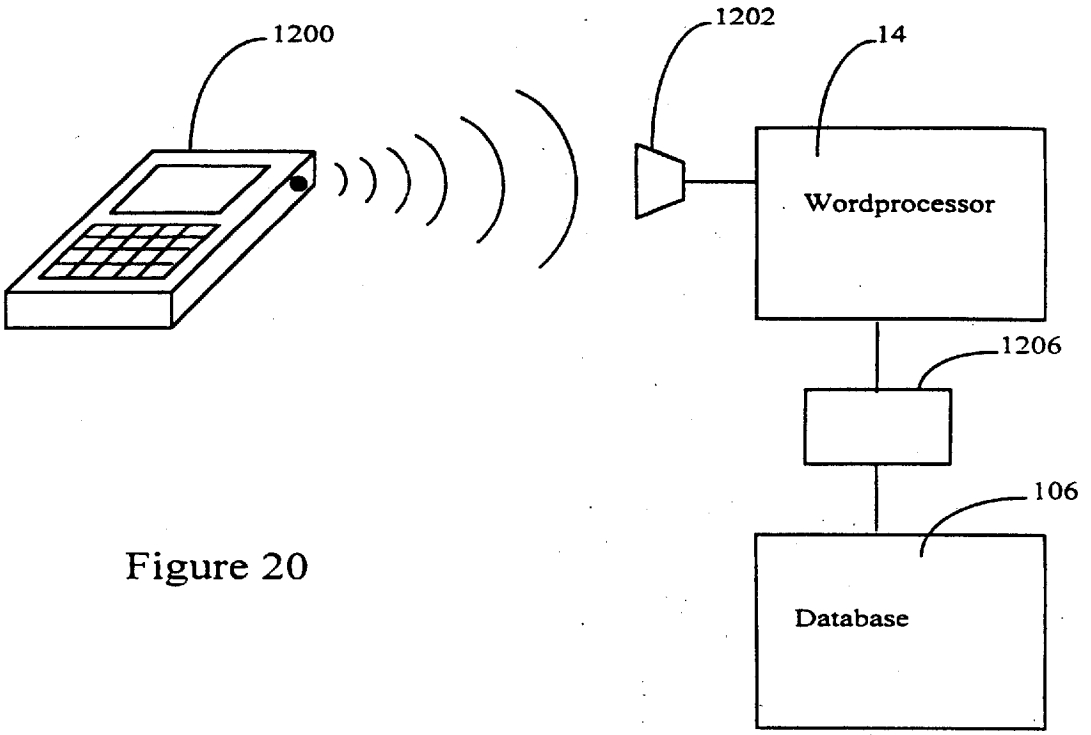


Figure 20

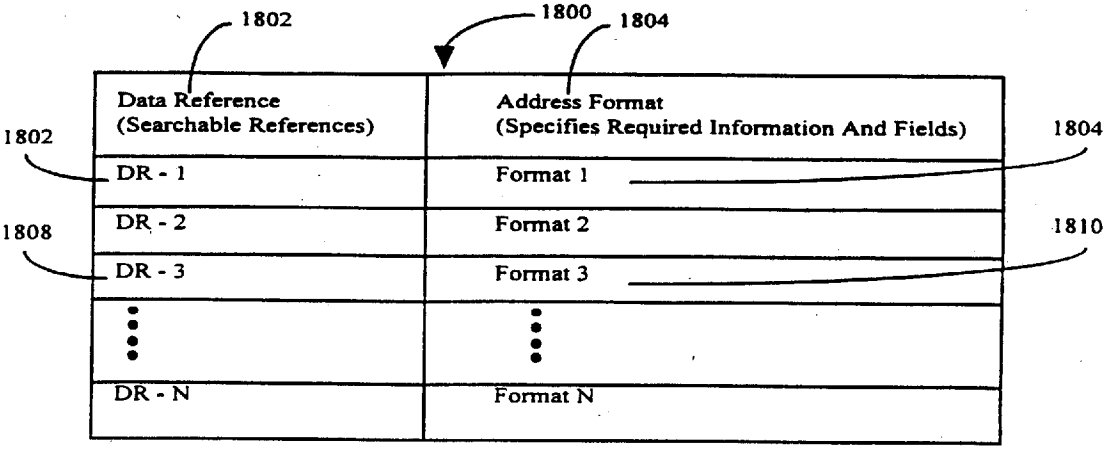


Figure 21

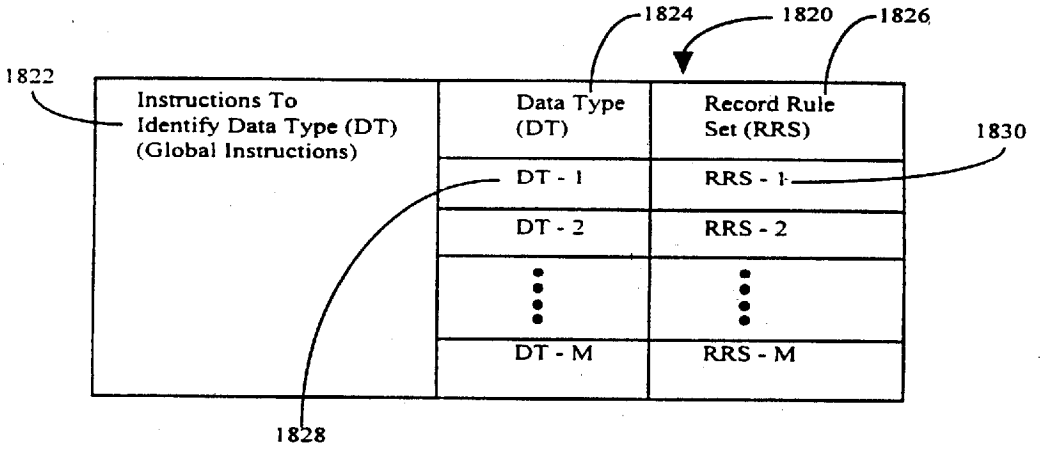


Figure 22

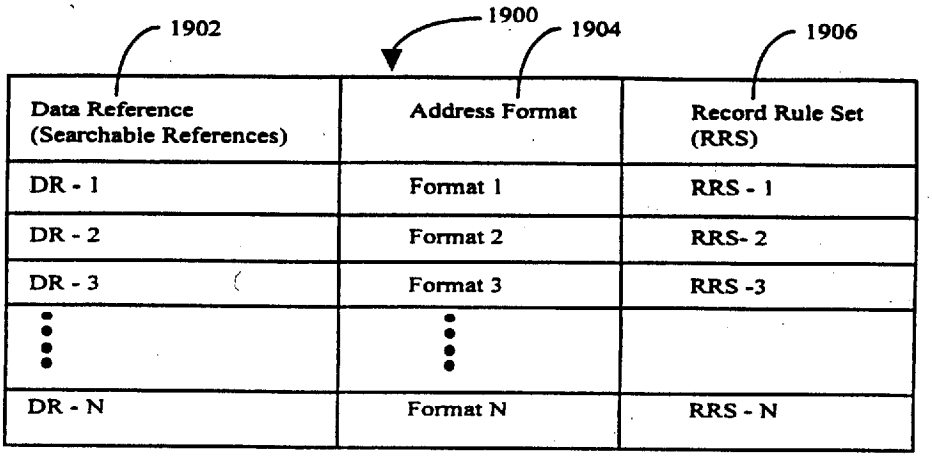


Figure 23

EXPRESS MAIL LABEL NO. EJ 636 882 900 US

PTO/SB/17 (12/98)
Approved for use through 9/30/2000. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

FEE TRANSMITTAL for FY 2000

Patent fees are subject to annual revision.
Small Entity payments must be supported by a small entity statement
otherwise large entity fees must be paid, See Forms PTO/SB/09-12

TOTAL AMOUNT OF PAYMENT	$ 780.00

Complete if Known	
Application Number	
Filing Date	Herewith
First Named Inventor	Carlos De La Huerga
Group Art Unit	
Examiner Name	
Attorney Docket Number	250591.90163

METHOD OF PAYMENT (check one)

1. [X] The Commissioner is hereby authorized to charge indicated fees and credit any over payments to:

Deposit Account Number: 17-0055

Deposit Account Name: Quarles & Brady LLP

[X] Charge Any Additional Fee Required Under 37 CFR 1.16 and 1.17

2. [ ] Payment Enclosed:

[ ] Check [ ] Money Order [ ] Other

FEE CALCULATION (fees effective 11/10/98)

1. FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	690	201	345	Utility filing fee	345.00
106	310	206	155	Design filing fee	
107	480	207	240	Plant filing fee	
108	690	208	345	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1) ($)345.00

2. CLAIMS

			Extra		Fee from below		Fee Paid
Total Claims	64	-20**=	44	X	9.00	=	369.00
Independent Claims	4	-3**=	1	X	39.00	=	39.00
Multiple Dependent Claims					130.00	=	0.00

** or number previously paid, if greater, For reissues see below

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	9	Claims in excess of 20
102	78	202	39	Independent claims in excess of 3
104	260	204	130	Multiple dependent claim
109	78	209	39	Reissue independent claims over original patent
110	18	210	9	Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)435.00

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920	112	920	Requesting publication of SIR prior to Examiner action	
113	1,840	113	1,840	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	380	216	190	Extension for reply within second month	
117	870	217	435	Extension for reply within third month	
118	1,360	218	680	Extension for reply within fourth month	
128	1,850	228	925	Extension for reply within fifth month	
119	300	219	150	Notice of Appeal	
120	300	220	150	Filing a brief in support of an appeal	
121	260	221	130	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive unavoidably abandoned application	
141	1,210	241	605	Petition to revive unintentionally abandoned application	
142	1,210	242	605	Utility issue fee (or reissue)	
143	430	243	215	Design issue fee	
144	580	244	290	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	
146	760	246	380	Filing a submission after final rejection (37 CFR 1.129(a))	
149	760	249	380	For each additional invention to be examined (37 CFR 1.129(b))	

Other fee (specify) _____

Other fee (specify) _____

SUBTOTAL (3) ($)0.00

* Reduced by Basic Filing Fee Paid

SUBMITTED BY				Complete (if applicable)	
Typed or Printed Name	Michael A. Jaskolski			Reg. Number	37,551
Signature	M. Jaskolski	Date	2/24/00	Deposit Account User ID	17-0055

MULTIPLE DEPENDENT CLAIM FEE CALCULATION SHEET

(FOR USE WITH FORM PTO-875)

SERIAL NO. 09/512,125

FILING DATE 2/24/00

APPLICANT(S)

CLAIMS

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT		*		*		*	
	IND.	DEP.	IND.	DEP.	IND.	DEP.	IND.	DEP.	IND.	DEP.	IND.	DEP.
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												
32												
33												
34												
35												
36												
37												
38												
39												
40												
41												
42												
43												
44												
45												
46												
47												
48												
49												
50												
TOTAL IND.												
TOTAL DEP.												
TOTAL CLAIMS												

	*		*		*	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
51						
52						
53						
54						
55						
56						
57						
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						
76						
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						
91						
92						
93						
94						
95						
96						
97						
98						
99						
100						
TOTAL IND.	4					
TOTAL DEP.	60					
TOTAL CLAIMS	64					

PTO-1360 (3-78)

*MAY BE USED FOR ADDITIONAL CLAIMS OR AMENDMENTS

U.S. DEPARTMENT of COMMERCE
Patent and Trademark Office

1111

PATENT APPLICATION FEE DETERMINATION RECORD
Effective December 29, 1999

Application or Docket Number: 09/512125

CLAIMS AS FILED - PART I

	(Column 1) NUMBER FILED		(Column 2) NUMBER EXTRA
FOR			
BASIC FEE			
TOTAL CLAIMS	04	minus 20=	* 44
INDEPENDENT CLAIMS	4	minus 3 =	* 1
MULTIPLE DEPENDENT CLAIM PRESENT			N

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
	345.00	OR		690.00
X$ 9=		OR	X$18=	—
X39=		OR	X78=	—
+130=		OR	+260=	—
TOTAL		OR	TOTAL	1560 —

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X39=		OR	X78=	
+130=		OR	+260=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875
(Rev. 12/99)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
*U.S. GPO: 2000-463-433/29044

UNITED STAT DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
09/512,125	02/24/00	DE LA HUERGA	C 250591.90163

026710
QUARLES & BRADY LLP
411 E. WISCONSIN AVENUE
SUITE 2040
MILWAUKEE WI 53202-4497

WM01/0118

EXAMINER	
COBY, F	
ART UNIT	**PAPER NUMBER**
2171	2

DATE MAILED: 01/18/01

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

PTO-90C (Rev. 2/95)
*U.S. GPO: 2000-473-000/44602

1- File Copy

Office Action Summary	Application No. 09/512,125	Applicant(s) De La Huerga
	Examiner Frantz Coby	Group Art Unit 2171

☒ Responsive to communication(s) filed on *Feb 24, 2000*

☐ This action is **FINAL.**

☐ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

A shortened statutory period for response to this action is set to expire *3* month(s), or thirty days, whichever is longer, from the mailing date of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

Disposition of Claim

☒ Claim(s) *1-64* is/are pending in the applicat

Of the above, claim(s) _____ is/are withdrawn from consideration

☐ Claim(s) _____ is/are allowed.

☒ Claim(s) *1-64* is/are rejected.

☐ Claim(s) _____ is/are objected to.

☐ Claims _____ are subject to restriction or election requirement.

Application Papers

☒ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

☐ The drawing(s) filed on _____ is/are objected to by the Examiner.

☐ The proposed drawing correction, filed on _____ is ☐ approved ☐ disapproved.

☐ The specification is objected to by the Examiner.

☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

☐ All ☐ Some* ☒ None of the CERTIFIED copies of the priority documents have been

☐ received.

☐ received in Application No. (Series Code/Serial Number) _____.

☐ received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

*Certified copies not received: _____

☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

☒ Notice of References Cited, PTO-892

☐ Information Disclosure Statement(s), PTO-1449, Paper No(s). _____

☐ Interview Summary, PTO-413

☒ Notice of Draftsperson's Patent Drawing Review, PTO-948

☐ Notice of Informal Patent Application, PTO-152

--- SEE OFFICE ACTION ON THE FOLLOWING PAGES ---

This is a first action in response to application filed on February 24, 2000 in which claims 1-64 are presented for examination

Drawings

1. This application has been filed with informal drawings which are acceptable for examination purposes only. Formal drawings will be required when the application is allowed.

Priority

Acknowledgment is made of benefit claim under Title 35, United States Code 120 of United States Applications 08/727,293 filed on October 09, 1996 now U.S. Patent no. 5,895,461 which claimed priority from provisional application 60//023,126 filed on July 30, 1996; 08/871,818 filed on June 09, 1997 now U.S. Patent no. 5,903,889; and 09/247,349 filed on February 10, 1999 now pending.

Double Patenting

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321© may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 1-64 are rejected under the judicially created doctrine of double patenting over claims 1-60 of U. S. Patent No. 5,895,461 since the claims, if allowed, would improperly extend the "right to exclude" already granted in the patent.

The subject matter claimed in the instant application is fully disclosed in the patent and is covered by the patent since the patent and the application are claiming common subject matter, as follows:

recognizing a first keyword phrase is associated with a keyword phrase and identifying second record which is referenced by the keyword phrase.

Furthermore, there is no apparent reason why applicant was prevented from presenting claims corresponding to those of the instant application during prosecution of the application which matured into a patent. See *In re Schneller*, 397 F.2d 350, 158 USPQ 210 (CCPA 1968). See also MPEP § 804.

Conclusion

Any response to this action should be mailed to:

Commissioner of Patents and trademarks

Washington, D.C. 20231

or faxed to:

(703) 305-9051, (for formal communications intended for entry)

Or:

(703) 308-5357 (for informal of draft communications, please label "PROPOSED" or "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2021 Crystal Drive, Arlington. VA., Sixth Floor (Receptionist).

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Frantz Coby whose telephone number is (703) 305-4006. The examiner can normally be reached Monday through Friday from 9:30 A.M. to 5:00 P.M.

5. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thomas black, can be reached on (703) 305-9707. The Fax phone number for this Group is (703) 308-5403.

Frantz Coby

January 9, 2001

Notice of References Cited

Application No.	Applicant(s)		
09/512,125	De La Huerga		
Examiner	Group Art Unit	Page 1 of 1	
Frantz Coby	2171		

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
	A	5,895,461	4/1999	De La Huerga et al.	707	1
	B	5,558,638	9/1996	Evers et al.	705	3
	C	5,291,399	3/1994	Chaco	705	3
	D	5,903,889	5/1999	De La Huerga et al.	707	3
	E	5,942,986	8/1999	Shabot et al.	340	825.44
	F	6,157,914	12/2000	Seto et al.	705	3
	G	5,867,821	2/1999	Ballantyne et al.	705	2
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	NAME	CLASS	SUBCLASS
	N						
	O						
	P						
	Q						
	R						
	S						
	T						

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	U		
	V		
	W		
	X		

#3 gmm 3-30-01

PTO/SB/26 (10-99)
Approved for use through 09/30/200, OMB 0651-0031
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A PRIOR PATENT	Docket Number (Optional) 250591.90163

In re Application of: **Carlos De La Huerga**

Application No.: **09/512,125**

Filed: **February 24, 2000**

For: **METHOD AND SYSTEM FOR AUTOMATED DATA STORAGE AND RETRIEVAL**

The owner*, **Carlos De La Huerga** of 100% percent interest in the instant application, hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application, which would extend beyond the expiration date of the full statutory term defined in 35 U.S.C. 154 to 156 and 173, as presently shortened by any terminal disclaimer, of prior Patent No. **5,895,461**. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of any patent granted on the instant application that would extend to the expiration date of the full statutory term as defined in 35 U.S.C. 154 to 156 and 173 of the prior patent, as presently shortened by any terminal disclaimer, in the event that it later: expires for failure to pay a maintenance fee, is held unenforceable, is found invalid by a court of competent jurisdiction, is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321, has all claims cancelled by a reexamination certificate, is reissued, or is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

Check either box 1 or 2 below, if appropriate.

1. [ ] For submissions on behalf of an organization (e.g., corporation, partnership, university, government agency, etc.), the undersigned is empowered to act on behalf of the organization.

   I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

2. [X] The undersigned is an attorney of record.

3. [X] Please charge the $55.00 fee for this Terminal Disclaimer and any other fees due in this application to Deposit Account No. 17-0055.

3/13/01
Date

Signature

Michael A. Jaskolski

* Statement under 37 CFR 3.373(b) is required if terminal disclaimer is signed by the assignee (owner). Form PTO/SB/96 may be used for making this statement. See MPEP §324

2171

I hereby certify that this correspondence is being deposited with the United States Postal Services on the date set forth below as First Class Mail in an envelope addressed to: Assistant Commissioner for Patents, Washington D.C. 20231.

Date of Signature and Deposit: 3/13/01

Michael A. Jaskolski, Reg. No. 37,551

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Carlos De La Huerga
Serial No.:	09/512,125
Filed:	February 24, 2000
Title:	METHOD AND SYSTEM FOR AUTOMATED DATA STORAGE AND RETREIVAL
Examiner:	F. Coby
Art Unit:	2171
Docket:	250591.90163

SUBMISSION OF TERMINAL DISCLAIMER AND ACCOMPANYING FEE

Commissioner for Patents
Washington DC 20231

Dear Sir:

Enclosed please find a terminal disclaimer for the above-referenced case and a fee transmittal form. Please enter these documents in the above-referenced case. Please call if you have any questions or need any other information.

Respectfully submitted,

CARLOS DE LA HUERGA

Date: March 9, 2001

Michael A. Jaskolski
Quarles & Brady, LLP
Reg. No. 37,551
Attorney for Applicant
411 East Wisconsin Avenue
Milwaukee WI 53202
414/277-5711

QBMKE\5020013.1

EXPRESS MAIL LABEL NO. _____

PTO/SB/17 (8/00)
Approved for use through 10/31/2002. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

FEE TRANSMITTAL for FY 2001

Patent fees are subject to annual revision.

TOTAL AMOUNT OF PAYMENT	$55.00

Complete if known

Application Number	09/512,125
Filing Date	February 24, 2000
First Named Inventor	Carlos De La Huerga
Group Art Unit	
Examiner Name	
Attorney Docket Number	250591.90163

METHOD OF PAYMENT (check one)

1. [X] The Commissioner is hereby authorized to charge indicated fees and credit any over payments to:

Deposit Account Number: 17-0055

Deposit Account Name: Quarles & Brady LLP

[X] Charge Any Additional Fee Required Under 37 CFR 1.16 and 1.17

[ ] Applicant claims small entity status See 37 CFR 1.27

2. [ ] Payment Enclosed:

[ ] Check [ ] Credit Card [ ] Money Order [ ] Other

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	710	201	355	Utility filing fee	
106	320	206	160	Design filing fee	
107	490	207	245	Plant filing fee	
108	710	208	355	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1) ($)0.00

2. CLAIMS

		Extra		Fee from below		Fee Paid
Total Claims		-20**=		X	=	
Independent		-3**=		X	=	
Multiple Dependent Claims					=	

** or number previously paid, if greater, For reissues see below

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	9	Claims in excess of 20
102	80	202	40	Independent claims in excess of 3
104	270	204	135	Multiple dependent claim
109	80	209	40	**Reissue independent claims over original patent
110	18	210	9	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)0.00

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920	112	920	Requesting publication of SIR prior to Examiner action	
113	1,840	113	1,840	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	390	216	195	Extension for reply within second month	
117	890	217	445	Extension for reply within third month	
118	1,390	218	695	Extension for reply within fourth month	
128	1,890	228	945	Extension for reply within fifth month	
119	310	219	155	Notice of Appeal	
120	310	220	155	Filing a brief in support of an appeal	
121	270	221	135	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive unavoidably abandoned application	
141	1,240	241	620	Petition to revive unintentionally abandoned application	
142	1,240	242	620	Utility issue fee (or reissue)	
143	440	243	220	Design issue fee	
144	600	244	300	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	
146	710	246	355	Filing a submission after final rejection (37 CFR 1.129(a))	
149	710	2496	355	For each additional invention td be examined (37 CFR 1.129(b))	
179	710	270	355	Request for Continued Examination (RCE)	
169	900	169	900	Request for expedited examination of a design application	
Other fee (specify)	248 - Terminal Disclaimer				55.00

* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($)55.00

SUBMITTED BY				Complete (if applicable)	
Typed or Printed Name	Michael A. Jaskolski	Registration No. (Attorney/Agent)	37,551	Telephone	414.277.5711
Signature	M. Jaskolski			Date	3/13/01

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
09/512,125	02/23/00	DE LA HUERGA	C

TM31/0409

QUARLES & BRADY LLP
411 E. WISCONSIN AVENUE
SUITE 2040
MILWAUKEE WI 53202-4497

COBY, F	EXAMINER

2174	
ART UNIT	PAPER NUMBER

04/09/01
DATE MAILED:

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

Notice of Allowability	Application No. 09/512,125	Applicant(s) DE LA HUERGA, CARLOS
	Examiner Frantz Coby	Art Unit 2171

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance and Issue Fee Due or other appropriate communication will be mailed in due course.

1. ☒ This communication is responsive to *communication filed on March 13, 2001*.
2. ☒ The allowed claim(s) is/are *1-64*.
3. ☐ The drawings filed on _____ are acceptable.
4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).
   a) ☐ All b) ☐ Some* c) ☐ None of the CERTIFIED copies of the priority documents have been
      1. ☐ received.
      2. ☐ received in Application No. (Series Code / Serial Number). _____ .
      3. ☐ received in this national stage application from the International Bureau (PCT Rule 17.2(a)).
   * Certified copies not received: _____.
5. ☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. & 119(e).

A SHORTENED STATUTORY PERIOD FOR REPLY to comply with the requirements noted below is set to EXPIRE **THREE MONTHS** FROM THE "DATE MAILED" of this Office Action. Failure to timely comply will result in ABANDONMENT of this application. Extensions of time may be available under the provisions of 37 CFR 1.136(a).

6. ☐ Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient. A SUBSTITUTE OATH OR DECLARATION IS REQUIRED.

7. ☒ Applicant MUST submit NEW FORMAL DRAWINGS
   (a) ☐ because the originally filed drawings were declared by applicant to be informal.
   (b) ☒ including changes required by the Notice of Draftsperson's Patent Drawing Review(PTO-948) attached
      1) ☒ hereto or 2) ☐ to Paper No. ____.
   (c) ☐ including changes required by the proposed drawing correction filed _____, which has been approved by the examiner.
   (d) ☐ including changes required by the attached Examiner's Amendment / Comment.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the reverse side of the drawings. The drawings should be filed as a separate paper with a transmittal letter addressed to the Official Draftsperson.

8. ☐ Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Any reply to this letter should include, in the upper right hand corner, the APPLICATION NUMBER (SERIES CODE / SERIAL NUMBER). If applicant has received a Notice of Allowance and Issue Fee Due, the ISSUE BATCH NUMBER and DATE of the NOTICE OF ALLOWANCE should also be included.

Attachment(s)

1☐ Notice of References Cited (PTO-892)
3☒ Notice of Draftperson's Patent Drawing Review (PTO-948)
5☐ Information Disclosure Statements (PTO-1449), Paper No. ____.
7☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
2☐ Notice of Informal Patent Application (PTO-152)
4☐ Interview Summary (PTO-413), Paper No.____ .
6☐ Examiner's Amendment/Comment
8☒ Examiner's Statement of Reasons for Allowance
9☐ Other

THOMAS BLACK
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

This is in response to Applicant's response filed on March 09, 2001 in which a terminal disclaimer was filed.

Allowable Subject Matter

1. Claims 1-64 are allowed.

2. The following is an examiner's statement of reasons for allowance: *the steps of determining if the first record phrase is associated with a keyword phrase; identifying at least one second record which is referenced by the keyword phrase and rendering the identified record accessible.* These limitations, in conjunction with all other limitations of the base claims were not shown by, would not have been obvious over, nor would have been fairly suggested by the prior art of record.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any response to this action should be mailed to:

Commissioner of Patents and trademarks

Washington, D.C. 20231

or faxed to:

(703) 305-9051, (for formal communications intended for entry)

Or:

(703) 308-5357 (for informal of draft communications, please label "PROPOSED" or "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2021 Crystal Drive, Arlington. VA., Sixth Floor (Receptionist).

3. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Frantz Coby whose telephone number is (703) 305-4006. The examiner can normally be reached Monday through Friday from 9:30 A.M. to 5:00 P.M.

4. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thomas black, can be reached on (703) 305-9707. The Fax phone number for this Group is (703) 308-5403.

Frantz Coby

Frantz Coby

April 2, 2001

THOMAS BLACK
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Form PTO 948 (Rev. 8-98) U.S. DEPARTMENT OF COMMERCE - Patent and Trademark Office Application No. 09/512,125

NOTICE OF DRAFTSPERSON'S PATENT DRAWING REVIEW

The drawing(s) filed (insert date) 2/24/00 are:

A. ☐ approved by the Draftsperson under 37 CFR 1.84 or 1.152.

B. ☒ objected to by the Draftsperson under 37 CFR 1.84 or 1.152 for the reasons indicated below. The Examiner will require submission of new, corrected drawings when necessary. Corrected drawing must be sumitted according to the instructions on the back of this notice.

1. DRAWINGS. 37 CFR 1.84(a): Acceptable categories of drawings: Black ink. Color.
   - ____ Color drawings are not acceptable until petiton is granted. Fig(s) ____
   - ____ Pencil and non black ink not permitted. Fig(s) ____
2. PHOTOGRAPHS. 37 CFR 1.84 (b)
   - ____ 1 full-tone set is required. Fig(s) ____
   - ____ Photographs not properly mounted (must use brystol board or photographic double-weight paper). Fig(s) ____
   - ____ Poor quality (half-tone). Fig(s) ____
3. TYPE OF PAPER. 37 CFR 1.84(e)
   - ____ Paper not flexible, strong, white, and durable. Fig(s) ____
   - ____ Erasures, alterations, overwritings, interlineations, folds, copy machine marks not accepted. Fig(s) ____
   - ____ Mylar, velum paper is not acceptable (too thin). Fig(s) ____
4. SIZE OF PAPER. 37 CFR 1.84(f): Acceptable sizes:
   - ____ 21.0 cm by 29.7 cm (DIN size A4)
   - ____ 21.6 cm by 27.9 cm (8 1/2 x 11 inches)
   - ____ All drawing sheets not the same size. Sheet(s) ____
   - ____ Drawings sheets not an acceptable size. Fig(s) ____
5. MARGINS. 37 CFR 1.84(g): Acceptable margins:

   Top 2.5 cm Left 2.5cm Right 1.5 cm Bottom 1.0 cm SIZE: A4 Size

   Top 2.5 cm Left 2.5 cm Right 1.5 cm Bottom 1.0 cm SIZE: 8 1/2 x 11

   Margins not acceptable. Fig(s) ____

   ____ Top (T) ____ Left (L)

   ____ Right (R) ____ Bottom (B)
6. VIEWS. 37 CFR 1.84(h)

   REMINDER: Specification may require revision to correspond to drawing changes.

   Partial views. 37 CFR 1.84(h)(2)
   - ____ Brackets needed to show figure as one entity. Fig(s) ____
   - ____ Views not labeled separately or properly. Fig(s) ____
   - ____ Enlarged view not labeled separetely or properly. Fig(s) ____
7. SECTIONAL VIEWS. 37 CFR 1.84 (h)(3)
   - ____ Hatching not indicated for sectional portions of an object. Fig(s) ____
   - ____ Sectional designation should be noted with Arabic or Roman numbers. Fig(s) ____
8. ARRANGEMENT OF VIEWS. 37 CFR 1.84(i)
   - ____ Words do not appear on a horizontal, left-to-right fashion when page is either upright or turned so that the top becomes the right side, except for graphs. Fig(s) ____
9. SCALE. 37 CFR 1.84(k)
   - ____ Scale not large enough to show mechanism without crowding when drawing is reduced in size to two-thirds in reproduction. Fig(s) ____
10. CHARACTER OF LINES, NUMBERS, & LETTERS. 37 CFR 1.84(i)
    - ____ Lines, numbers & letters not uniformly thick and well defined, clean, durable, and black (poor line quality). Fig(s) ____
11. SHADING. 37 CFR 1.84(m)
    - ____ Solid black areas pale. Fig(s) ____
    - ____ Solid black shading not permitted. Fig(s) ____
    - ____ Shade lines, pale, rough and blurred. Fig(s) ____
12. NUMBERS, LETTERS, & REFERENCE CHARACTERS. 37 CFR 1.84(p)
    - ____ Numbers and reference characters not plain and legible. Fig(s) ____
    - ____ Figure legends are poor. Fig(s) ____
    - ____ Numbers and reference characters not oriented in the same direction as the view. 37 CFR 1.84(p)(1) Fig(s) ____
    - ____ English alphabet not used. 37 CFR 1.84(p)(2) Figs ____
    - ☒ Numbers, letters and reference characters must be at least .32 cm (1/8 inch) in height. 37 CFR 1.84(p)(3) Fig(s) 1-3B, 5B-18, 21-23
13. LEAD LINES. 37 CFR 1.84(q)
    - ____ Lead lines cross each other. Fig(s) ____
    - ____ Lead lines missing. Fig(s) ____
14. NUMBERING OF SHEETS OF DRAWINGS. 37 CFR 1.84(t)
    - ____ Sheets not numbered consecutively, and in Arabic numerals beginning with number 1. Sheet(s) ____
15. NUMBERING OF VIEWS. 37 CFR 1.84(u)
    - ____ Views not numbered consecutively, and in Arabic numerals, beginning with number 1. Fig(s) ____
16. CORRECTIONS. 37 CFR 1.84(w)
    - ____ Corrections not made from prior PTO-948 dated ____
17. DESIGN DRAWINGS. 37 CFR 1.152
    - ____ Surface shading shown not appropriate. Fig(s) ____
    - ____ Solid black shading not used for color contrast. Fig(s) ____

COMMENTS

REVIEWER A.D DATE 4/3/01 TELEPHONE NO. ____

ATTACHMENT TO PAPER NO. ____

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office

NOTICE OF ALLOWANCE AND ISSUE FEE DUE

026710
QUARLES & BRADY LLP
411 E. WISCONSIN AVENUE
SUITE 2040
MILWAUKEE WI 53202-4497

TM31/0409

APPLICATION NO.	FILING DATE	TOTAL CLAIMS	EXAMINER AND GROUP ART UNIT		DATE MAILED
09/512,125	02/24/00	064	COBY, F	2171	04/09/01
First Named Applicant	DE LA HUERGA,	35 USC 154(b) term ext. =		0 Days.	

TITLE OF INVENTION: METHOD AND SYSTEM FOR AUTOMATED DATA STORAGE AND RETRIEVAL

ATTY'S DOCKET NO.	CLASS-SUBCLASS	BATCH NO.	APPLN. TYPE	SMALL ENTITY	FEE DUE	DATE DUE
3 250591.90163	707-003.000	E69	UTILITY	NO	$1240.00	07/09/01

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED.

THE ISSUE FEE MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED.

HOW TO RESPOND TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.
If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is changed, pay twice the amount of the FEE DUE shown above and notify the Patent and Trademark Office of the change in status, or
B. If the status is the same, pay the FEE DUE shown above.

If the SMALL ENTITY is shown as NO:

A. Pay FEE DUE shown above, or

B. File verified statement of Small Entity Status before, or with, payment of 1/2 the FEE DUE shown above.

II. Part B-Issue Fee Transmittal should be completed and returned to the Patent and Trademark Office (PTO) with your ISSUE FEE. Even if the ISSUE FEE has already been paid by charge to deposit account, Part B Issue Fee Transmittal should be completed and returned. If you are charging the ISSUE FEE to your deposit account, section "4b" of Part B-Issue Fee Transmittal should be completed and an extra copy of the form should be submitted.

III. All communications regarding this application must give application number and batch number.
Please direct all communications prior to issuance to Box ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PATENT AND TRADEMARK OFFICE COPY

PTOL-85 (REV. 10-96) Approved for use through 06/30/99. (0651-0033)

B11 #22

I hereby certify that this correspondence is being deposited with the United States Postal Service on the date set forth below as First Class Mail in an envelope addressed to: Assistant Commissioner for Patents, Washington D.C. 20231.

Date of Signature
and Deposit: 5/30/01 [signature]
Michael A. Jaskolski, Reg. No. 37,551

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Carlos De La Huerga
Serial No.:	09/512,125
Filed:	February 24, 2000
Title:	METHOD AND SYSTEM FOR AUTOMATED DATA STORAGE AND RETRIEVAL
Examiner:	F. Coby
Art Unit:	2171
Docket:	250591.90163

SUBMISSION OF FORMAL DRAWINGS

Attention: Official Draftsperson

Commissioner for Patents
Washington DC 20231

Dear Sir:

Enclosed you will find a copy of new formal drawings being submitted in the above-referenced case. Please contact me if you have any questions or need any other information.

Respectfully submitted,

CARLOS DE LA HUERGA

Date: May 30, 2001

[signature]

Michael A. Jaskolski
Quarles & Brady, LLP
Reg. No. 37,551
Attorney for Applicant
411 East Wisconsin Avenue
Milwaukee WI 53202
414/277-5711

QBMKE\5064444.1

04/01

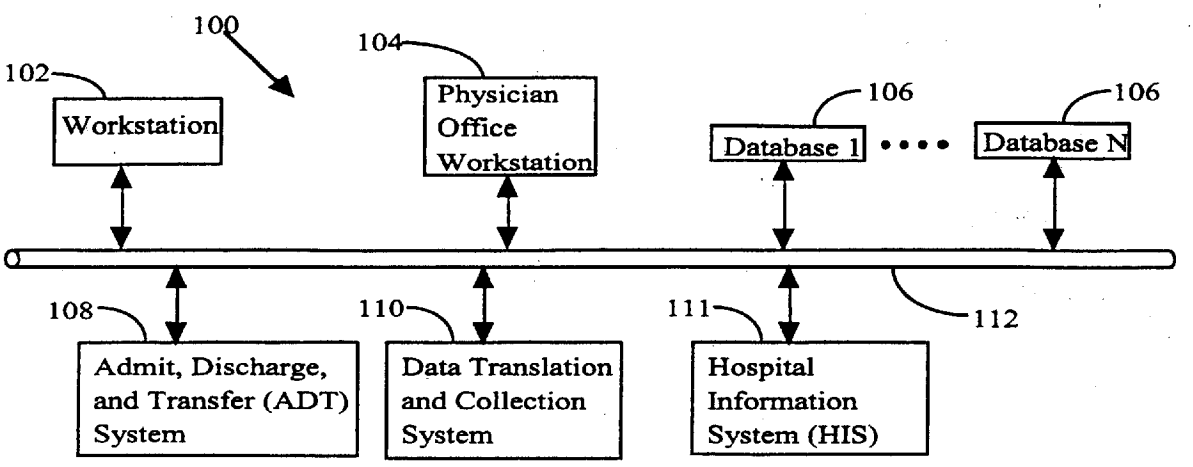


Figure 1

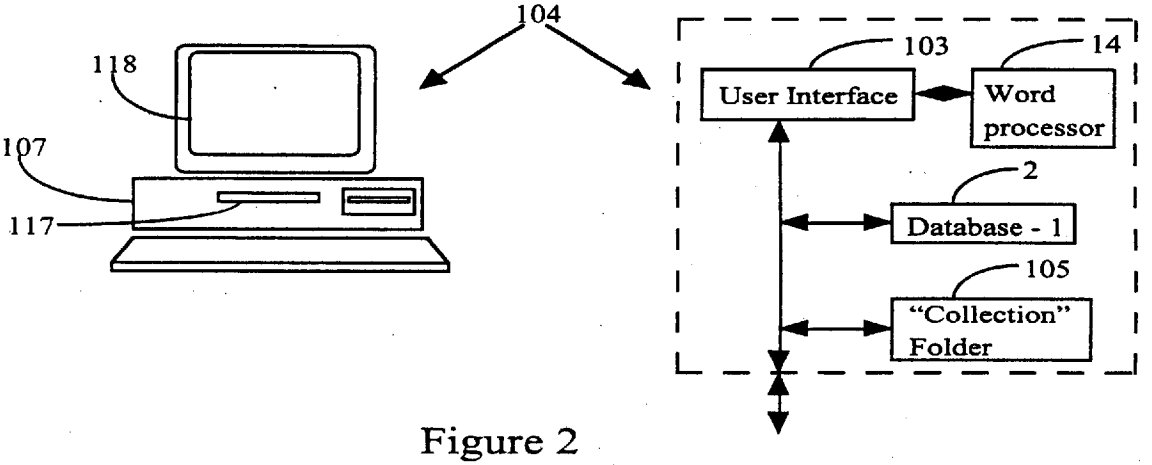


Figure 2

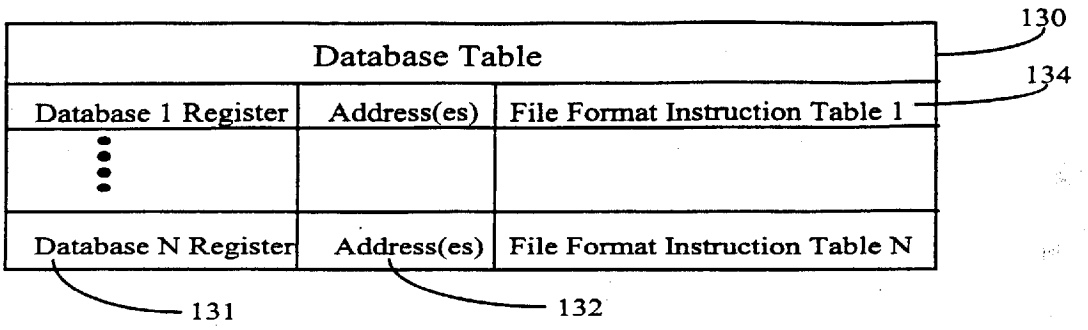


Figure 3A

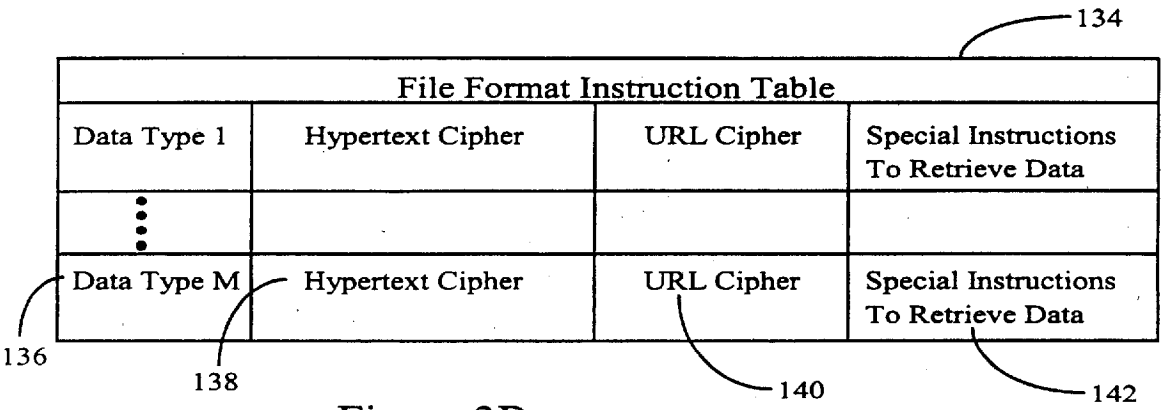


Figure 3B

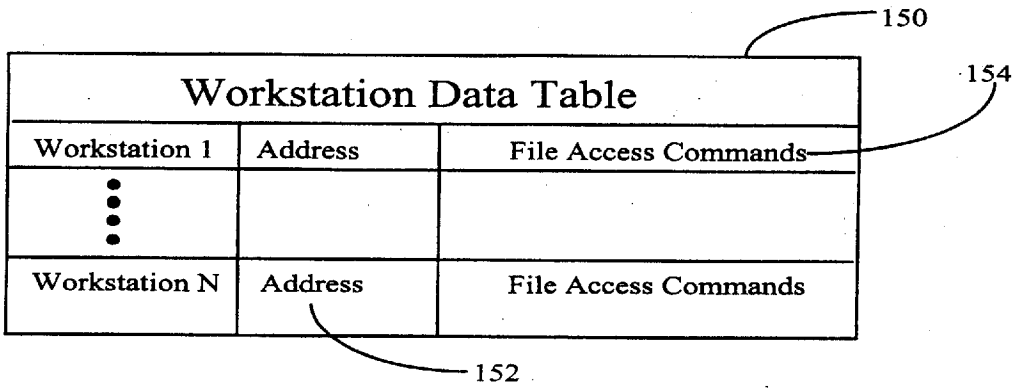


Figure 4A

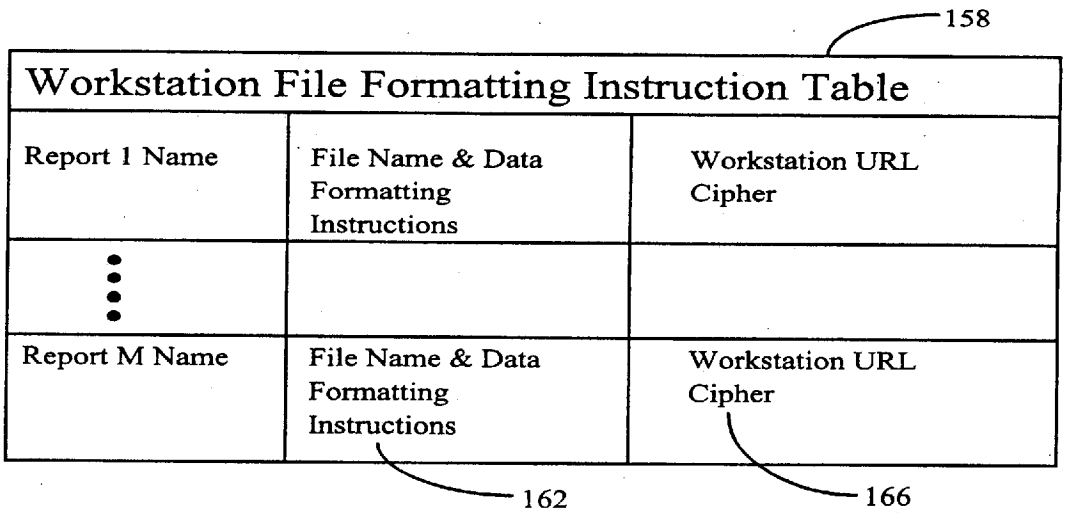


Figure 4B

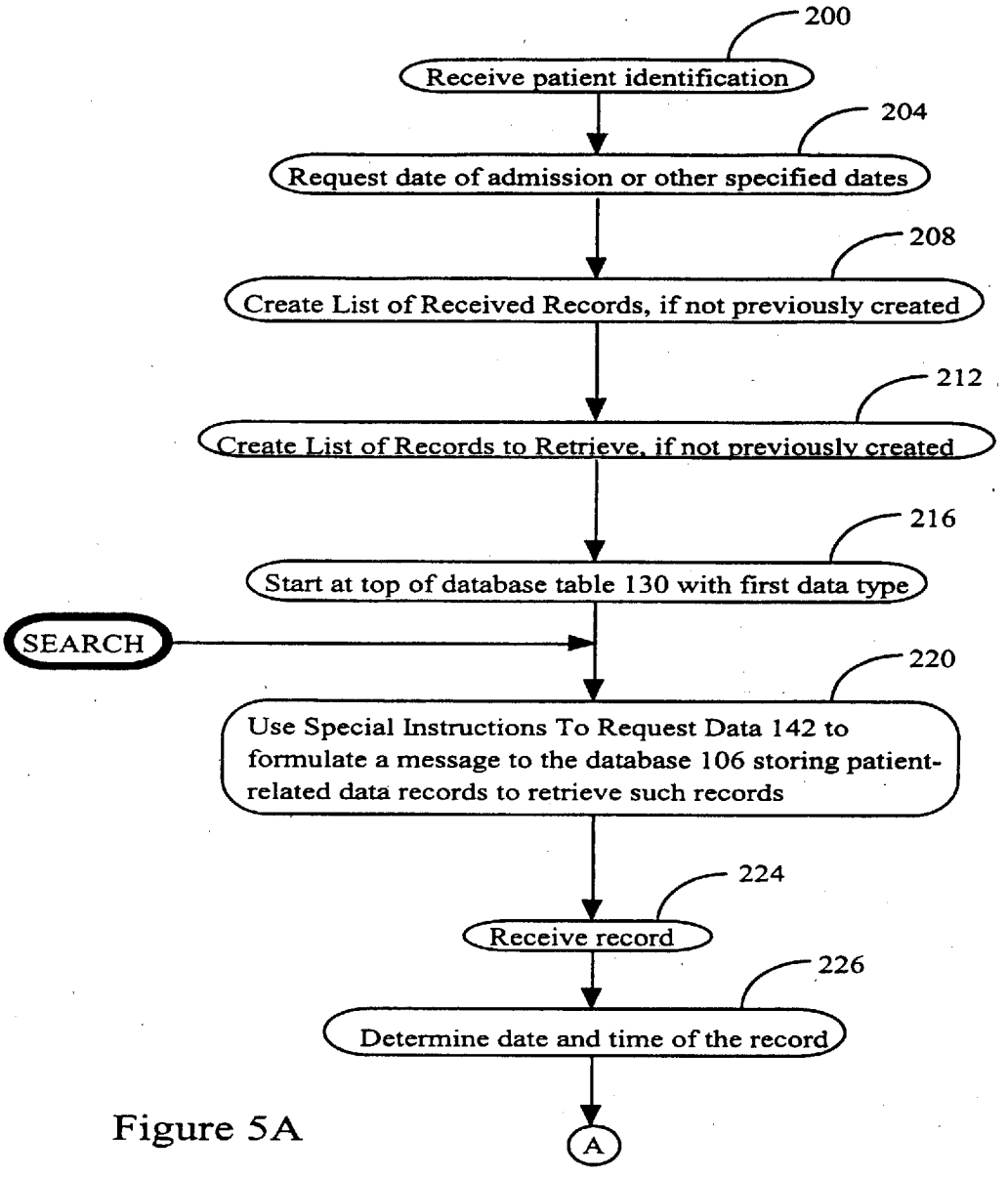

200
Receive patient identification
204
Request date of admission or other specified dates
208
Create List of Received Records, if not previously created
212
Create List of Records to Retrieve, if not previously created
216
Start at top of database table 130 with first data type
SEARCH
220
Use Special Instructions To Request Data 142 to formulate a message to the database 106 storing patient-related data records to retrieve such records
224
Receive record
226
Determine date and time of the record
A

Figure 5A

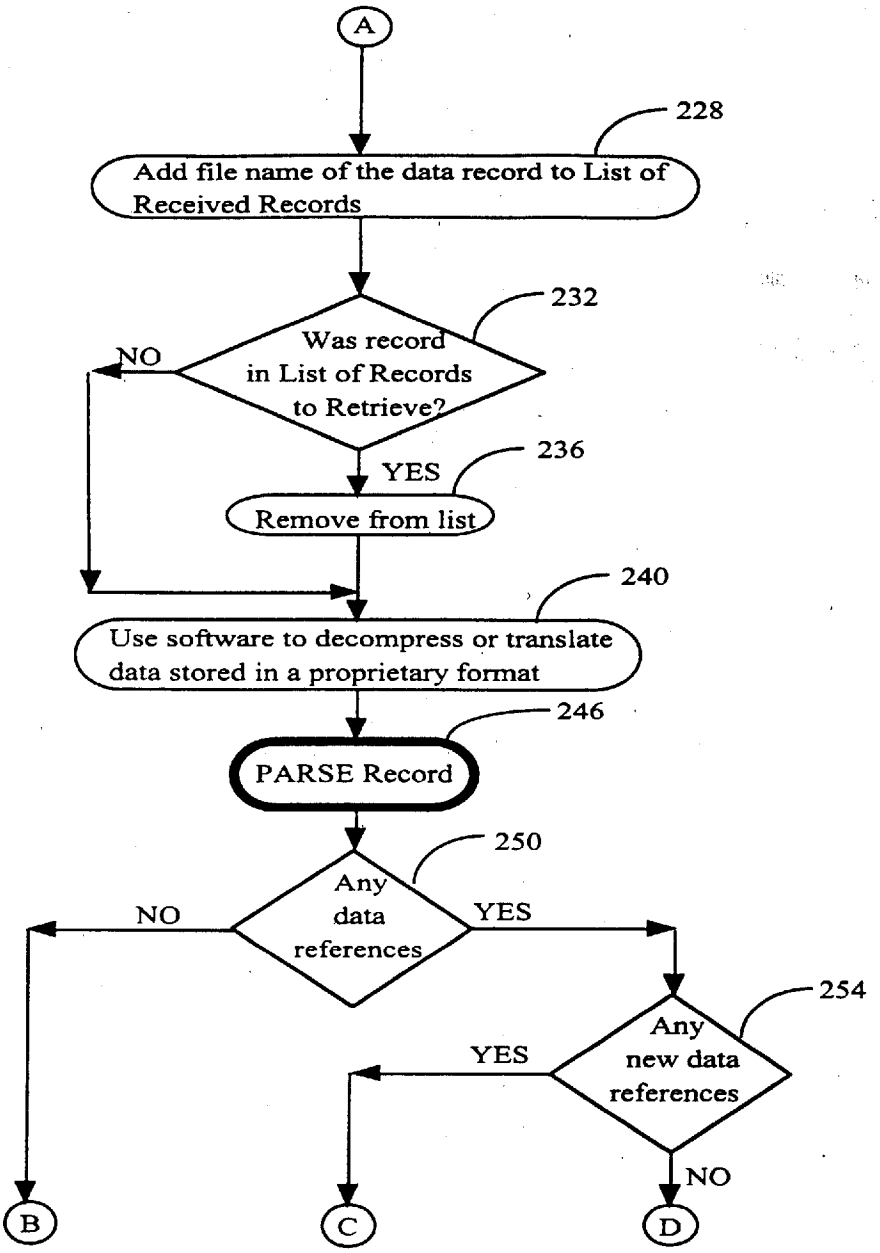

A
228
Add file name of the data record to List of Received Records
232
Was record in List of Records to Retrieve?
NO
YES
236
Remove from list
240
Use software to decompress or translate data stored in a proprietary format
246
PARSE Record
250
Any data references
NO
YES
254
Any new data references
YES
NO
B
C
D

Figure 5B

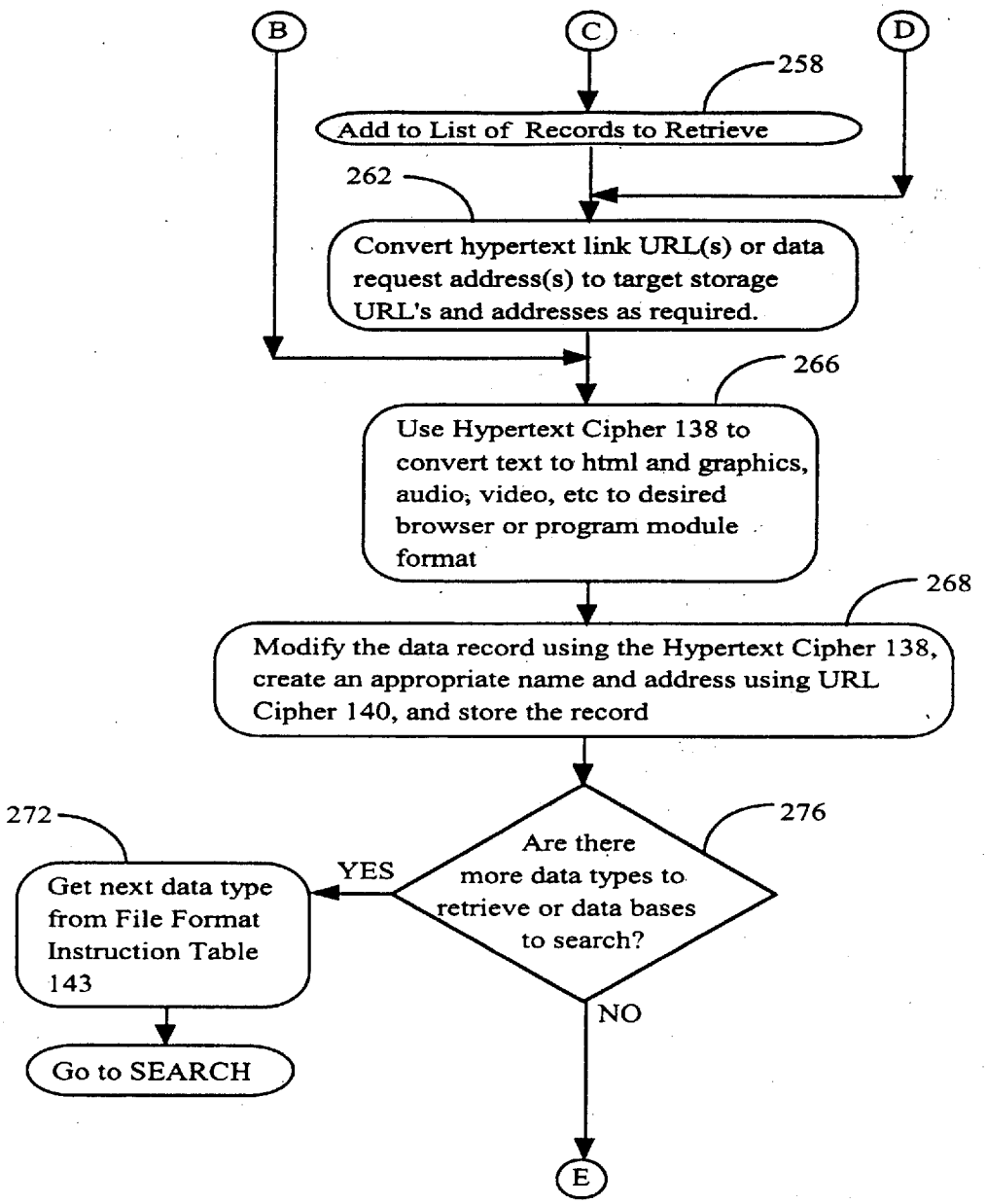

B
C
D
258
Add to List of Records to Retrieve
262
Convert hypertext link URL(s) or data request address(s) to target storage URL's and addresses as required.
266
Use Hypertext Cipher 138 to convert text to html and graphics, audio, video, etc to desired browser or program module format
268
Modify the data record using the Hypertext Cipher 138, create an appropriate name and address using URL Cipher 140, and store the record
272
Get next data type from File Format Instruction Table 143
YES
Are there more data types to retrieve or data bases to search?
276
NO
Go to SEARCH
E

Figure 5C

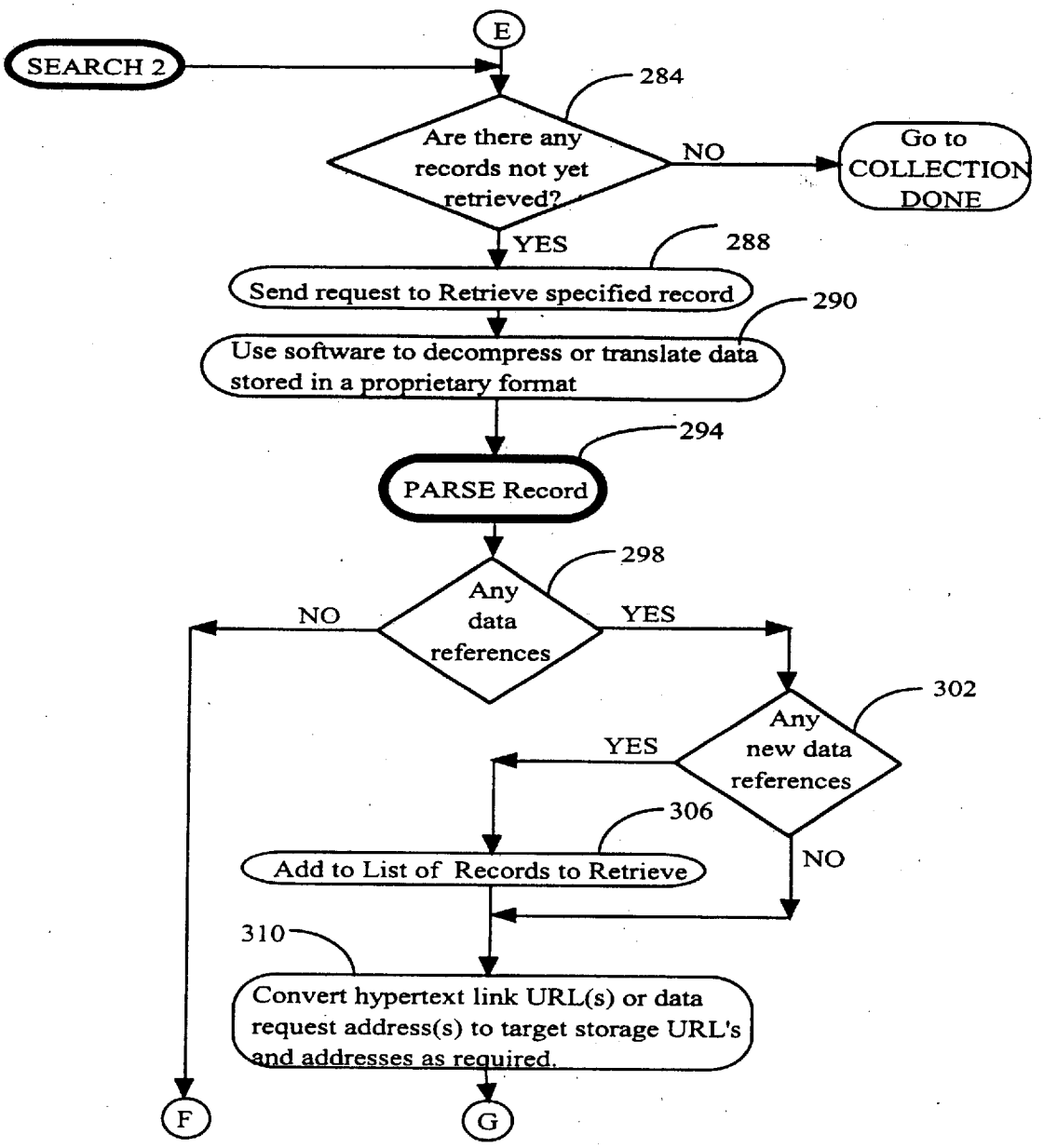

E
SEARCH 2
284
Are there any records not yet retrieved?
NO
Go to COLLECTION DONE
YES
288
Send request to Retrieve specified record
290
Use software to decompress or translate data stored in a proprietary format
294
PARSE Record
298
Any data references
NO
YES
302
Any new data references
YES
306
Add to List of Records to Retrieve
NO
310
Convert hypertext link URL(s) or data request address(s) to target storage URL's and addresses as required.
F
G

Figure 5D

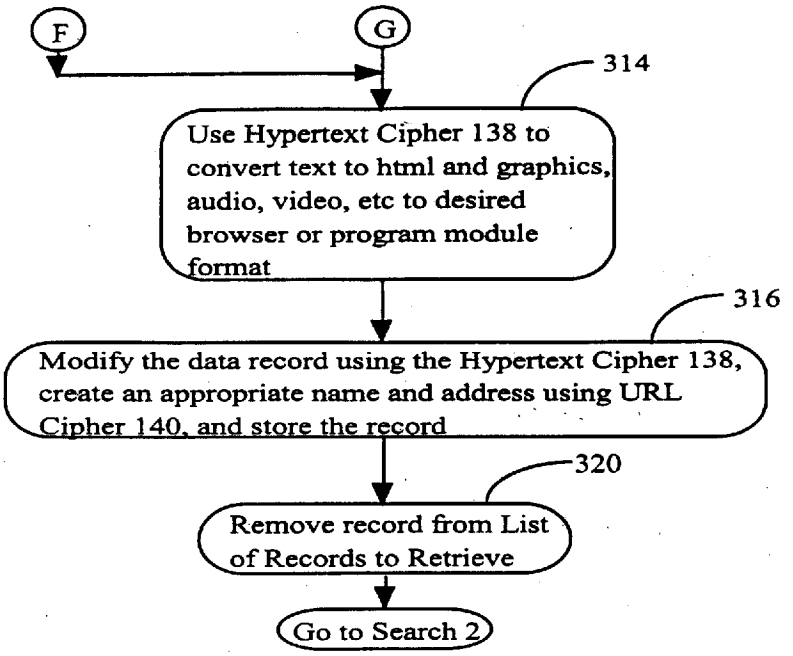


Figure 5E

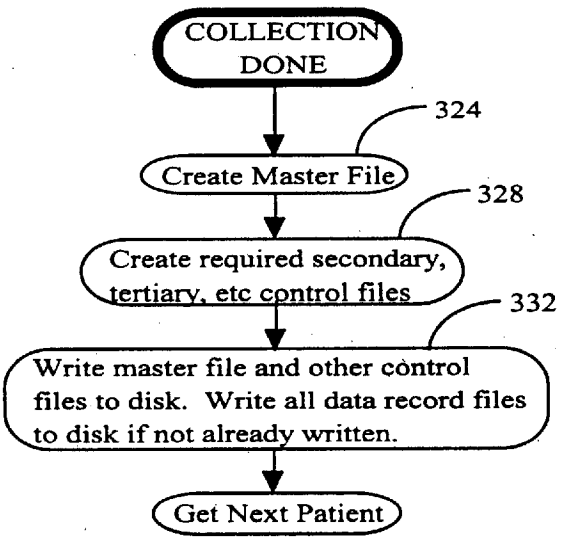


Figure 5F

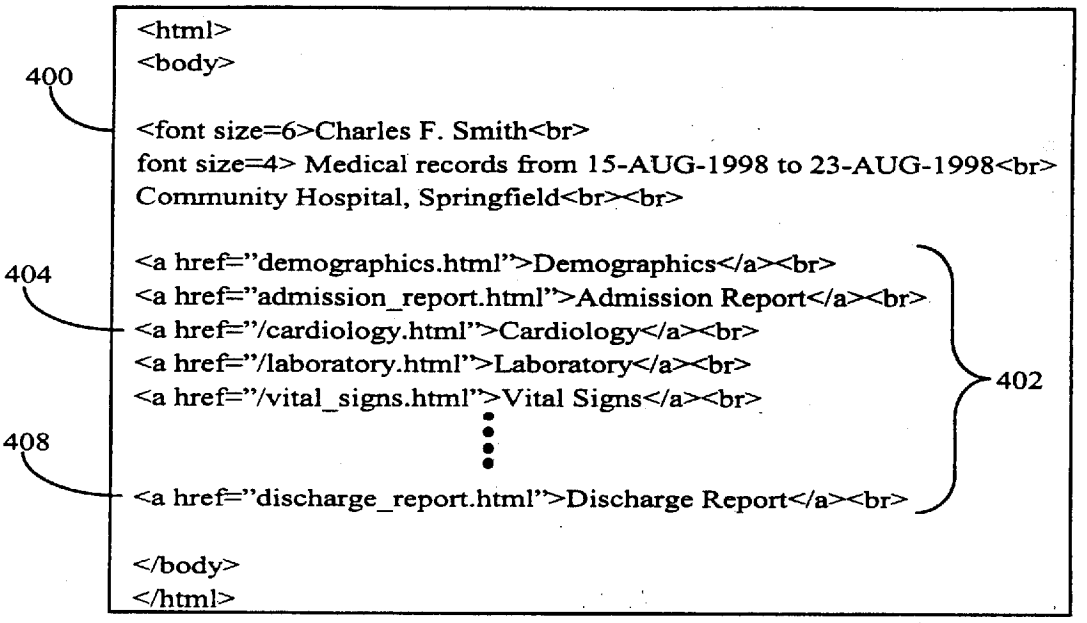


Figure 6A

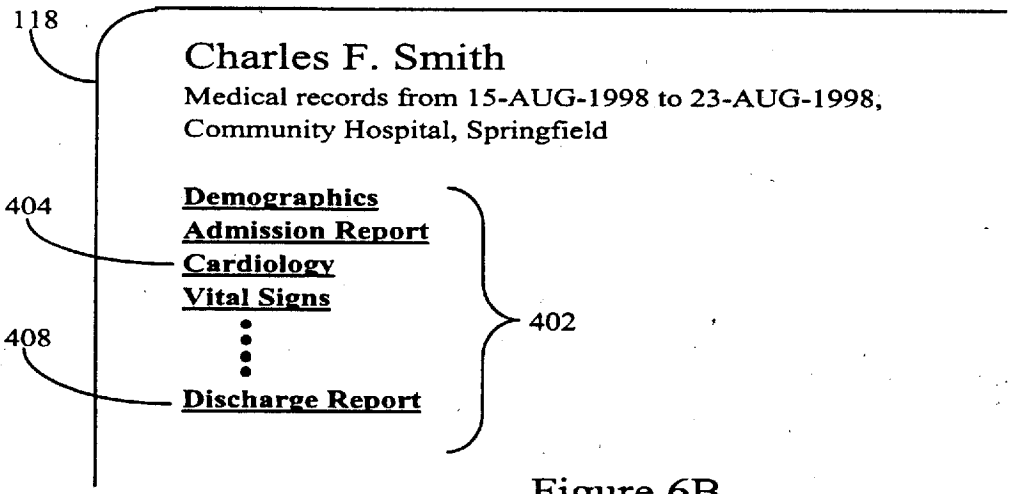


Figure 6B

412

```
<html>
<body>
DISCHARGE SUMMARY for Charles F. Smith<br><br>

Date of Admission: 15-AUG-98<br>
Date of Discharge: 17-AUG-98<br><br>

HISTORY OF PRESENT ILLNESS:<br>
Patient #1 is a 47-year-old male admitted for further evaluation of recent exertional
angina and abnormal exercise test.<br><br>

HOSPITAL COURSE:<br>
The patient was admitted to the Telemetry unit and underwent <a
href="charles_f_smith/cardiology/cath/1998-08-15/10:25/report.html">cardiac
catheterization on August 15, 1998</a>. Catheterization demonstrated normal
ventricular function without evidence for prior infarction. The coronary arteriogram
showed moderate stenosis throughout the mid and distal portions of the left anterior
descending artery and diagonal branch, as well as ••••
```

416

Figure 7A

118

416

DISCHARGE SUMMARY for Charles F. Smith

Date of Admission: 15-AUG-98
Date of Discharge: 17-AUG-98

HISTORY OF PRESENT ILLNESS:
Patient #1 is a 47-year-old male admitted for further evaluation of recent exertional angina and abnormal exercise test.

HOSPITAL COURSE:
The patient was admitted to the Telemetry unit and underwent **catheterization on August 15, 1998**. Catheterization demonstrated normal ventricular function without evidence for prior infarction. The coronary arteriogram showed moderate stenosis throughout the mid and distal portions of the left anterior descending artery and diagonal branch, as well as ••••

Figure 7B

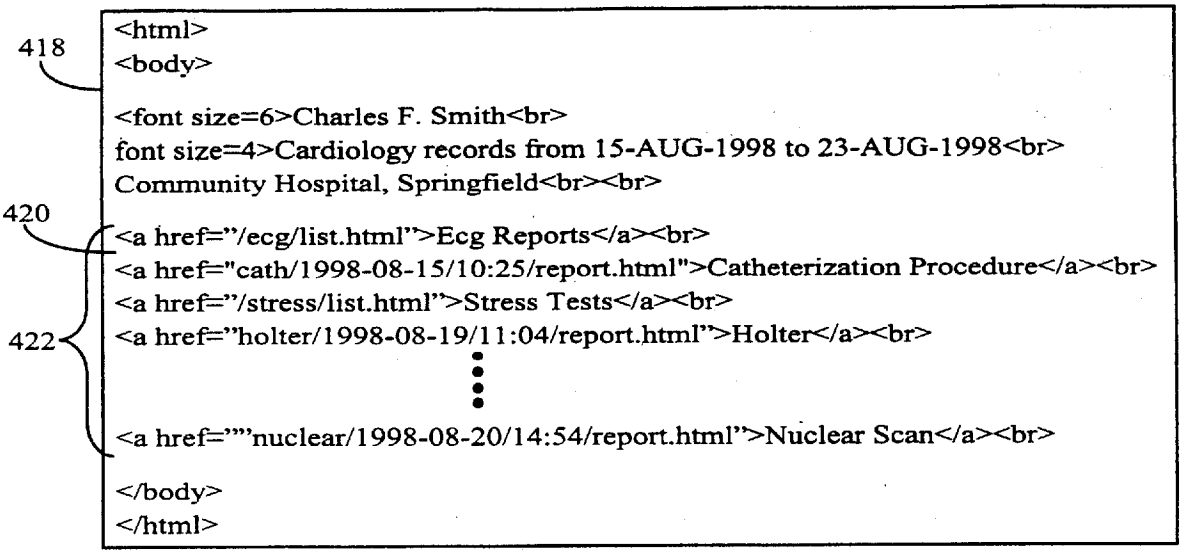


Figure 8A

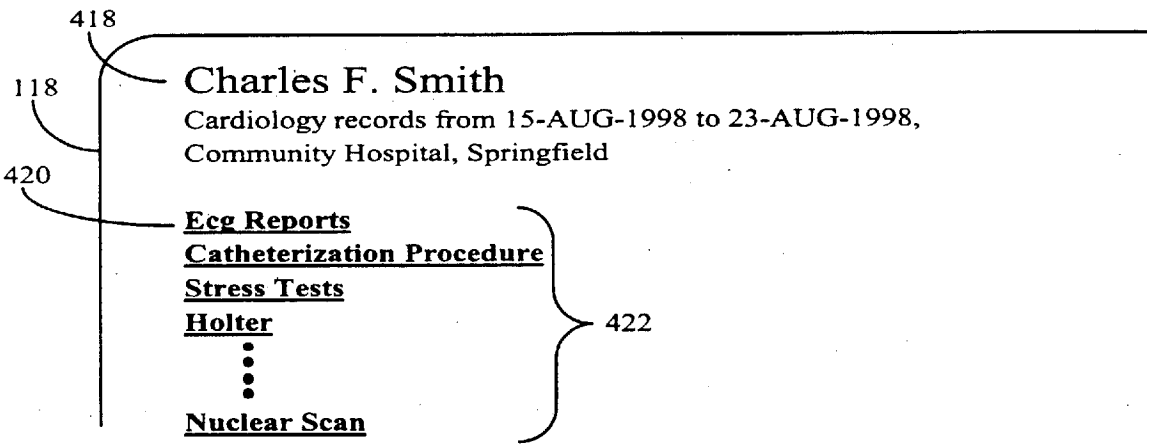


Figure 8B

424

```
<html>
<body>

<font size=6>Charles F. Smith<br>
font size=4>ECG records from 15-AUG-1998 to 23-AUG-1998<br>
Community Hospital, Springfield<br><br>

<a href="ecg/1998-08-15/09:15/report.html">15-AUG-1998
09:15</a><br>
<a href="ecg/1998-08-15/16:40/report.html">15-AUG-1998
14:40</a><br>
<a href="ecg/1998-08-17/11:03/report.html">17-AUG-1998
11:03</a><br>
<a href="ecg/1998-08-19/10:25/report.html">19-AUG-1998
09:15</a><br>
                .
                .
                .
<a href="ecg/1998-08-23/08:14/report.html">23-AUG-1998
08:14</a><br>

</body>
</html>
```

426

Figure 9A

118

Charles F. Smith

Ecg records from 15-AUG-1998 to 23-AUG-1998,
Community Hospital, Springfield

426

15-AUG-1998 09:15
15-AUG-1998 14:40
17-AUG-1998 11:03
19-AUG-1998 09:15
.
.
.
23-AUG-1998 08:14

Figure 9B

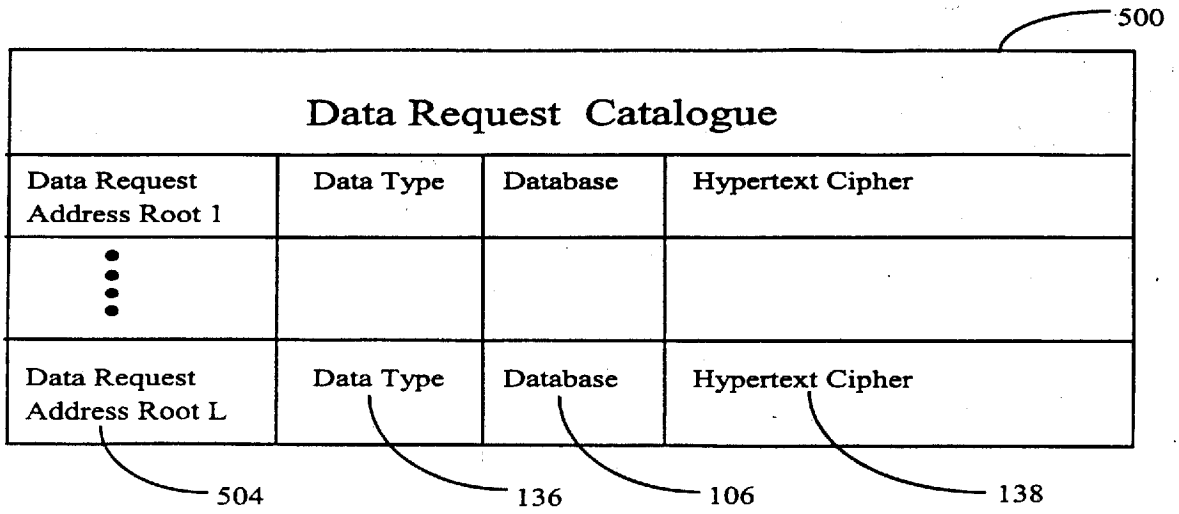


Figure 10

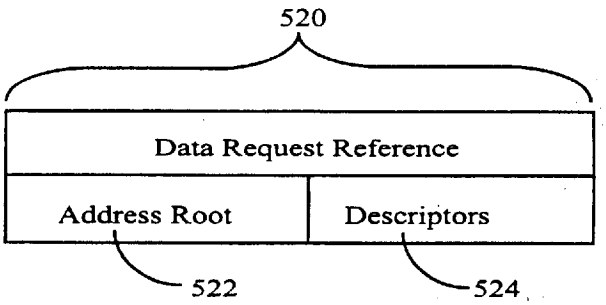


Figure 11

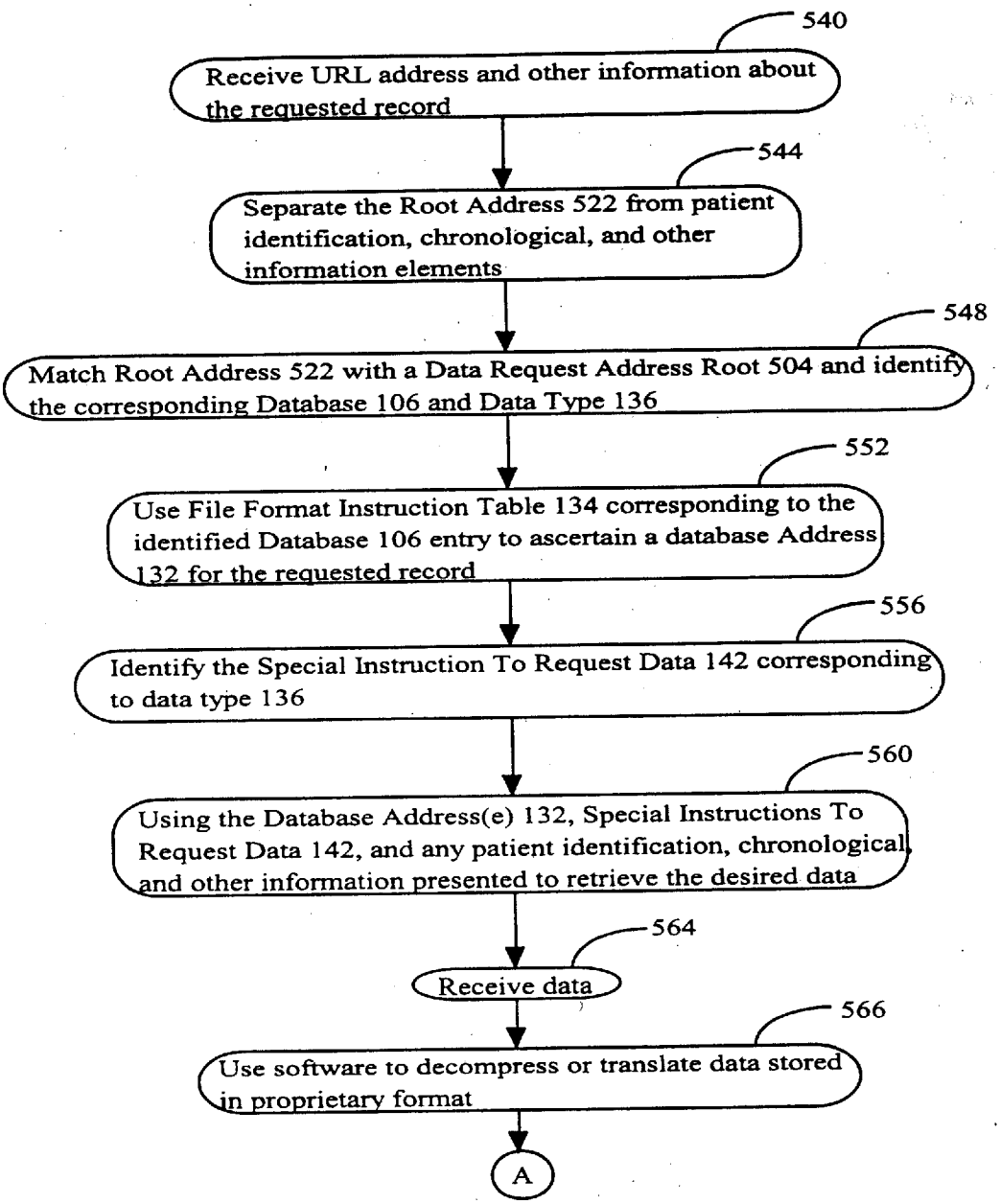

540
Receive URL address and other information about the requested record
544
Separate the Root Address 522 from patient identification, chronological, and other information elements
548
Match Root Address 522 with a Data Request Address Root 504 and identify the corresponding Database 106 and Data Type 136
552
Use File Format Instruction Table 134 corresponding to the identified Database 106 entry to ascertain a database Address 132 for the requested record
556
Identify the Special Instruction To Request Data 142 corresponding to data type 136
560
Using the Database Address(e) 132, Special Instructions To Request Data 142, and any patient identification, chronological, and other information presented to retrieve the desired data
564
Receive data
566
Use software to decompress or translate data stored in proprietary format
A

Figure 12A

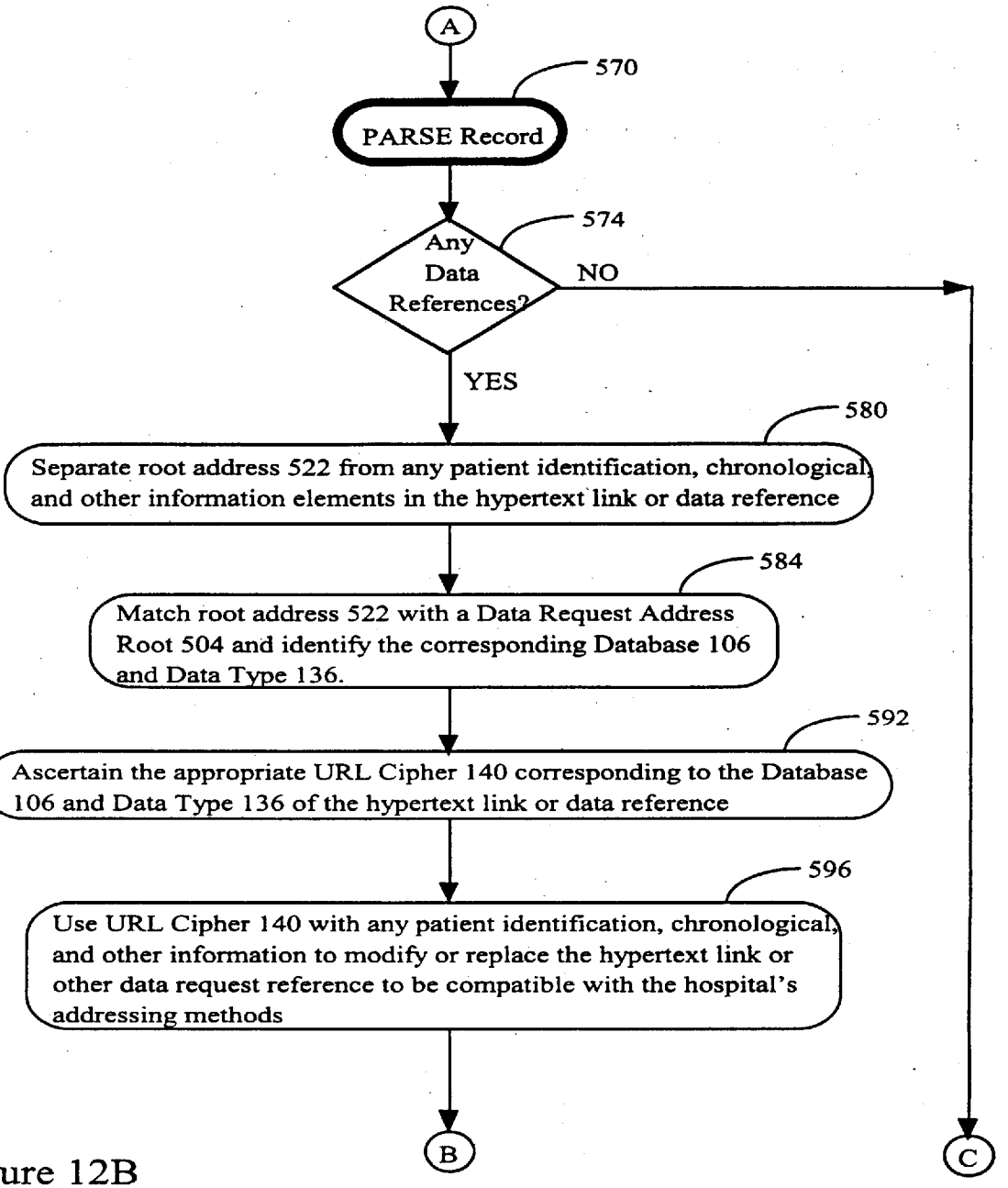

A
570
PARSE Record
574
Any Data References?
NO
YES
580
Separate root address 522 from any patient identification, chronological and other information elements in the hypertext link or data reference
584
Match root address 522 with a Data Request Address Root 504 and identify the corresponding Database 106 and Data Type 136.
592
Ascertain the appropriate URL Cipher 140 corresponding to the Database 106 and Data Type 136 of the hypertext link or data reference
596
Use URL Cipher 140 with any patient identification, chronological, and other information to modify or replace the hypertext link or other data request reference to be compatible with the hospital's addressing methods
B
C

Figure 12B

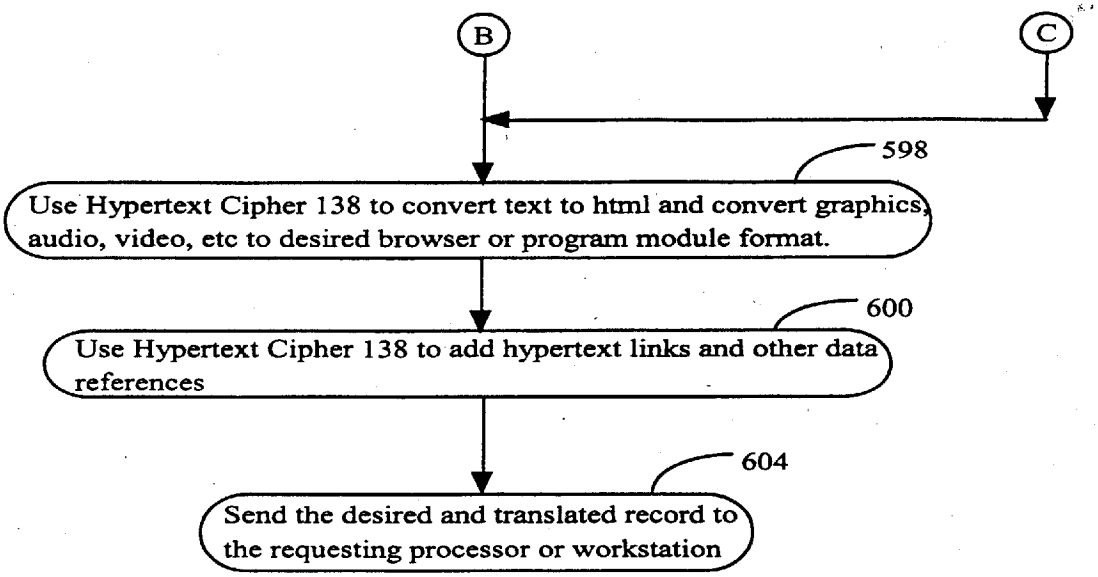

B
C
598
Use Hypertext Cipher 138 to convert text to html and convert graphics, audio, video, etc to desired browser or program module format.
600
Use Hypertext Cipher 138 to add hypertext links and other data references
604
Send the desired and translated record to the requesting processor or workstation

Figure 12C

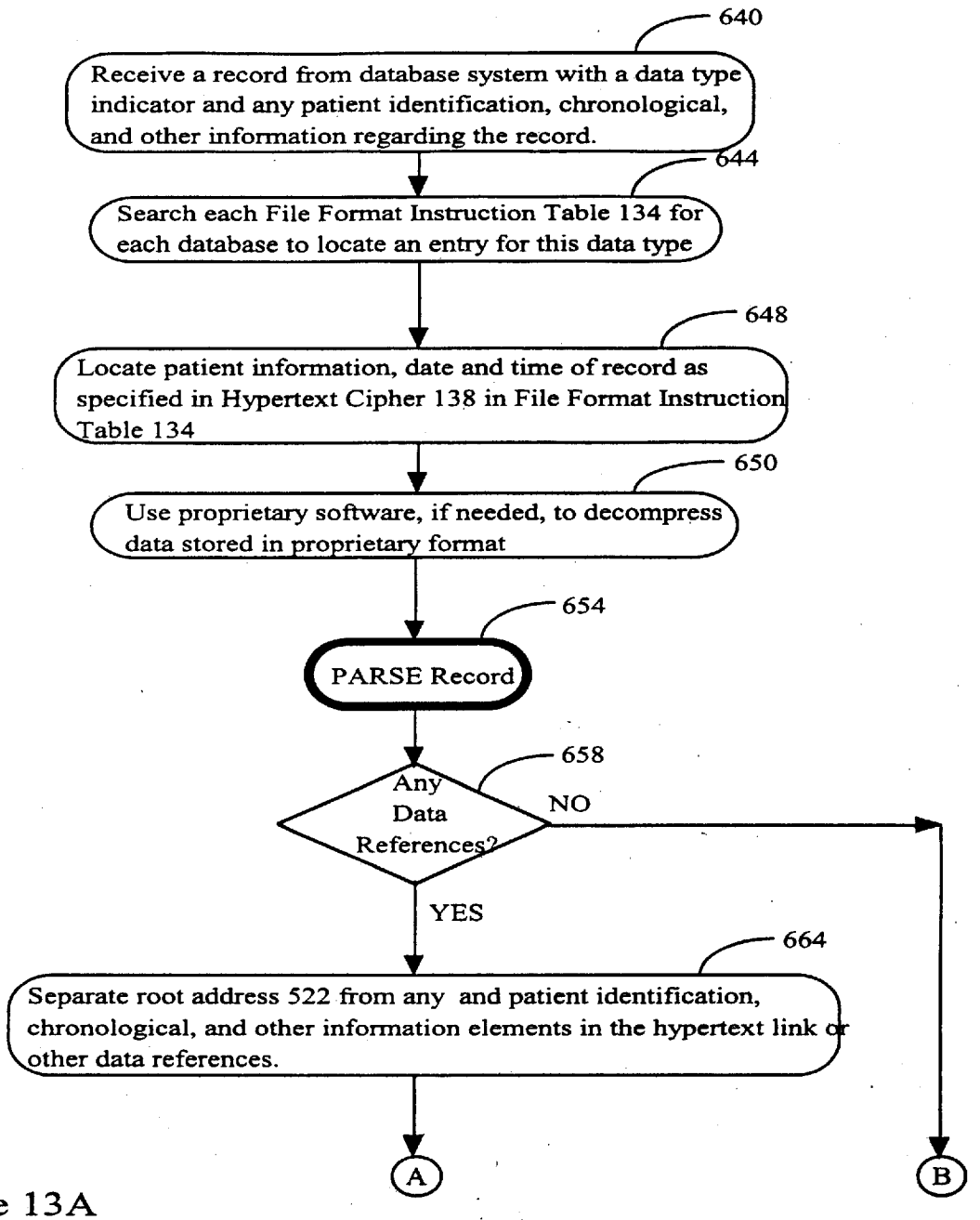

640
Receive a record from database system with a data type indicator and any patient identification, chronological, and other information regarding the record.
644
Search each File Format Instruction Table 134 for each database to locate an entry for this data type
648
Locate patient information, date and time of record as specified in Hypertext Cipher 138 in File Format Instruction Table 134
650
Use proprietary software, if needed, to decompress data stored in proprietary format
654
PARSE Record
658
Any Data References?
NO
YES
664
Separate root address 522 from any and patient identification, chronological, and other information elements in the hypertext link or other data references.
A
B

Figure 13A

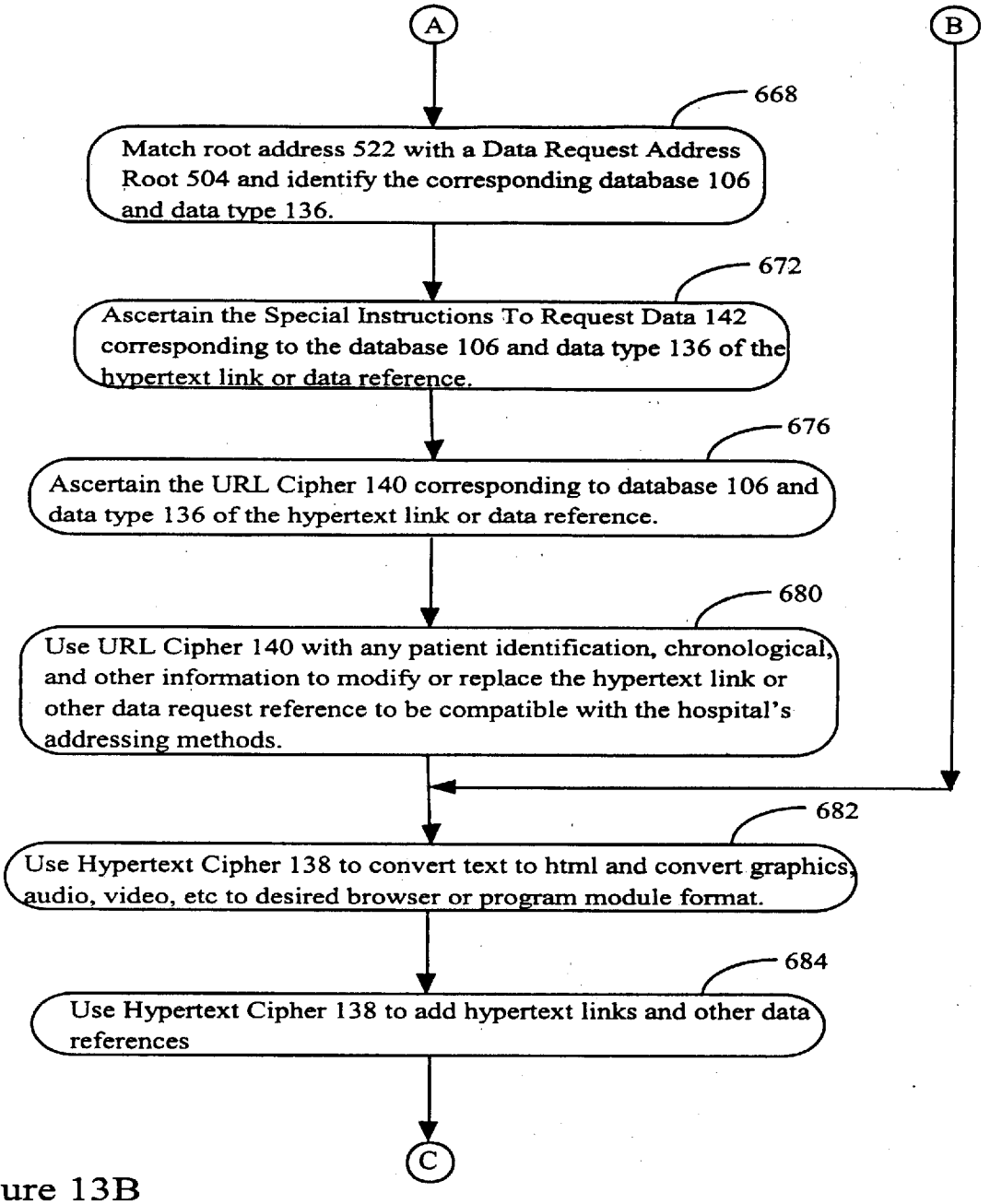

A
B
668
Match root address 522 with a Data Request Address Root 504 and identify the corresponding database 106 and data type 136.
672
Ascertain the Special Instructions To Request Data 142 corresponding to the database 106 and data type 136 of the hypertext link or data reference.
676
Ascertain the URL Cipher 140 corresponding to database 106 and data type 136 of the hypertext link or data reference.
680
Use URL Cipher 140 with any patient identification, chronological, and other information to modify or replace the hypertext link or other data request reference to be compatible with the hospital's addressing methods.
682
Use Hypertext Cipher 138 to convert text to html and convert graphics, audio, video, etc to desired browser or program module format.
684
Use Hypertext Cipher 138 to add hypertext links and other data references
C

Figure 13B

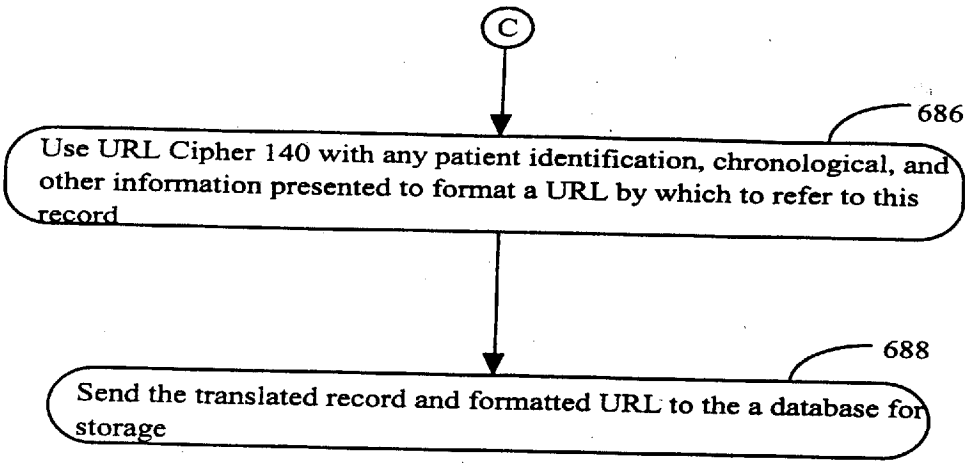

C
686
Use URL Cipher 140 with any patient identification, chronological, and other information presented to format a URL by which to refer to this record
688
Send the translated record and formatted URL to the a database for storage

Figure 13C

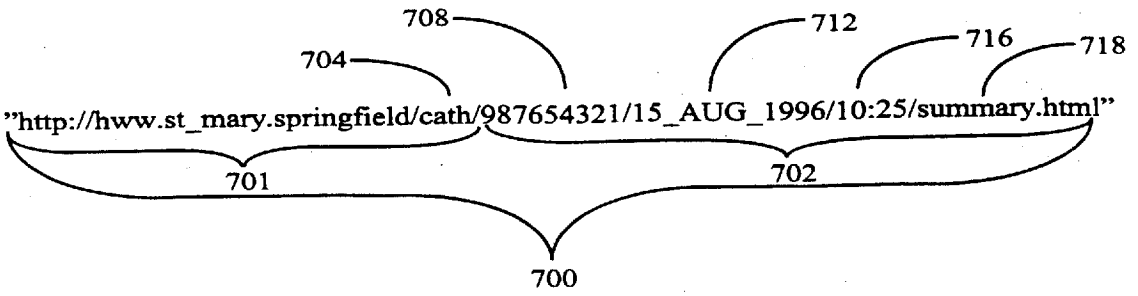


Figure 14A

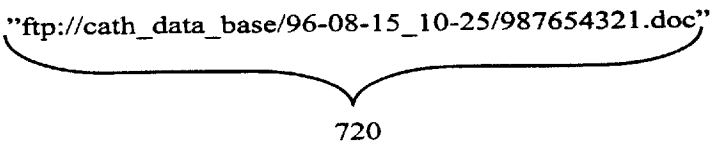


Figure 14B

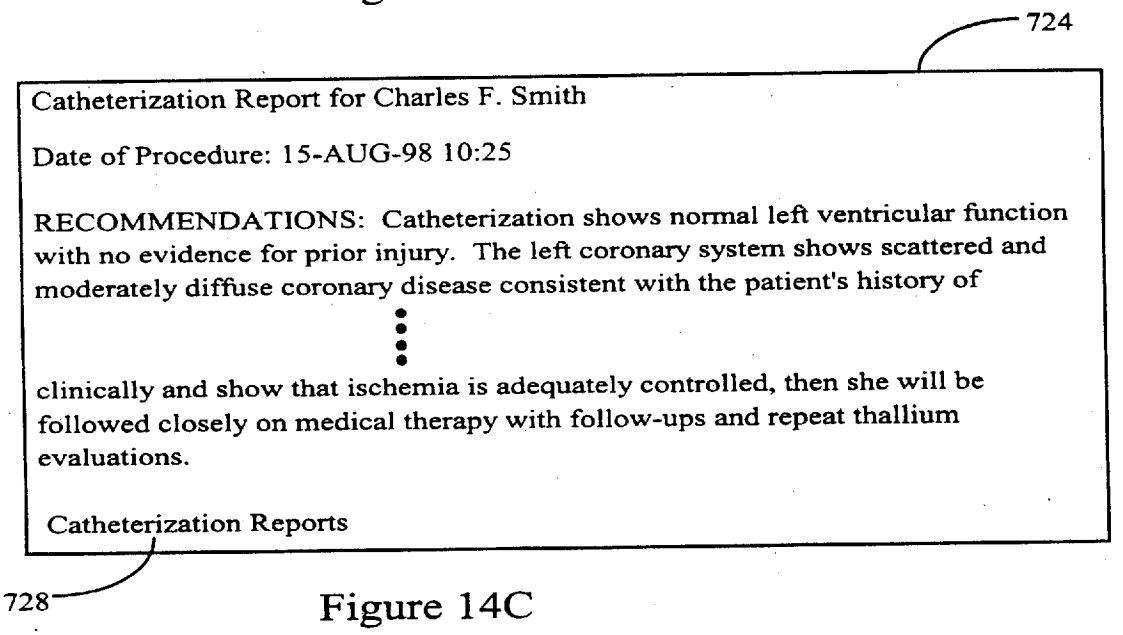
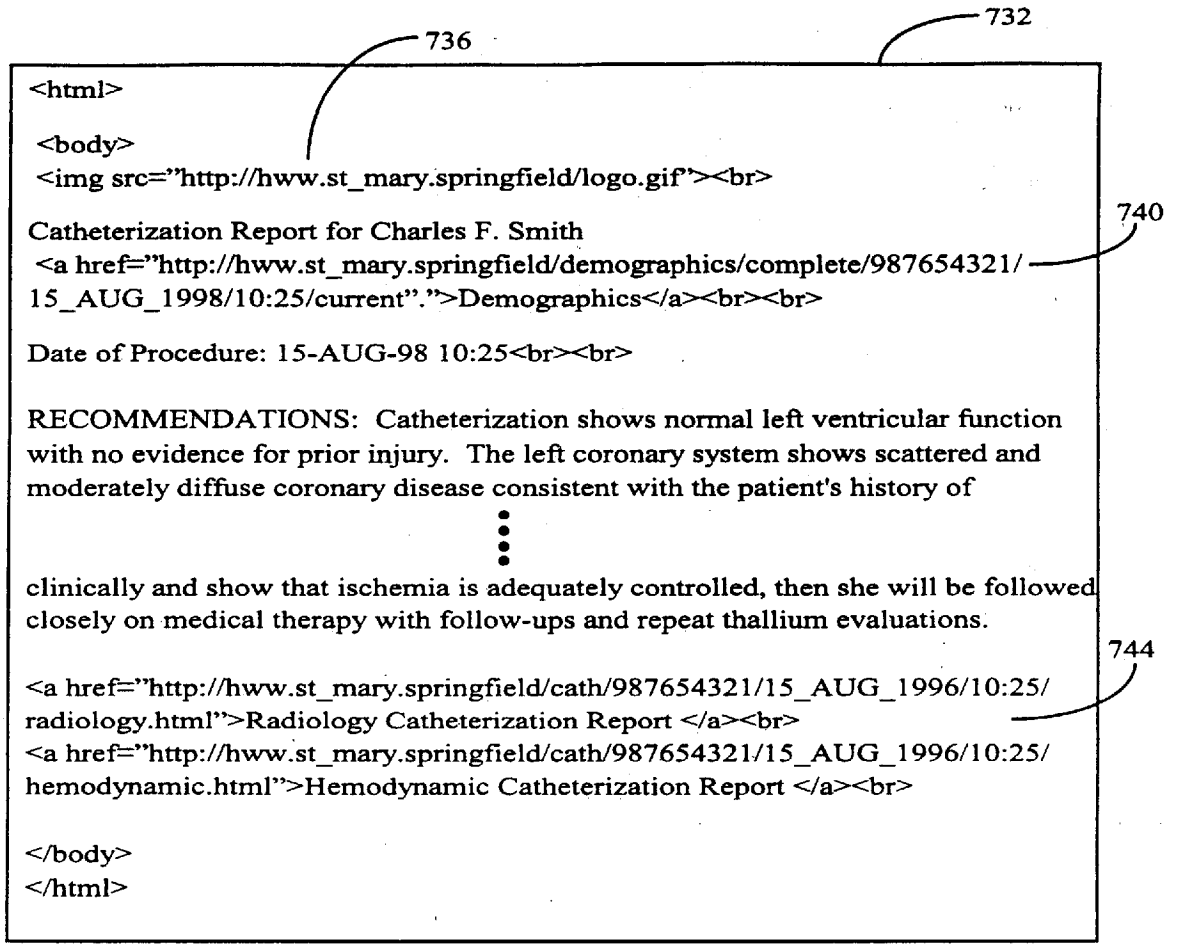


Figure 14C


736
732
740
744
<html>
<body>
<img src="http://hww.st_mary.springfield/logo.gif">
Catheterization Report for Charles F. Smith
Demographics

Date of Procedure: 15-AUG-98 10:25

RECOMMENDATIONS: Catheterization shows normal left ventricular function with no evidence for prior injury. The left coronary system shows scattered and moderately diffuse coronary disease consistent with the patient's history of
clinically and show that ischemia is adequately controlled, then she will be followed closely on medical therapy with follow-ups and repeat thallium evaluations.
Radiology Catheterization Report
Hemodynamic Catheterization Report
</body>
</html>

Figure 14D

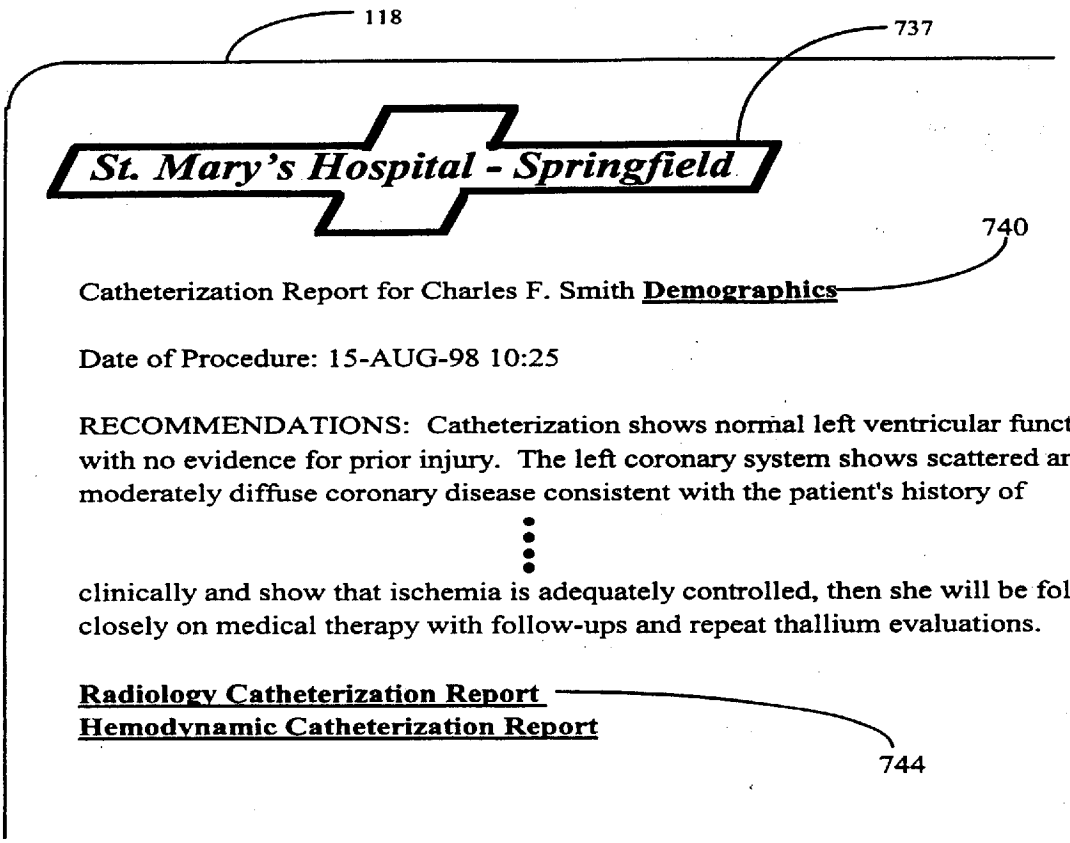

118
737
St. Mary's Hospital - Springfield
740
Catheterization Report for Charles F. Smith Demographics
Date of Procedure: 15-AUG-98 10:25
RECOMMENDATIONS: Catheterization shows normal left ventricular function with no evidence for prior injury. The left coronary system shows scattered and moderately diffuse coronary disease consistent with the patient's history of
clinically and show that ischemia is adequately controlled, then she will be followed closely on medical therapy with follow-ups and repeat thallium evaluations.
Radiology Catheterization Report
Hemodynamic Catheterization Report
744

Figure 14E

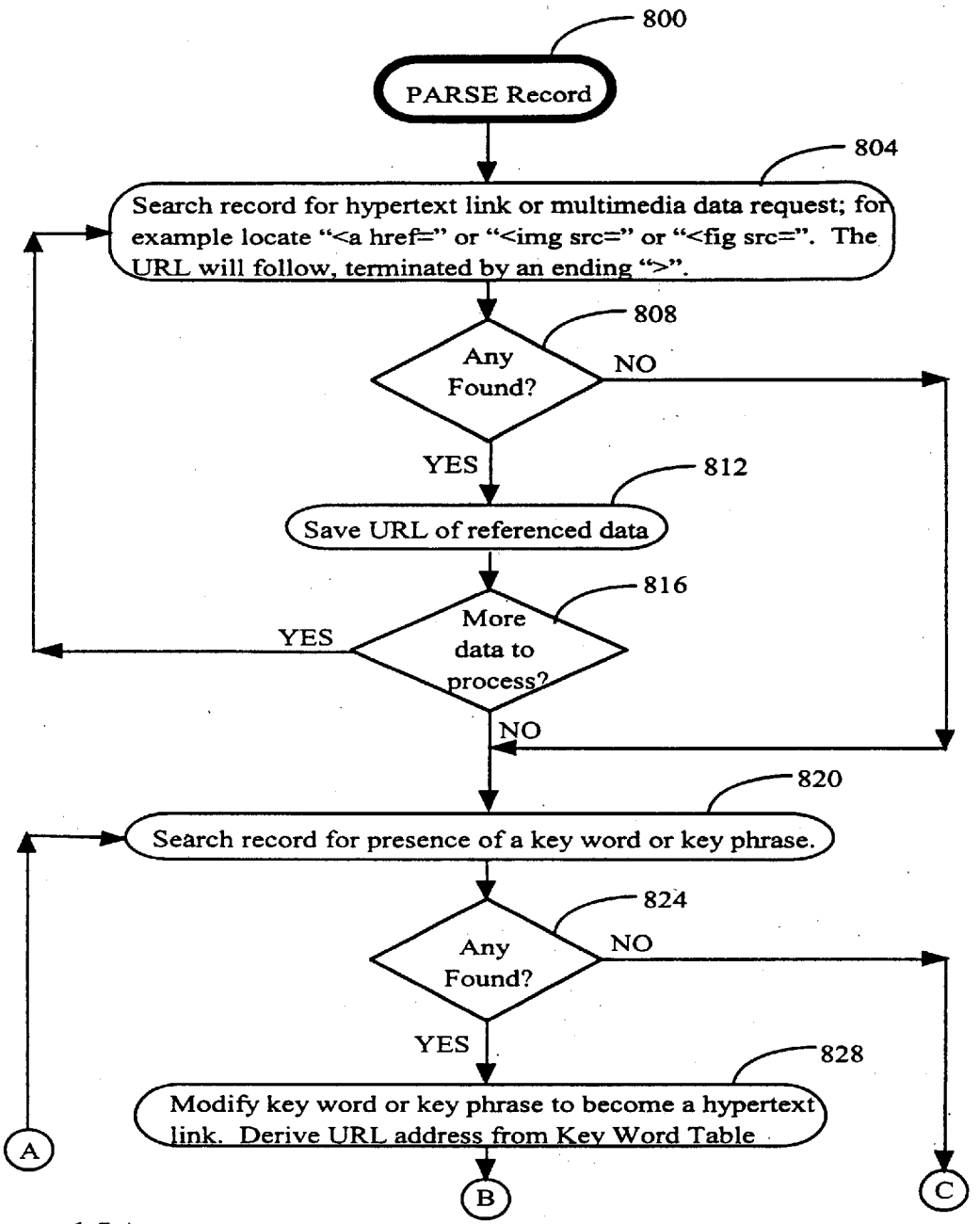

800
PARSE Record
804
Search record for hypertext link or multimedia data request; for example locate "<a href=" or "<img src=" or "<fig src=". The URL will follow, terminated by an ending ">".
808
Any Found?
NO
YES
812
Save URL of referenced data
816
More data to process?
YES
NO
820
Search record for presence of a key word or key phrase.
824
Any Found?
NO
YES
828
Modify key word or key phrase to become a hypertext link. Derive URL address from Key Word Table
A
B
C

Figure 15A

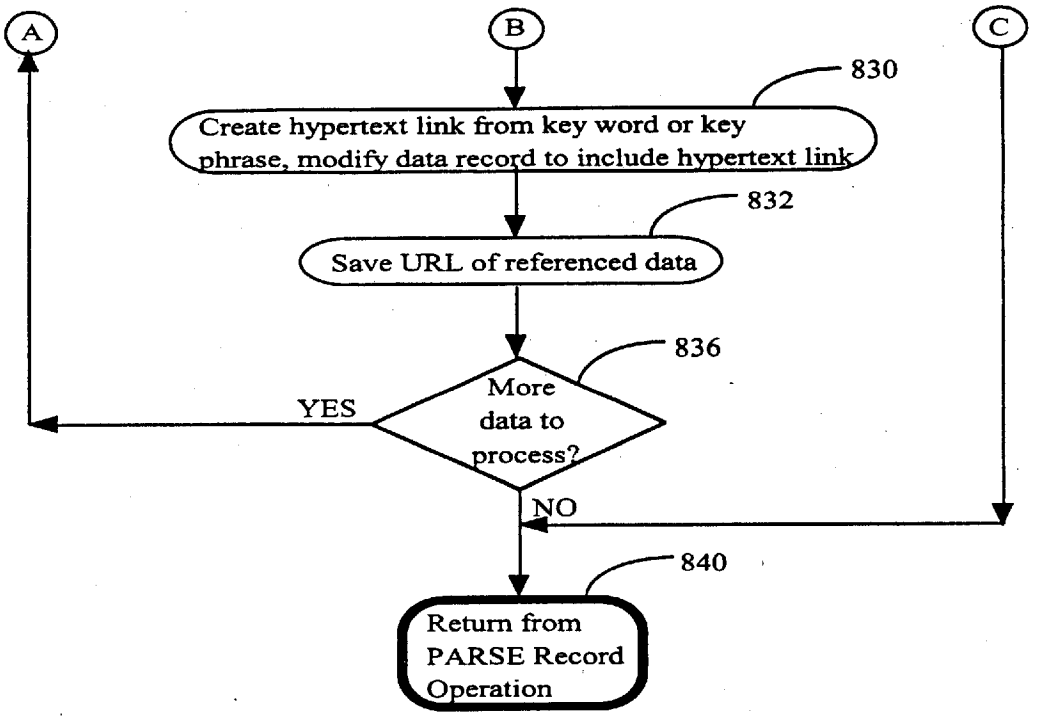

A
B
C
830
Create hypertext link from key word or key phrase, modify data record to include hypertext link
832
Save URL of referenced data
836
More data to process?
YES
NO
840
Return from PARSE Record Operation

Figure 15B

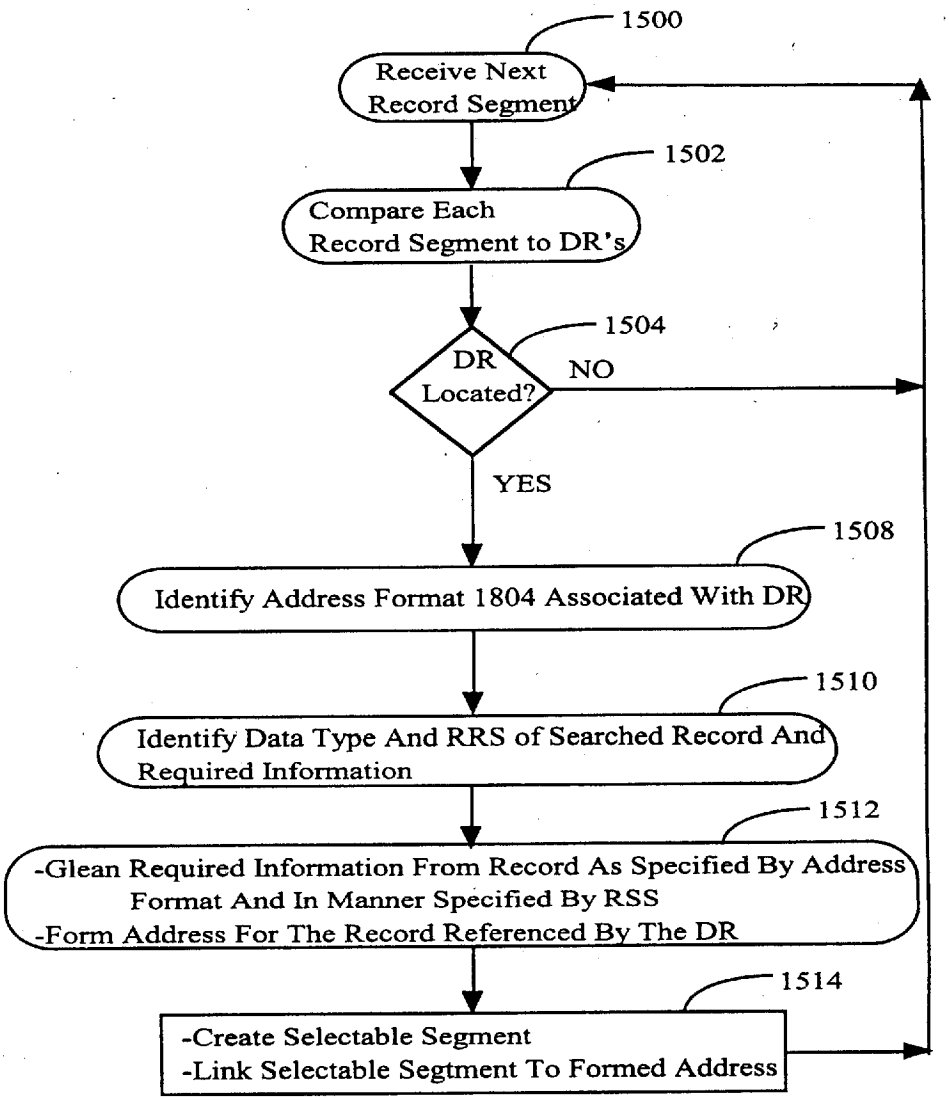

1500
Receive Next Record Segment
1502
Compare Each Record Segment to DR's
1504
DR Located?
NO
YES
1508
Identify Address Format 1804 Associated With DR
1510
Identify Data Type And RRS of Searched Record And Required Information
1512
-Glean Required Information From Record As Specified By Address Format And In Manner Specified By RSS
-Form Address For The Record Referenced By The DR
1514
-Create Selectable Segment
-Link Selectable Segtment To Formed Address

Figure 16

Text As It Appears in the Word Processor

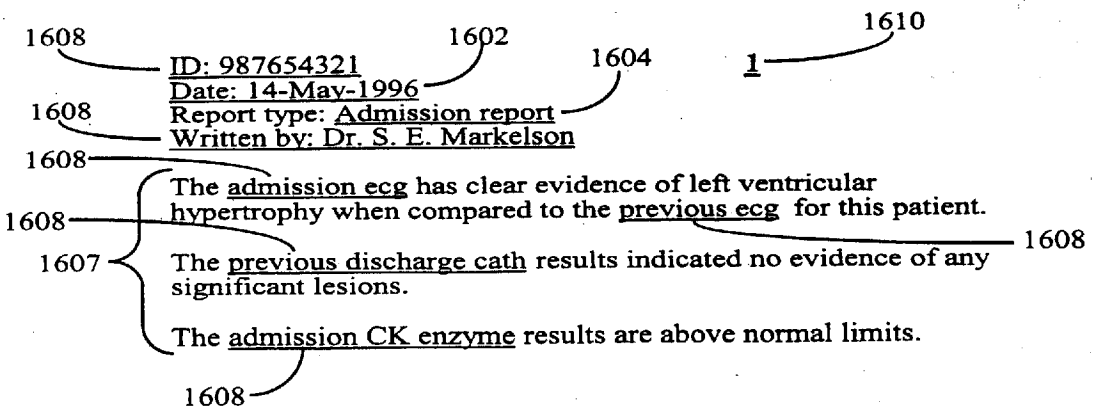


Figure 17

Text after Being Converted to HTML with Hypertext Links Added

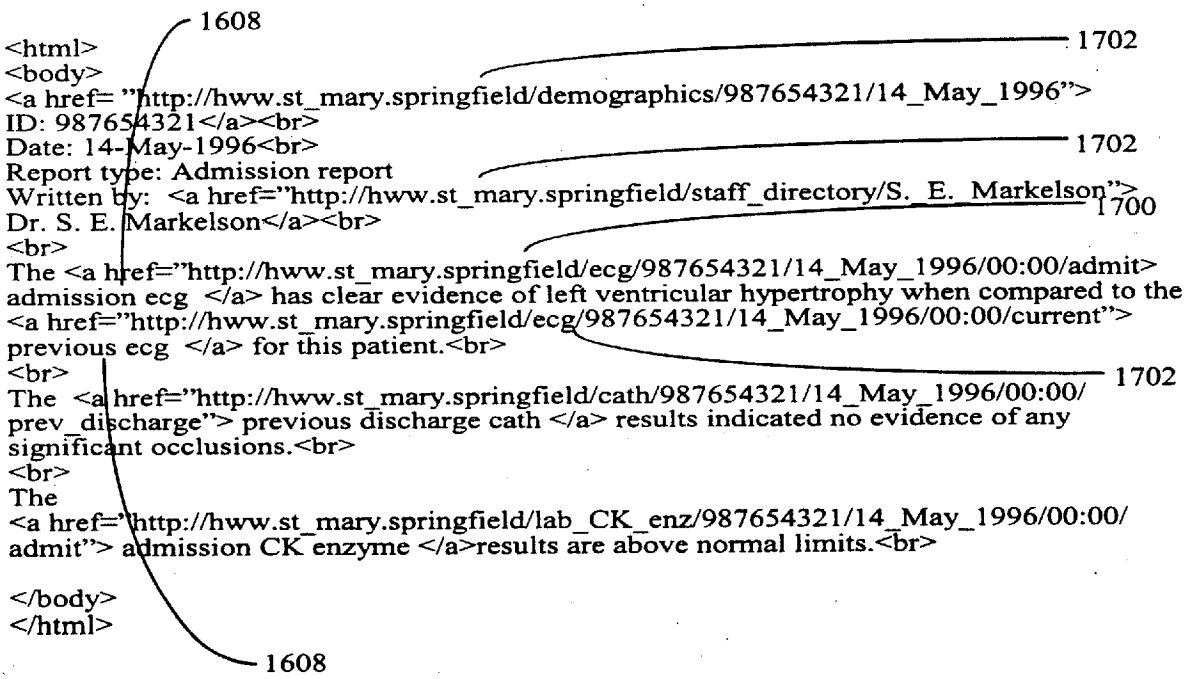


Figure 18

Text As Viewed via Word Processor 14 or Browser

ID: 987654321
Date: 14-May-1996
Report type: Admission report
Written by: Dr. S. E. Markelson

The admission ecg has clear evidence of left ventricular hypertrophy when compared to the current ecg for this patient.

The previous discharge cath results indicated no evidence of any significant lesions.

The admission CK enzyme results are above normal limits.

Figure 19

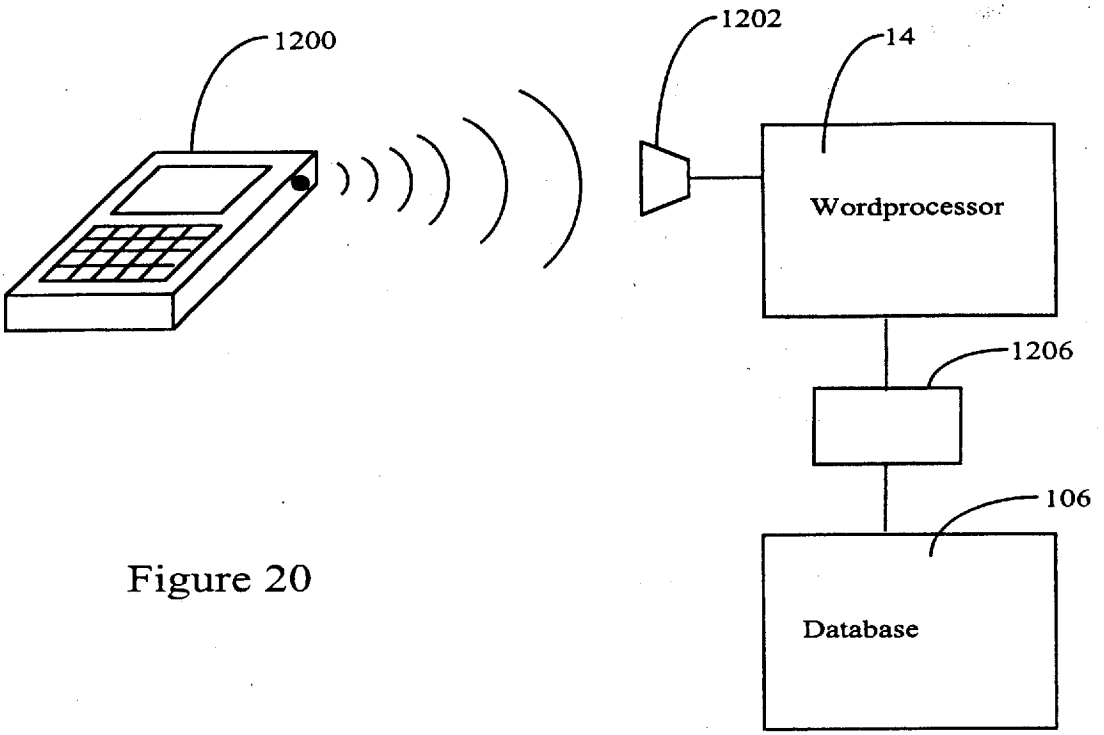


Figure 20

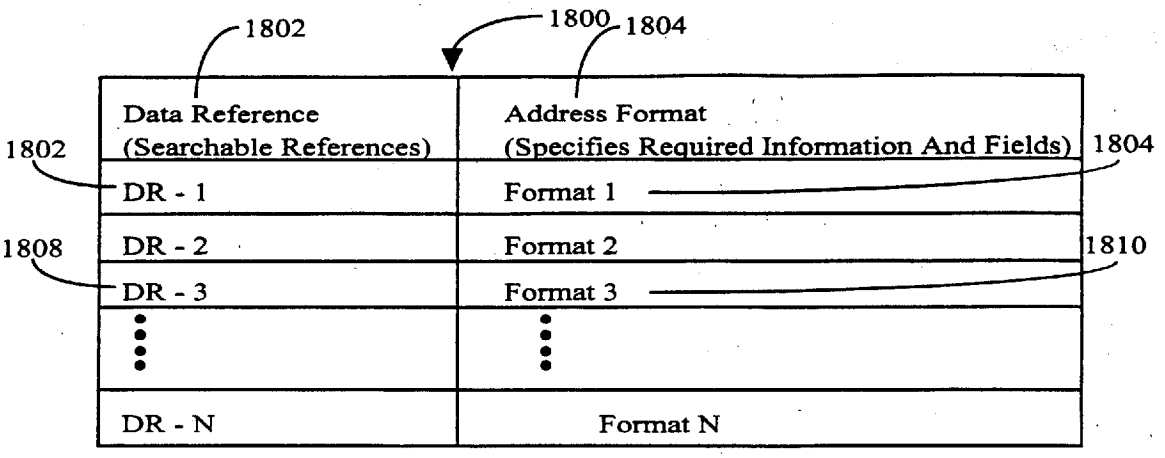


Figure 21

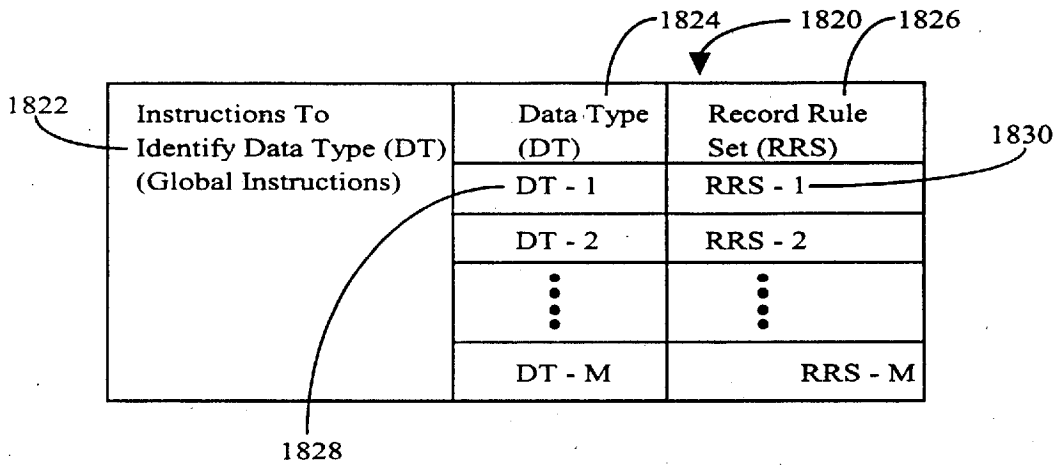


Figure 22

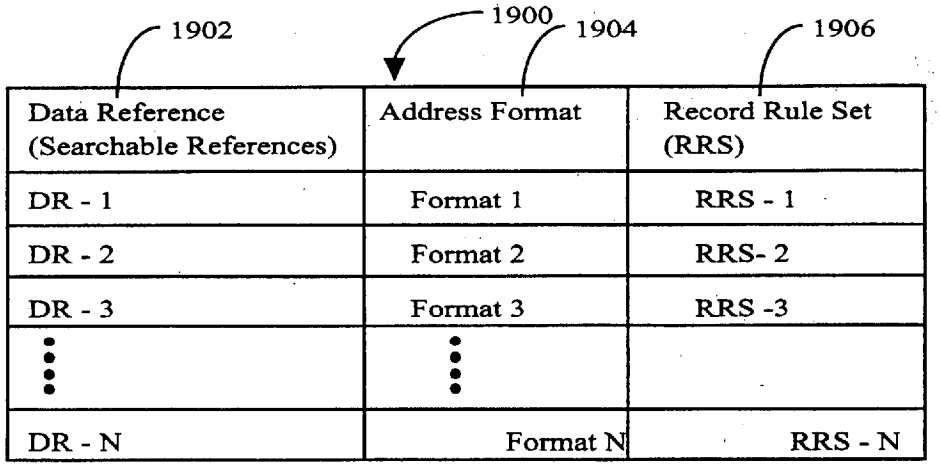


Figure 23

08/27/00 TUE 08:38 FAX 7033058 P.T.O 001

PART B—ISSUE FEE TRANSMITTAL

Complete and mail this form, together with applicable fees, to: Box ISSUE FEE
Assistant Commissioner for Patents
Washington, D.C. 20231

MAILING INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE. Blocks 1 through 4 should be completed where appropriate. All further correspondence including the Issue Fee receipt, the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Legibly mark-up with any corrections or use Block 1)

QUARLES & BRADY LLP
411 E. WISCONSIN AVENUE
SUITE 2040
MILWAUKEE , WI 53202-4497

Note: The certificate of mailing below can only be used for domestic mailings of the Issue Fee Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing.

Certificate of Mailing

I hereby certify that this Issue Fee Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Box Issue Fee address above on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	TOTAL CLAIMS	EXAMINER AND GROUP ART UNIT		DATE MAILED
09/512125 /	02/24/2000	064	Coby, F	2171	04/09/2001

First Named Applicant: DE LA HUERGA

TITLE OF INVENTION: **METHOD AND SYSTEM FOR AUTOMATED DATA STORAGE AND RETRIEVAL**

ATTY'S DOCKET NO.	CLASS-SUBCLASS	BATCH NO.	APPLN. TYPE	SMALL ENTITY	FEE DUE	DATE DUE
250591.90163	707/003.000	E69	Utility	NO	$1240.00	07/09/2001

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363). Use of PTO form(s) and Customer Number are recommended, but not required.

☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47) attached.

2. For printing on the patent front page, list (1) the names of up to 3 registered patent attorneys or agents OR, alternatively, (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)
PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. Inclusion of assignee data is only appropriate when an assignment has been previously submitted to the PTO or is being submitted under separate cover. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY & STATE OR COUNTRY)

Please check the appropriate assignee category indicated below (will not be printed on the patent)

☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fees are enclosed (make check payable to Commissioner of Patents and Trademarks):

☐ Issue Fee

☐ Advance Order - # of Copies _____

4b. The following fees or deficiency in these fees should be charged to:

DEPOSIT ACCOUNT NUMBER _____
(ENCLOSE AN EXTRA COPY OF THIS FORM)

☐ Issue Fee

☐ Advance Order - # of Copies _____

The COMMISSIONER OF PATENTS AND TRADEMARKS IS requested to apply the Issue Fee to the application identified above.

(Authorized Signature) (Date)

NOTE: The Issue Fee will be collected from the applicant or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

06/04/2001	00000103	1	142	$1,240.00	06/01/2001	DA 170055
06/04/2001	00000104	1	561	$30.00	06/01/2001	DA 170055

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending on the needs of the individual case. Any comments on the amount of time required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND FEES AND THIS FORM TO: Box Issue Fee, Assistant Commissioner for Patents, Washington D.C. 20231

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

EXHIBIT G

US006128635A

United States Patent [19]

Ikeno

[11] Patent Number: 6,128,635

[45] Date of Patent: Oct. 3, 2000

[54] **DOCUMENT DISPLAY SYSTEM AND ELECTRONIC DICTIONARY**

[75] Inventor: **Atsushi Ikeno**, Tokyo, Japan

[73] Assignee: **Oki Electric Industry Co., Ltd.**, Tokyo, Japan

[21] Appl. No.: **08/855,593**

[22] Filed: **May 13, 1997**

[30] **Foreign Application Priority Data**

May 13, 1996	[JP]	Japan	8-117415
May 13, 1996	[JP]	Japan	8-117661
May 14, 1996	[JP]	Japan	8-118766
May 14, 1996	[JP]	Japan	8-118795

[51] **Int. Cl.**[7] **G06F 17/24**

[52] **U.S. Cl.** **707/532**; 707/501; 707/513

[58] **Field of Search** 707/532, 501, 707/513

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,359,514	10/1994	Manthuruthil et al.	364/419.11
5,475,586	12/1995	Sata	364/419.02
5,822,720	10/1998	Bookman et al.	704/3
5,930,787	7/1999	Minakuchi et al.	707/4
5,940,614	8/1999	Allen et al.	395/701
5,963,205	10/1999	Sotomayor	345/333
6,009,382	12/1999	Martino et al.	704/1
6,011,554	1/2000	King et al.	345/352

FOREIGN PATENT DOCUMENTS

0483576	5/1992	European Pat. Off. .
28240	2/1993	Japan .
325081	11/1994	Japan .

OTHER PUBLICATIONS

Mejis, Inferring grammar from lexis, Applications and Alternatives, IEE Colloquium, pp. 3/1–3/5, 1993.

WordPerfect 6.1 Screen Thesaurus Dumps, Nov. 4, 1994.

WordNet: a lexical database for English, Miller, G.A, Communications of the ACM, abstract, Nov. 1995.

A Hypertext Based Software–Engineering Environment, Cybulsky et al, IEEE software, p. 62–68, Mar. 1992.

"Implicit Reader Links in Online Information", IBM Technical Disclosure Bulletin, US, IBM Corp. New York, vol. 37, No. 3, Mar. 1, 1994 , pp. 511–512.

"Dictionary Access Method with Word Occurrence", IBM Technical Disclosure Bulletin, US, IBM Corp. New York, vol. 38, No. 8, Aug. 1995, pp. 365–366.

"Automatic Hypermedia Link Generation", IBM Technical Disclosure Bulletin, US, IBM Corp. New York, vol. 35, No. 1A, Jun. 1, 1992, pp. 447–448.

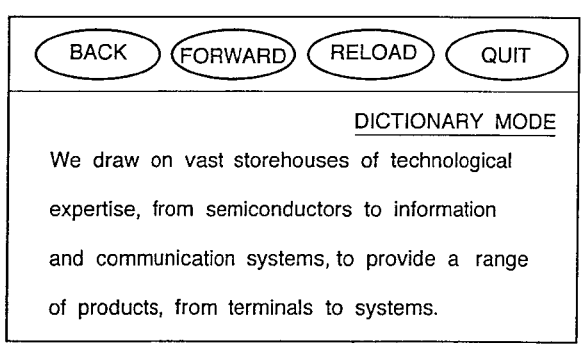
"The Art of Navigating", Communications of the Association for Computing Machinery, US, Jakob Nielsen Association for Computing Machinery New York, vol. 33, No. 3, Mar. 1, 1990, pp. 296–310.

Primary Examiner—Stephen S. Hong
Assistant Examiner—Cesar B. Paula
Attorney, Agent, or Firm—Wenderoth, Lind & Ponack, L.L.P.

[57] **ABSTRACT**

A document display system displays the contents of an electronic document. If the user requests dictionary access, the electronic document is modified by attaching invisible dictionary access information to words in the document. The modified document is then displayed, the words to which the dictionary access information is attached being visibly marked. If the user selects a word to which dictionary access information is attached, an electronic dictionary entry for the word is automatically retrieved and displayed.

42 Claims, 44 Drawing Sheets



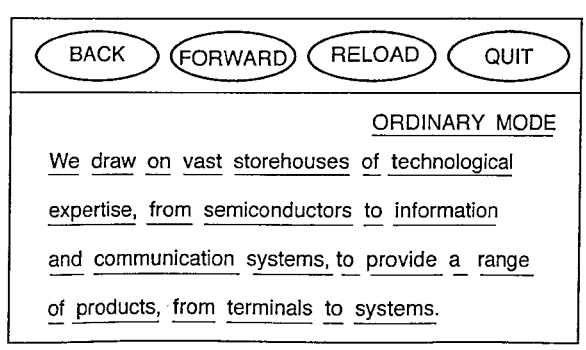


FIG.1

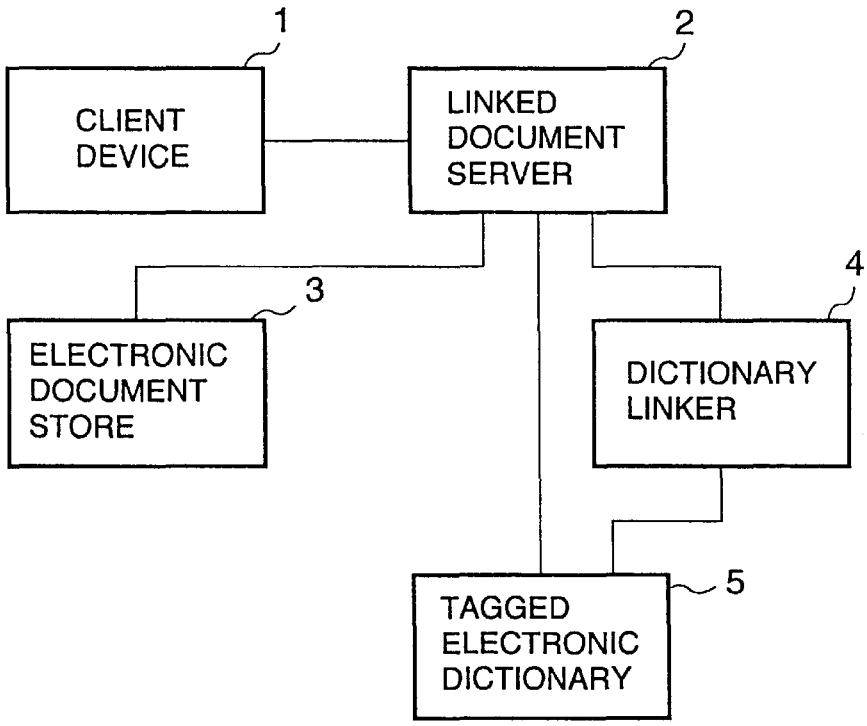

1
2
CLIENT DEVICE
LINKED DOCUMENT SERVER
3
4
ELECTRONIC DOCUMENT STORE
DICTIONARY LINKER
5
TAGGED ELECTRONIC DICTIONARY

FIG.2

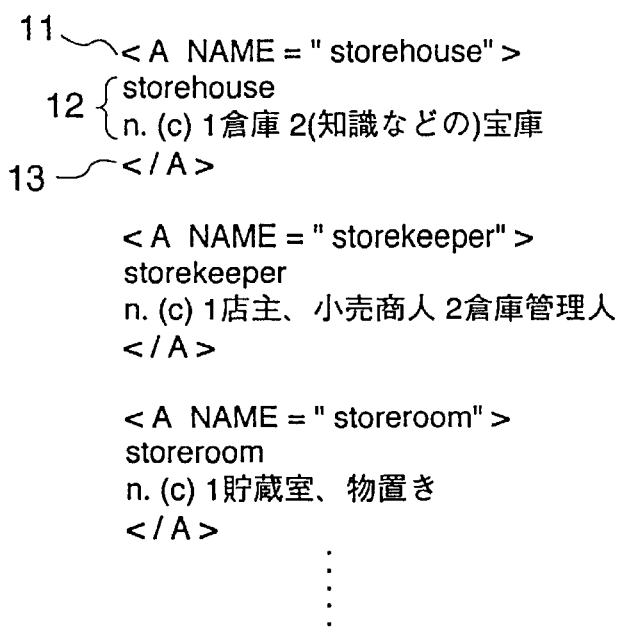

11
< A NAME = " storehouse" >
12
storehouse
n. (c) 1倉庫 2(知識などの)宝庫
13
< / A >
< A NAME = " storekeeper" >
storekeeper
n. (c) 1店主、小売商人 2倉庫管理人
< / A >
< A NAME = " storeroom" >
storeroom
n. (c) 1貯蔵室、物置き
< / A >

FIG.3

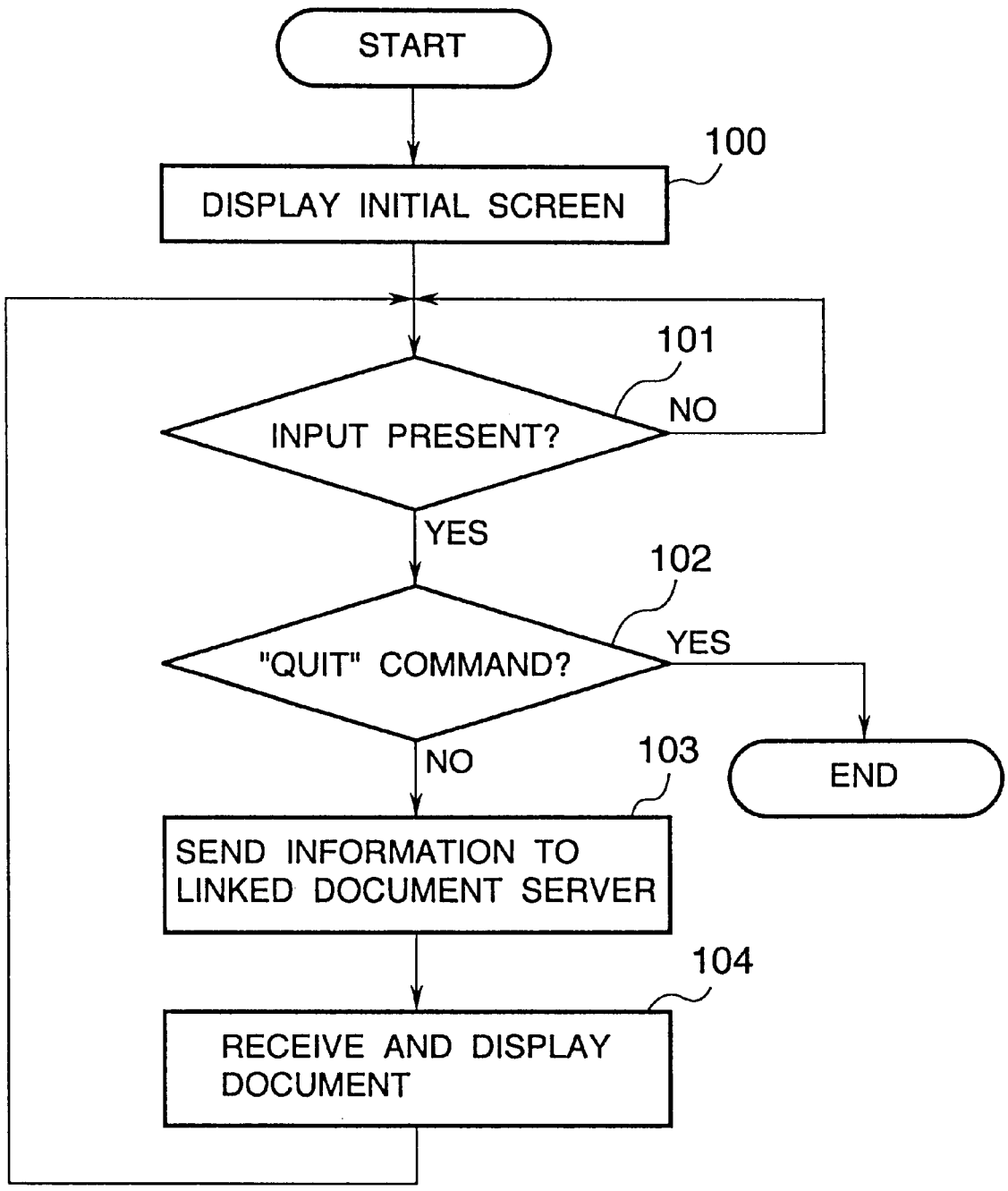

START
100
DISPLAY INITIAL SCREEN
101
INPUT PRESENT?
NO
YES
102
"QUIT" COMMAND?
YES
END
NO
103
SEND INFORMATION TO LINKED DOCUMENT SERVER
104
RECEIVE AND DISPLAY DOCUMENT

FIG.4

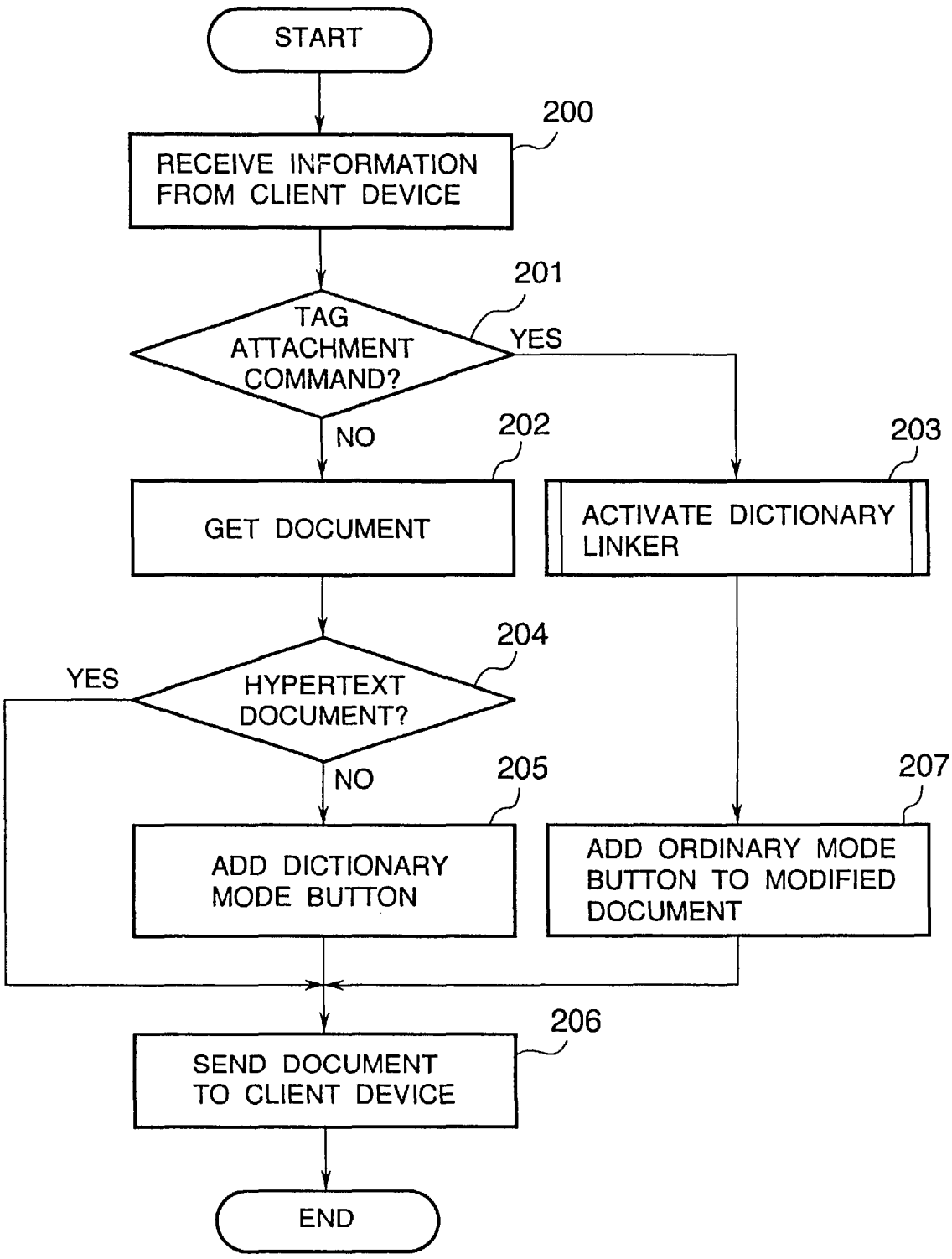

START
200
RECEIVE INFORMATION FROM CLIENT DEVICE
201
TAG ATTACHMENT COMMAND?
YES
NO
202
GET DOCUMENT
203
ACTIVATE DICTIONARY LINKER
204
YES
HYPERTEXT DOCUMENT?
NO
205
ADD DICTIONARY MODE BUTTON
207
ADD ORDINARY MODE BUTTON TO MODIFIED DOCUMENT
206
SEND DOCUMENT TO CLIENT DEVICE
END

FIG.5

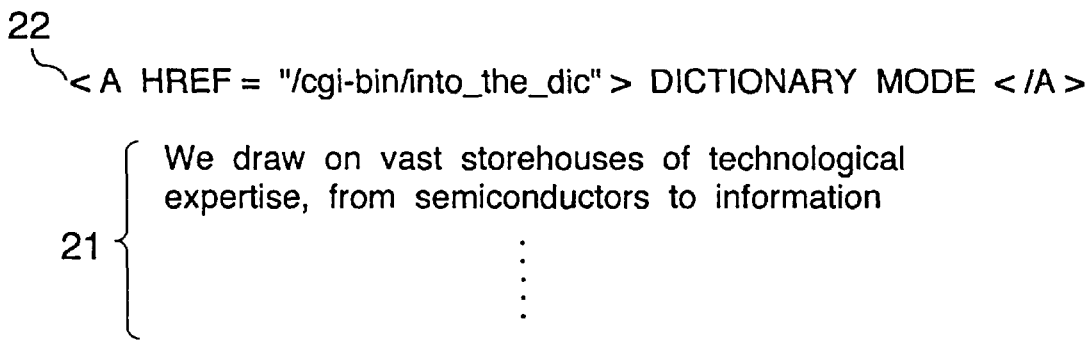

22
< A HREF = "/cgi-bin/into_the_dic" > DICTIONARY MODE < /A >
21
We draw on vast storehouses of technological expertise, from semiconductors to information

FIG.6

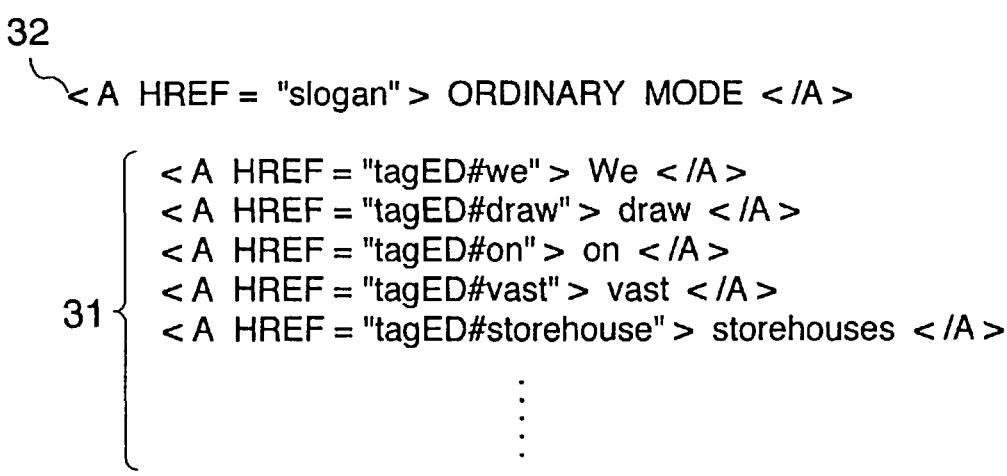

32
< A HREF = "slogan" > ORDINARY MODE < /A >
31
< A HREF = "tagED#we" > We < /A >
< A HREF = "tagED#draw" > draw < /A >
< A HREF = "tagED#on" > on < /A >
< A HREF = "tagED#vast" > vast < /A >
< A HREF = "tagED#storehouse" > storehouses < /A >

FIG.7

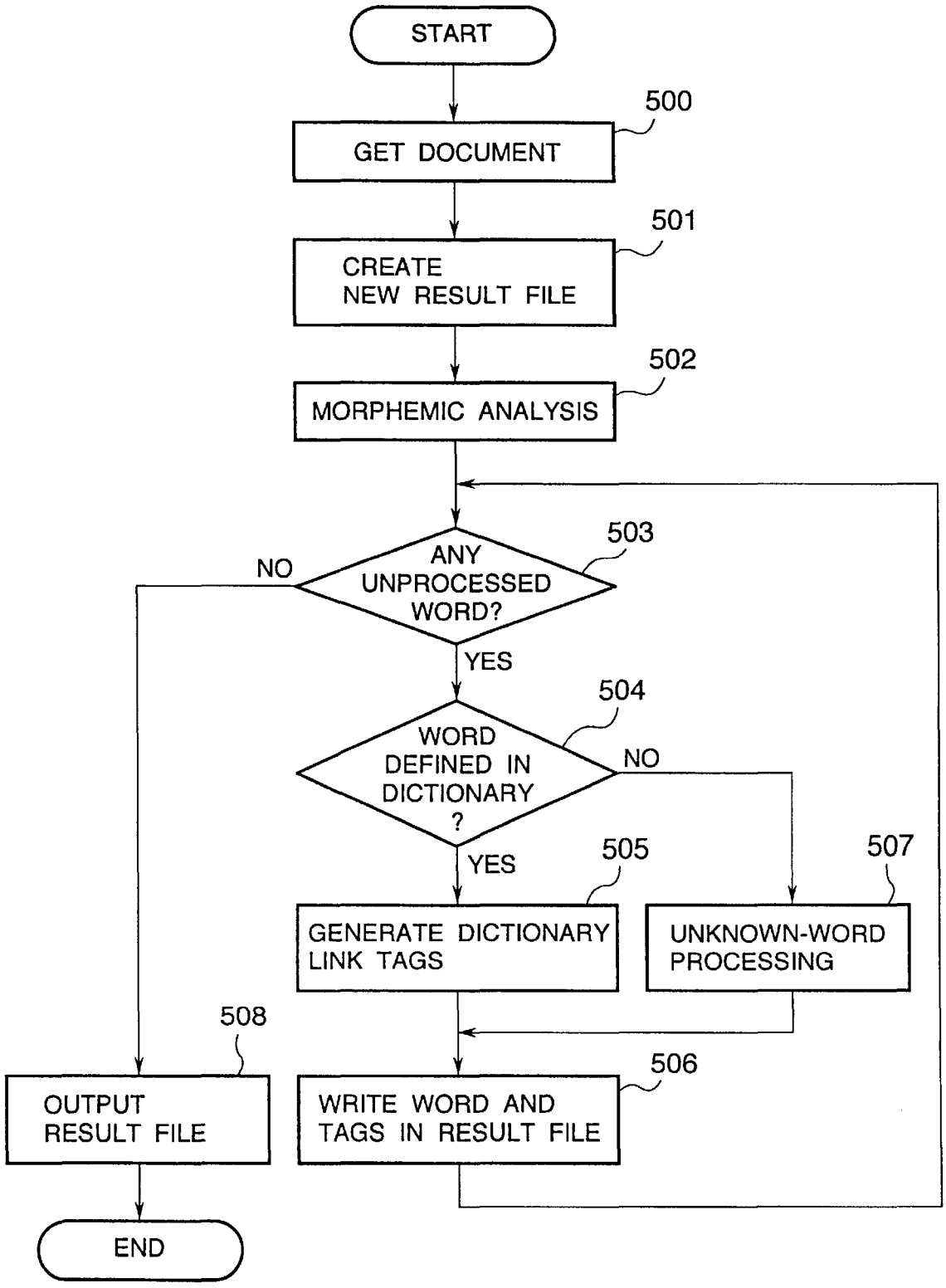

START
500
GET DOCUMENT
501
CREATE NEW RESULT FILE
502
MORPHEMIC ANALYSIS
503
ANY UNPROCESSED WORD?
NO
YES
504
WORD DEFINED IN DICTIONARY ?
NO
YES
505
GENERATE DICTIONARY LINK TAGS
507
UNKNOWN-WORD PROCESSING
506
WRITE WORD AND TAGS IN RESULT FILE
508
OUTPUT RESULT FILE
END

FIG.8

```
< A HREF = "tagED#we" > We < /A >
< A HREF = "tagED#draw" > draw < /A >
< A HREF = "tagED#on" > on < /A >
< A HREF = "tagED#vast" > vast < /A >
< A HREF = "tagED#storehouse" > storehouses < /A >
```

FIG.9

```
< A HREF = "tagED#we" > We < /A >
< A HREF = "tagED#draw" > draw < /A >
< A HREF = "tagED#on" > on < /A >
 vast
< A HREF = "tagED#storehouse" > storehouses < /A >
```

FIG.10

```
< A NAME = "UNKNOWN_WORDS" >
その単語は辞書にありません。
< /A >
```

FIG.11

```
< A HREF = "tagED#we" > We < /A >
< A HREF = "tagED#draw" > draw < /A >
< A HREF = "tagED#on" > on < /A >
< A HREF = "tagED#UNKNOWN_WORDS" > vast < /A >
< A HREF = "tagED#storehouse" > storehouses < /A >
```

FIG.12

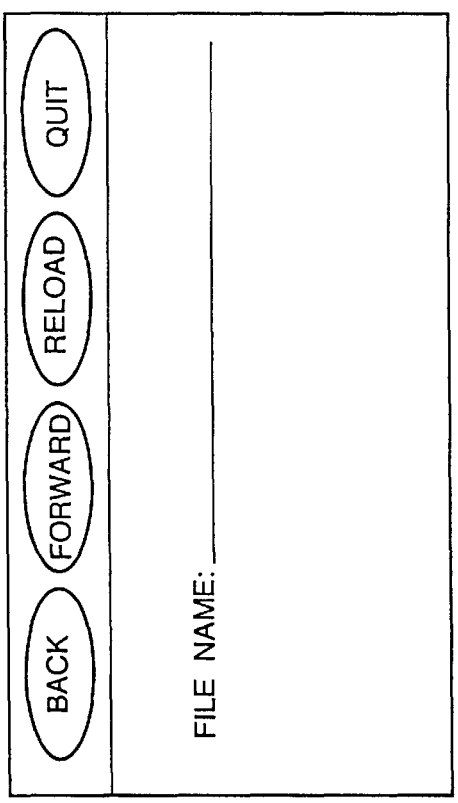


FIG.13

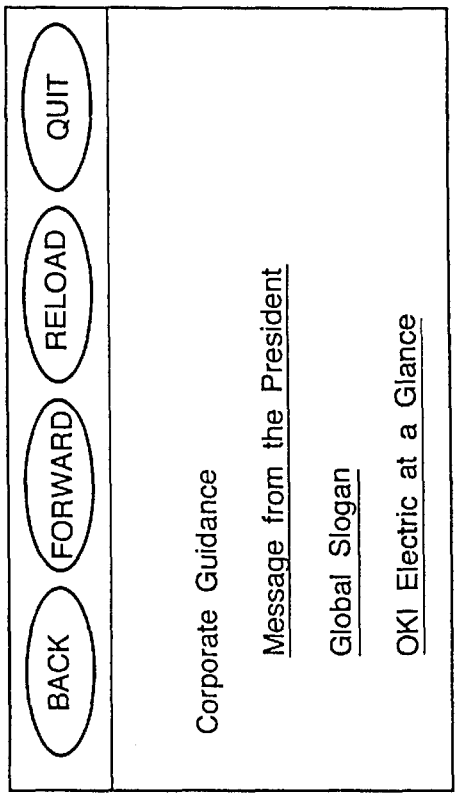


FIG.14

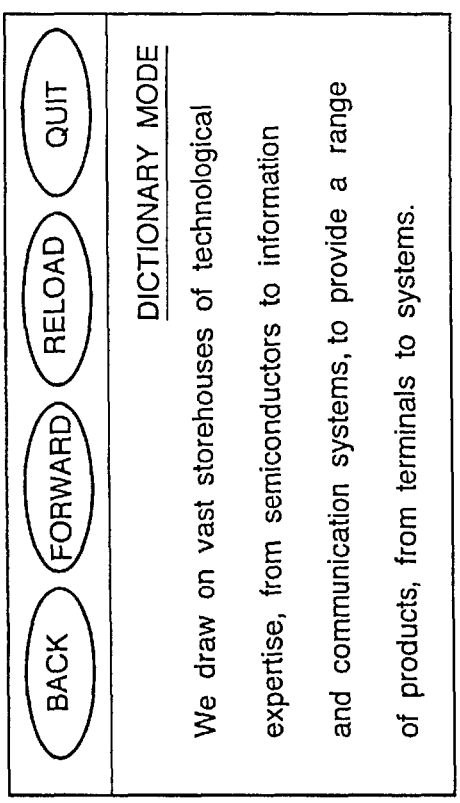


FIG.15

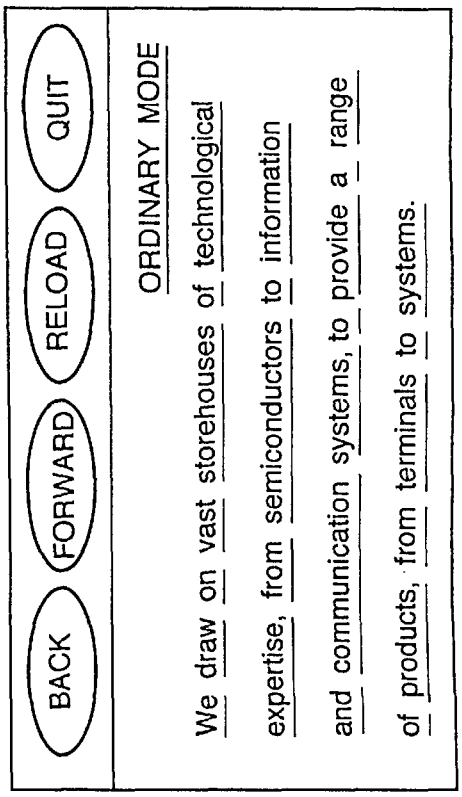


FIG.16

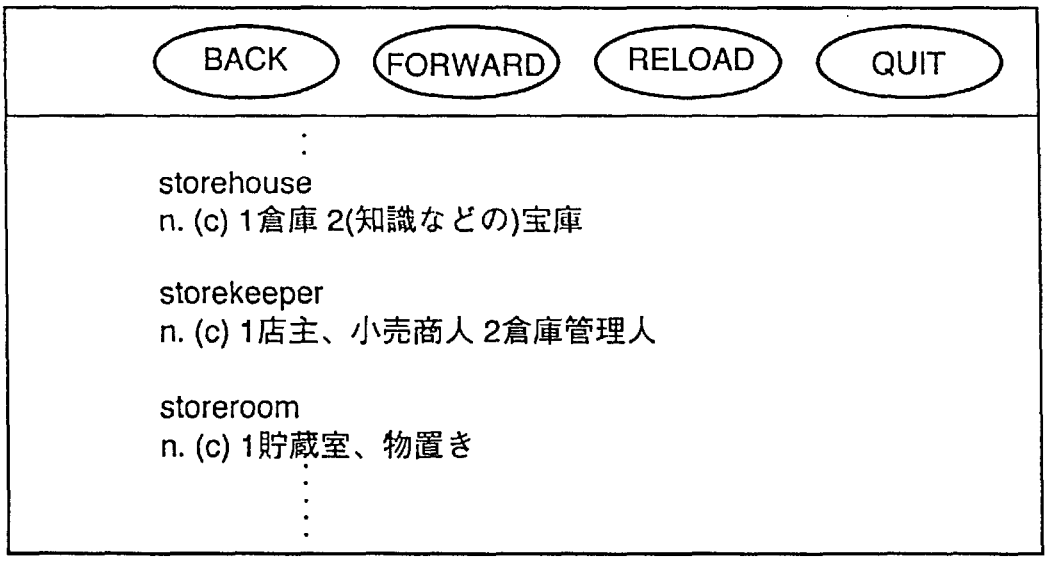

BACK
FORWARD
RELOAD
QUIT
storehouse
n. (c) 1倉庫 2(知識などの)宝庫
storekeeper
n. (c) 1店主、小売商人 2倉庫管理人
storeroom
n. (c) 1貯蔵室、物置き

FIG.17

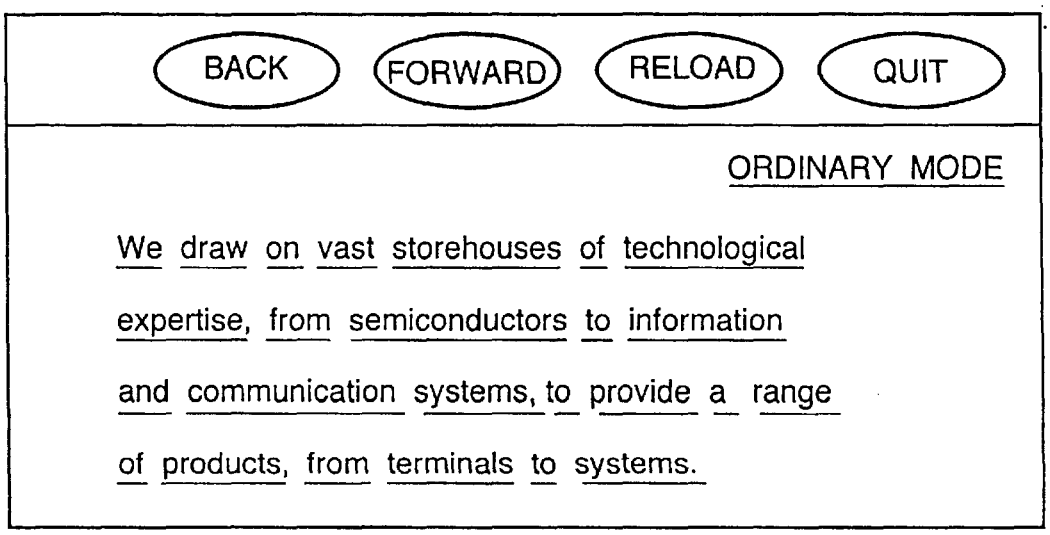

BACK
FORWARD
RELOAD
QUIT
ORDINARY MODE
We draw on vast storehouses of technological expertise, from semiconductors to information and communication systems, to provide a range of products, from terminals to systems.

FIG.18

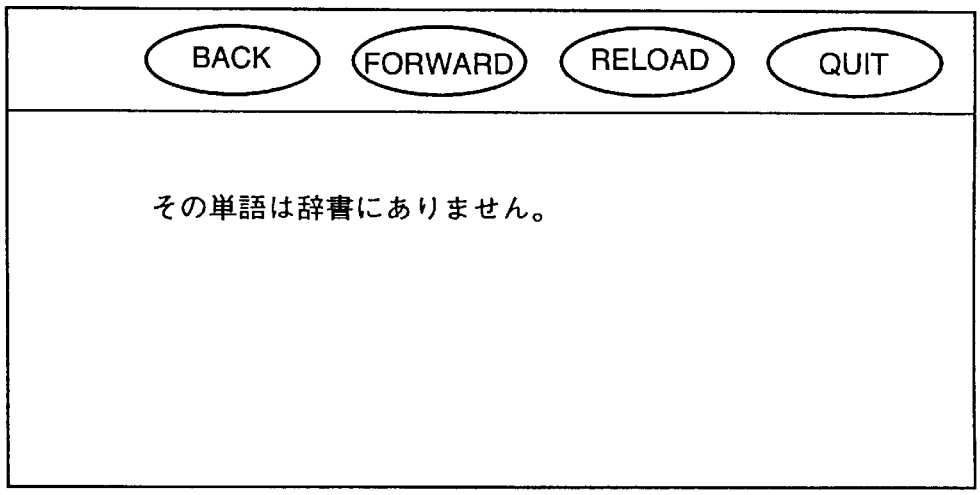

BACK
FORWARD
RELOAD
QUIT
その単語は辞書にありません。

FIG.19

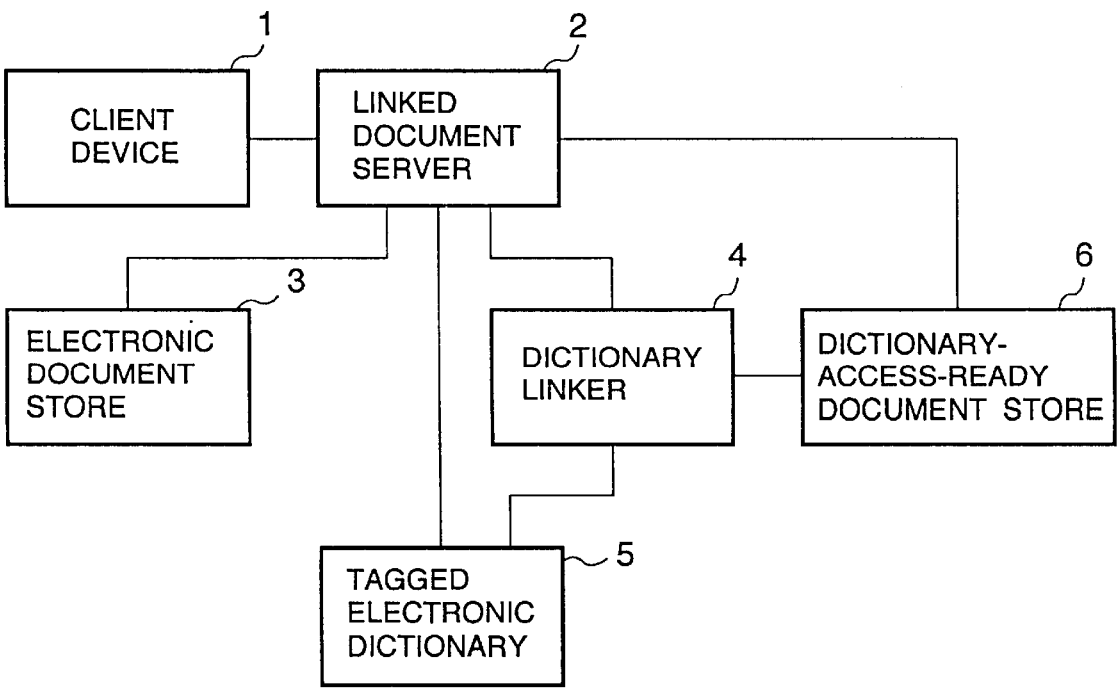

1
CLIENT DEVICE
2
LINKED DOCUMENT SERVER
3
ELECTRONIC DOCUMENT STORE
4
DICTIONARY LINKER
6
DICTIONARY-ACCESS-READY DOCUMENT STORE
5
TAGGED ELECTRONIC DICTIONARY

FIG.20

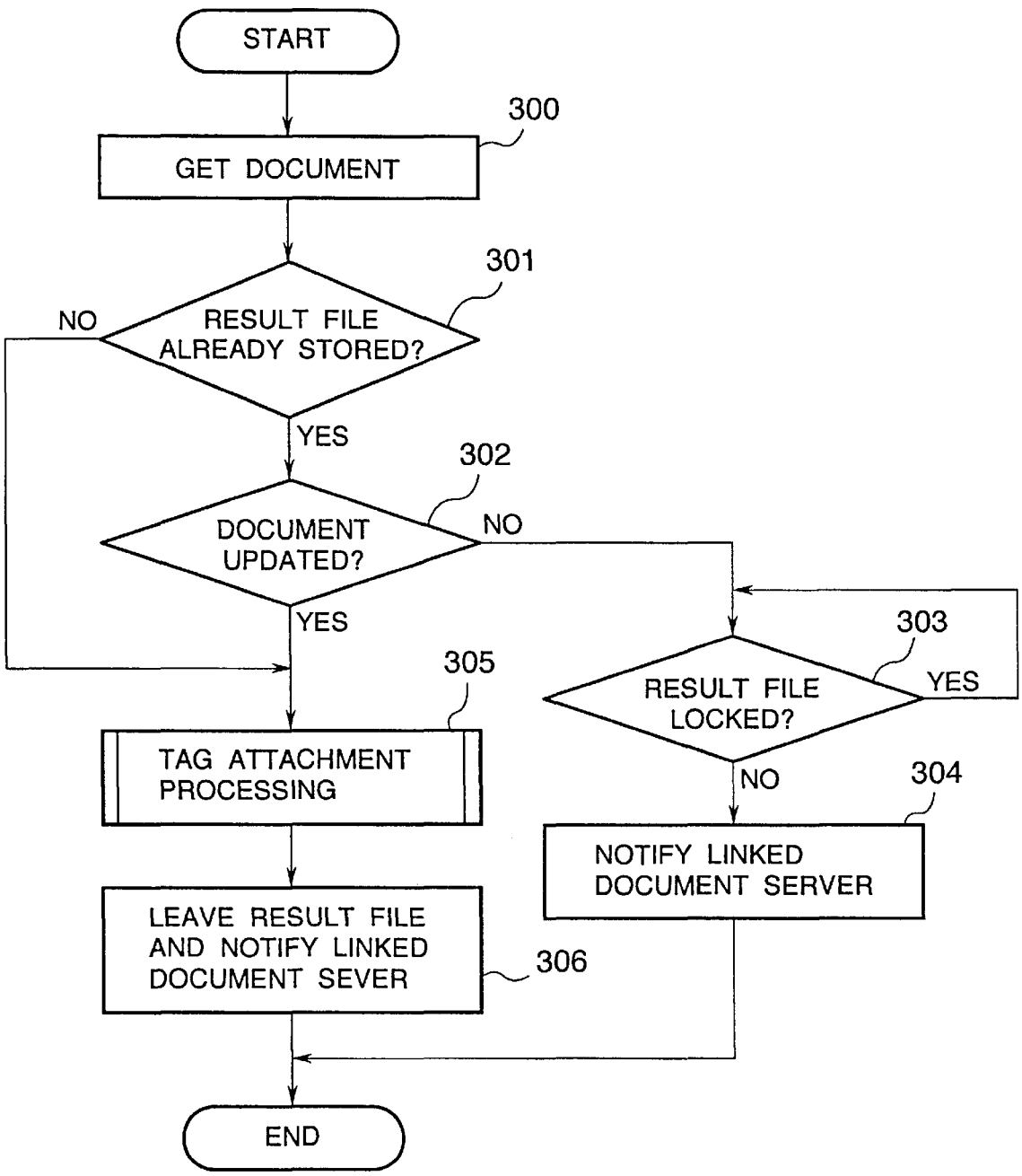

START
300
GET DOCUMENT
301
NO
RESULT FILE ALREADY STORED?
YES
302
DOCUMENT UPDATED?
NO
YES
303
RESULT FILE LOCKED?
YES
NO
304
NOTIFY LINKED DOCUMENT SERVER
305
TAG ATTACHMENT PROCESSING
LEAVE RESULT FILE AND NOTIFY LINKED DOCUMENT SEVER
306
END

FIG.21

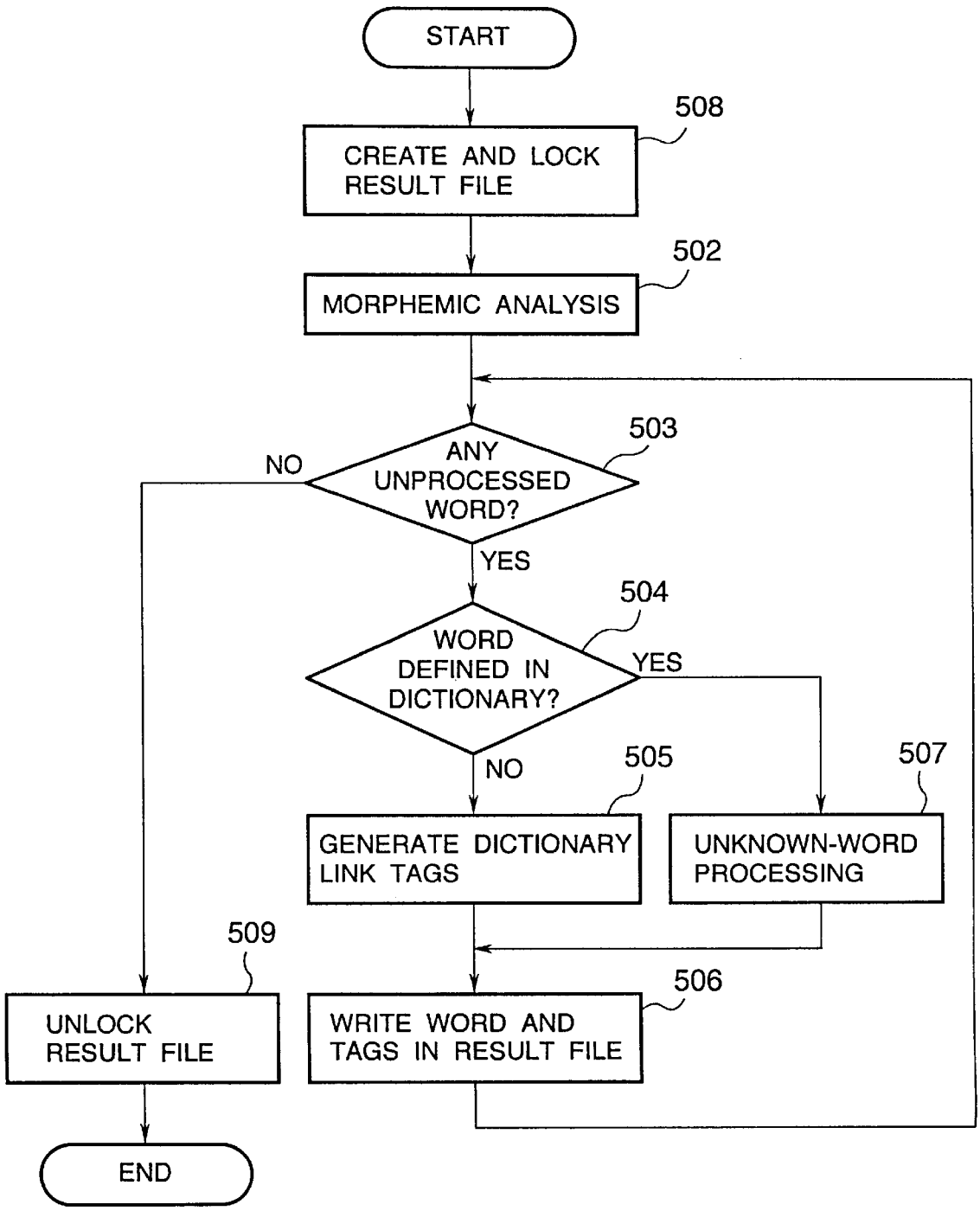

START
508
CREATE AND LOCK RESULT FILE
502
MORPHEMIC ANALYSIS
503
ANY UNPROCESSED WORD?
NO
YES
504
WORD DEFINED IN DICTIONARY?
YES
NO
505
GENERATE DICTIONARY LINK TAGS
507
UNKNOWN-WORD PROCESSING
509
UNLOCK RESULT FILE
506
WRITE WORD AND TAGS IN RESULT FILE
END

FIG.22

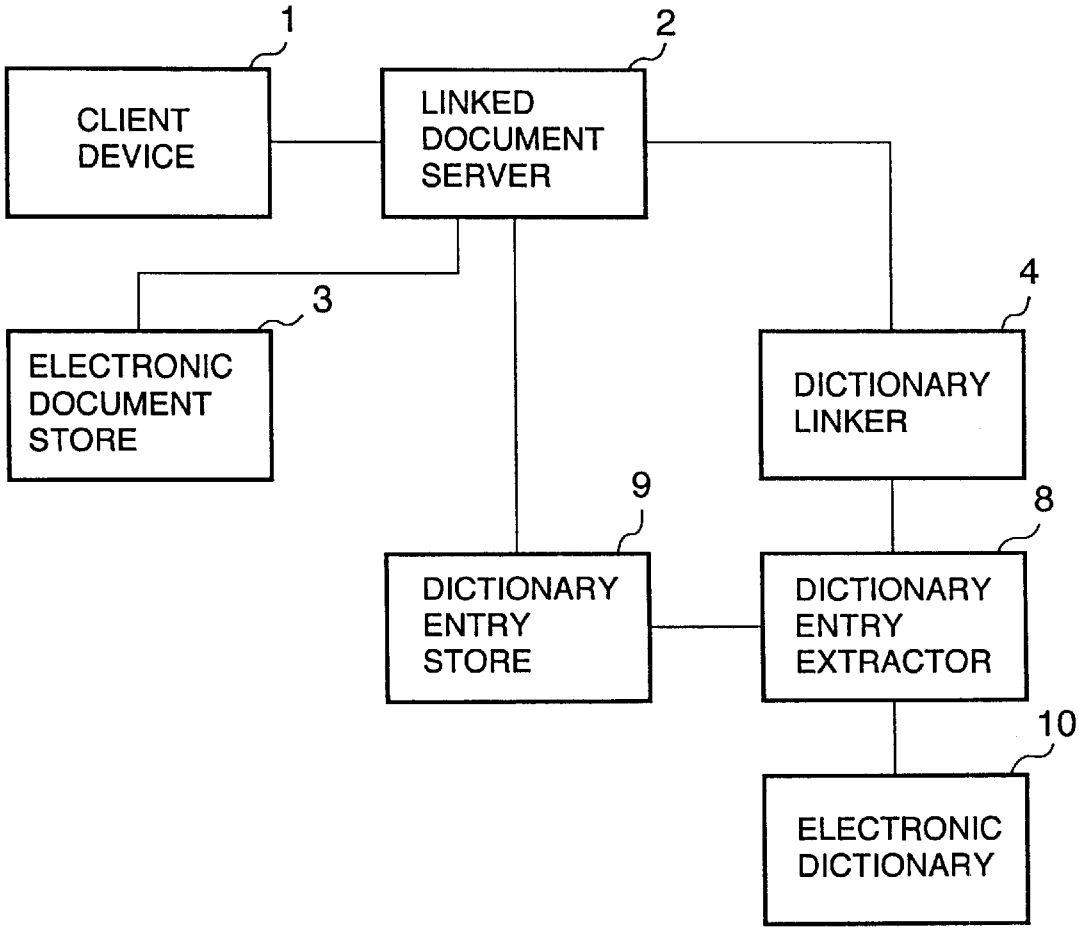


FIG.23

storehouse
n. (c) 1倉庫 2 (知識などの)宝庫

storekeeper
n. (c) 1店主、小売商人 2 倉庫管理人

storeroom
n. (c) 1貯蔵室、物置き

FIG.24

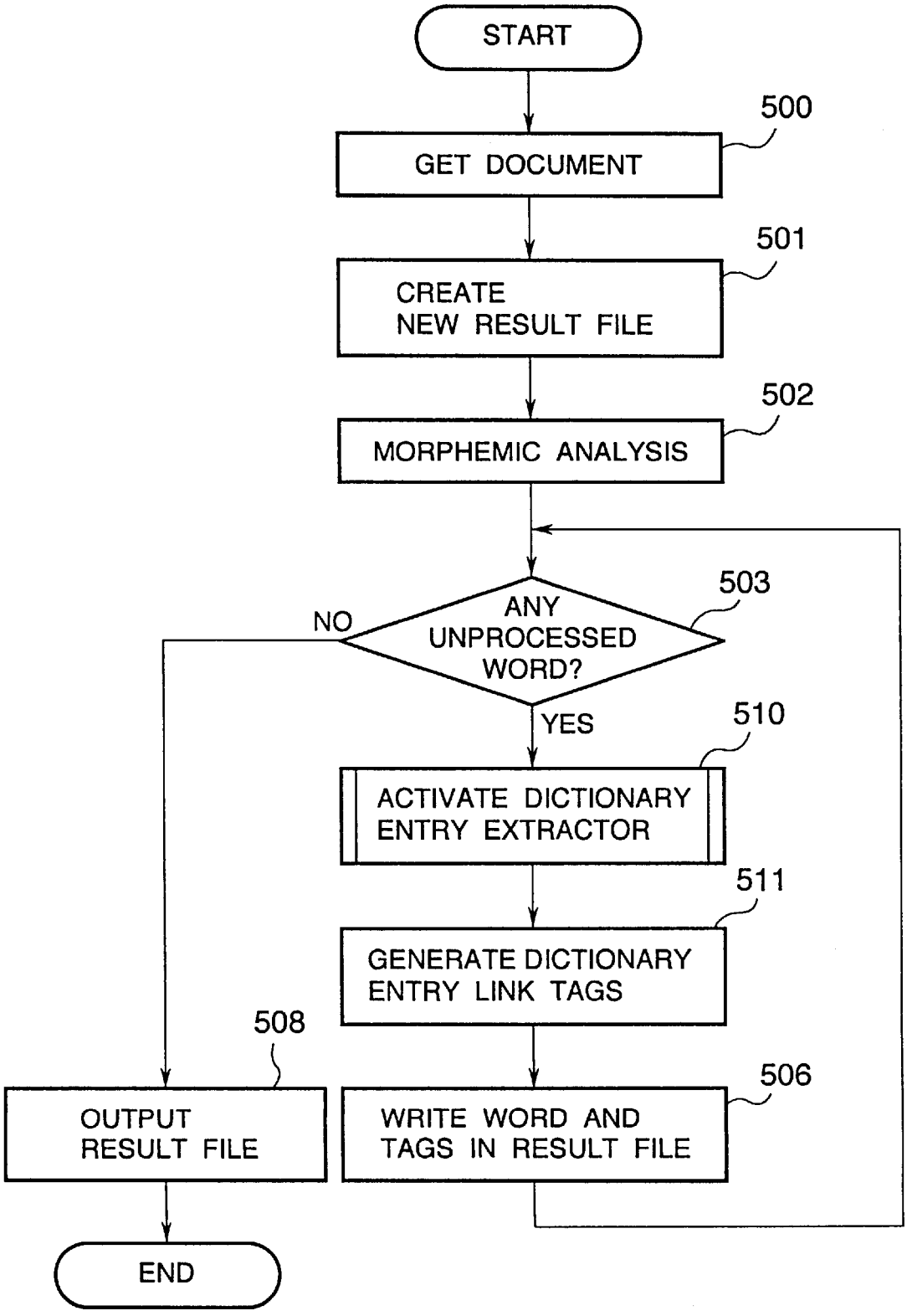

START
500
GET DOCUMENT
501
CREATE NEW RESULT FILE
502
MORPHEMIC ANALYSIS
503
ANY UNPROCESSED WORD?
NO
YES
510
ACTIVATE DICTIONARY ENTRY EXTRACTOR
511
GENERATE DICTIONARY ENTRY LINK TAGS
506
WRITE WORD AND TAGS IN RESULT FILE
508
OUTPUT RESULT FILE
END

FIG.25

```
< A HREF = "/dic/keep/we" > We < /A >
< A HREF = "/dic/keep/draw" > draw < /A >
< A HREF = "/dic/keep/on" > on < /A >
< A HREF = "/dic/keep/vast" > vast < /A >
< A HREF = "/dic/keep/storehouse" > storehouses < /A >
```

FIG.26

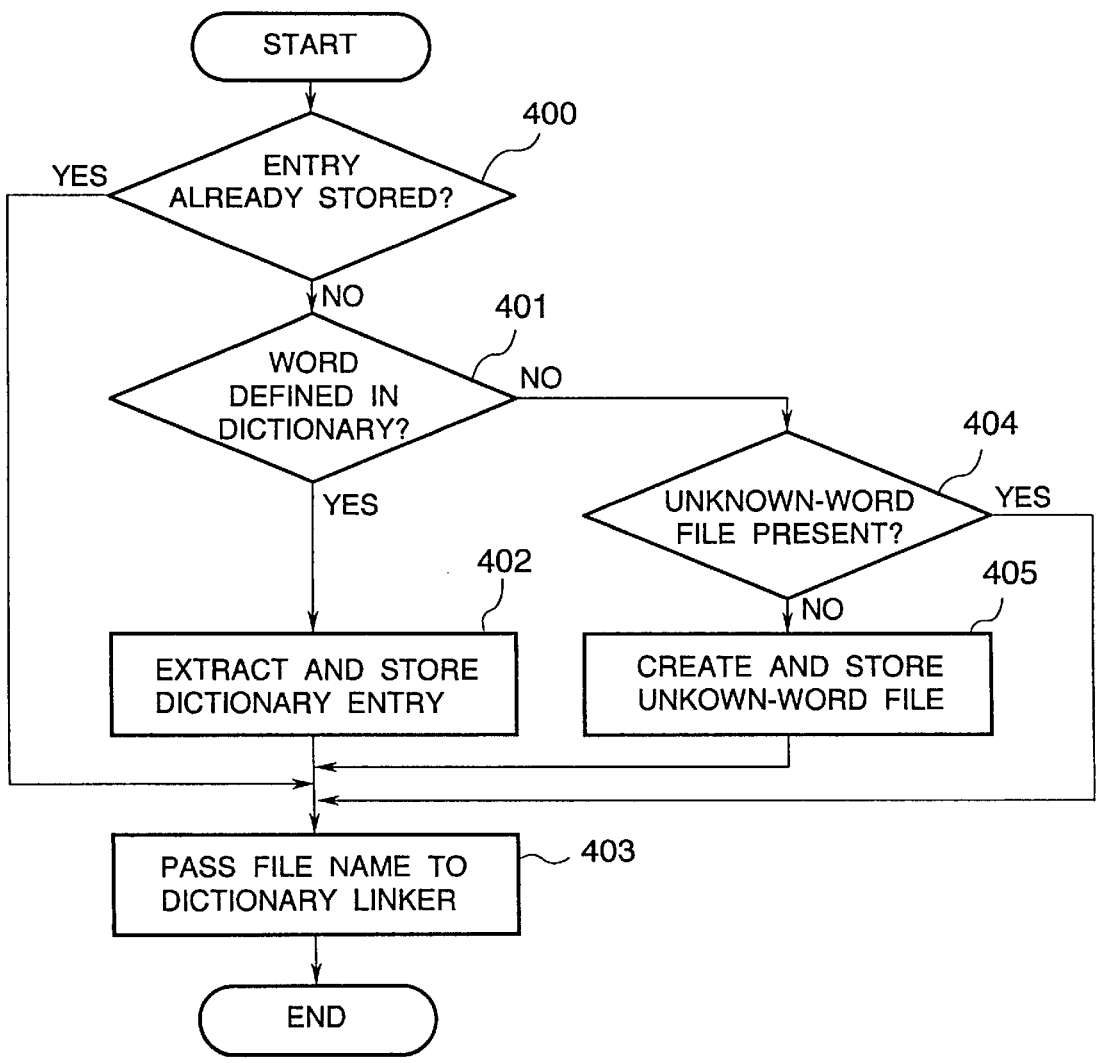


FIG.27

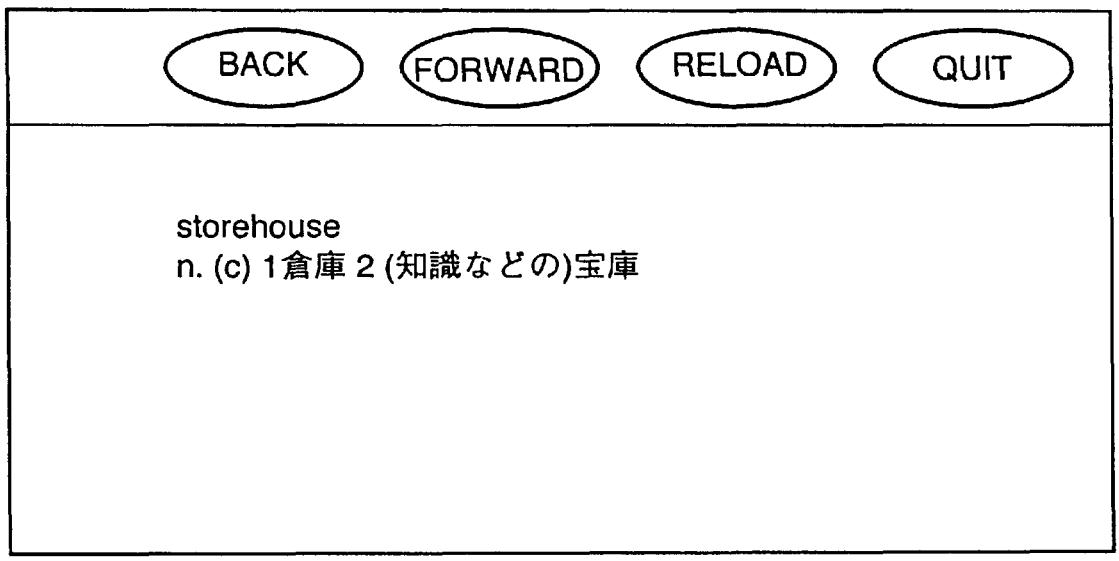


FIG.28

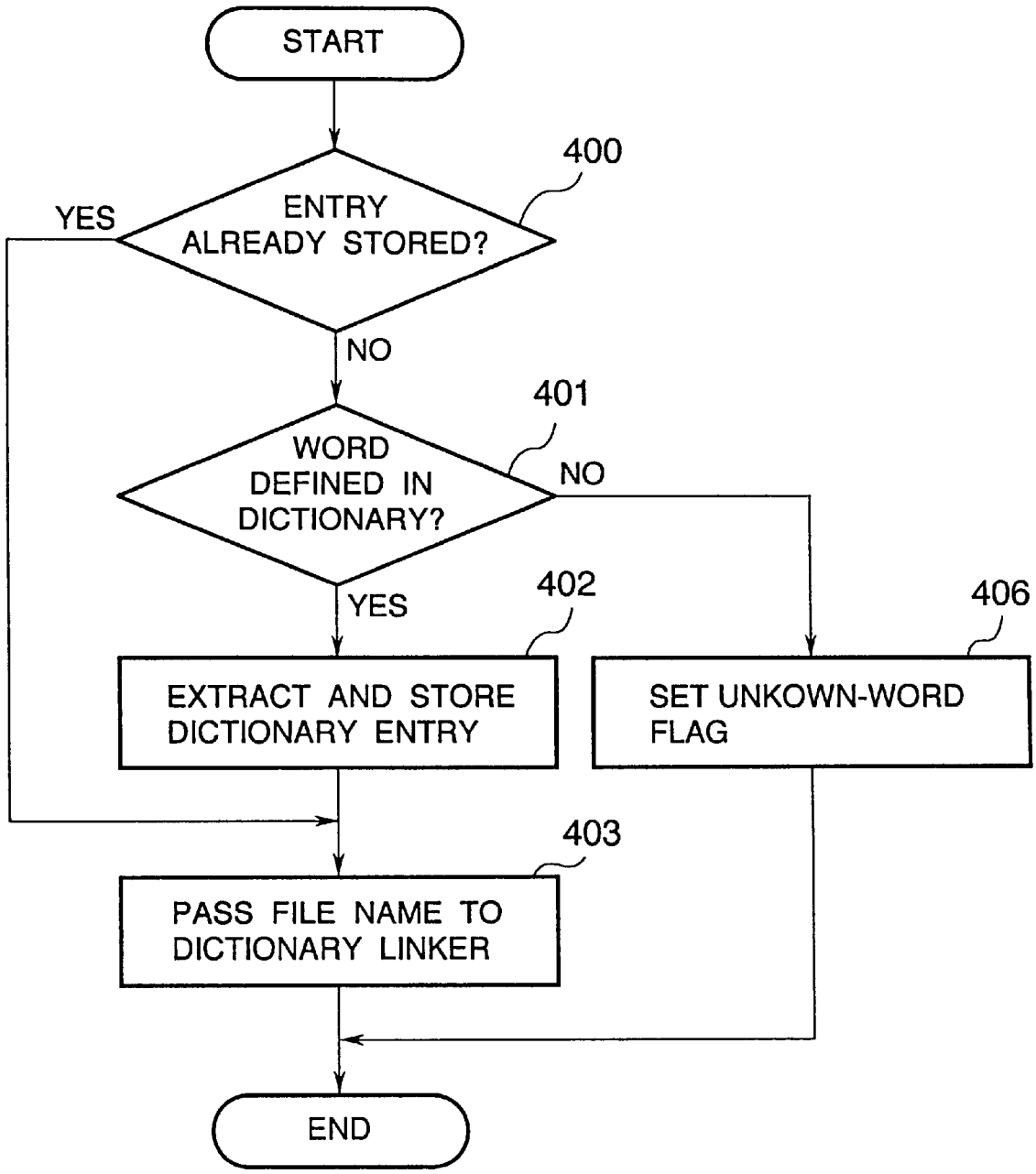

START
400
ENTRY ALREADY STORED?
YES
NO
401
WORD DEFINED IN DICTIONARY?
NO
YES
402
406
EXTRACT AND STORE DICTIONARY ENTRY
SET UNKOWN-WORD FLAG
403
PASS FILE NAME TO DICTIONARY LINKER
END

FIG.29

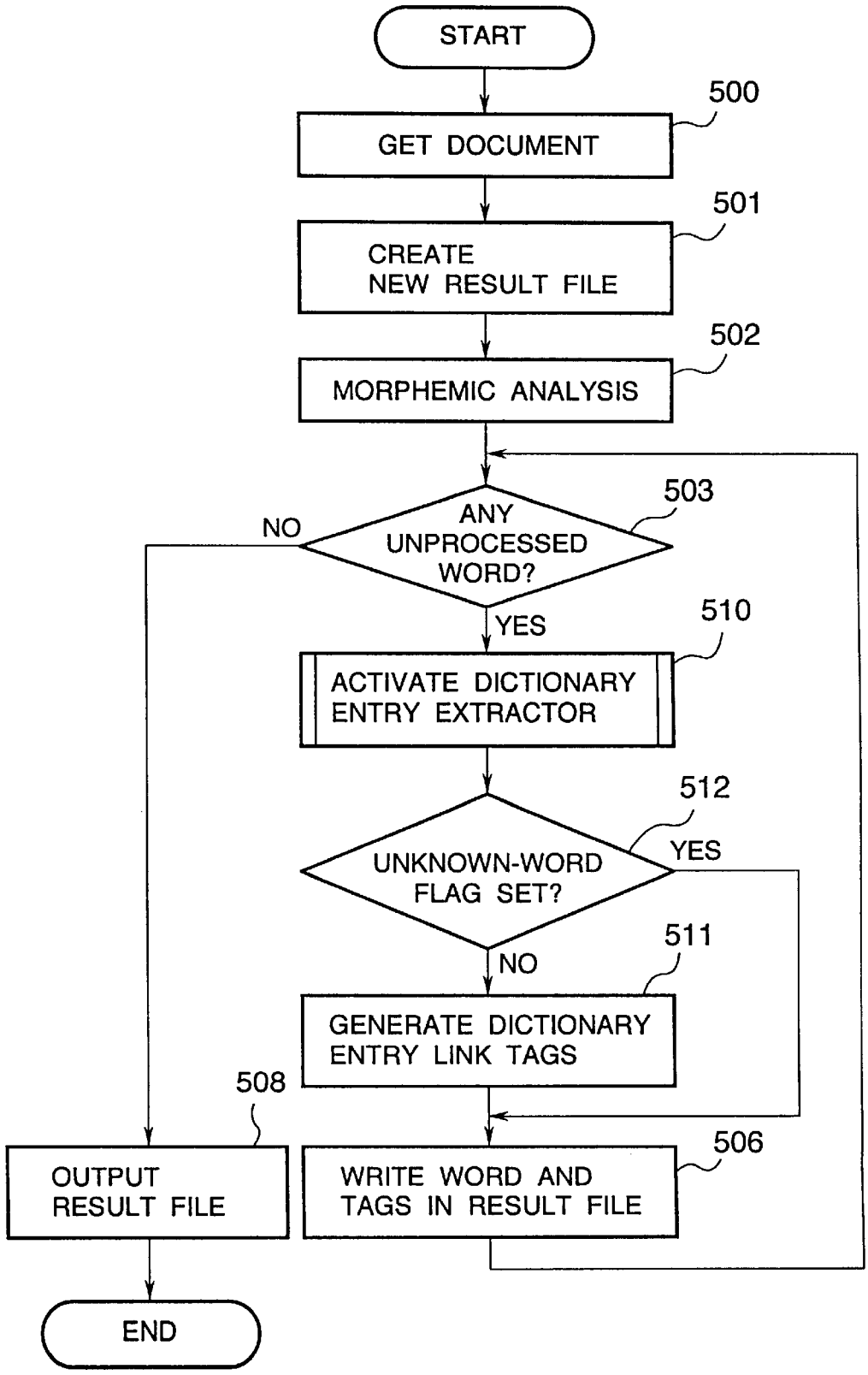

START
500
GET DOCUMENT
501
CREATE NEW RESULT FILE
502
MORPHEMIC ANALYSIS
503
ANY UNPROCESSED WORD?
NO
YES
510
ACTIVATE DICTIONARY ENTRY EXTRACTOR
512
UNKNOWN-WORD FLAG SET?
YES
NO
511
GENERATE DICTIONARY ENTRY LINK TAGS
508
OUTPUT RESULT FILE
506
WRITE WORD AND TAGS IN RESULT FILE
END

FIG.30

```
< A  HREF = "/dic/keep/we" > We < /A >
< A  HREF = "/dic/keep/draw" > draw < /A >
< A  HREF = "/dic/keep/on" > on < /A >
 vast
< A  HREF = "/dic/keep/storehouse" > storehouses < /A >
                    .
                    .
                    .
```

FIG.31

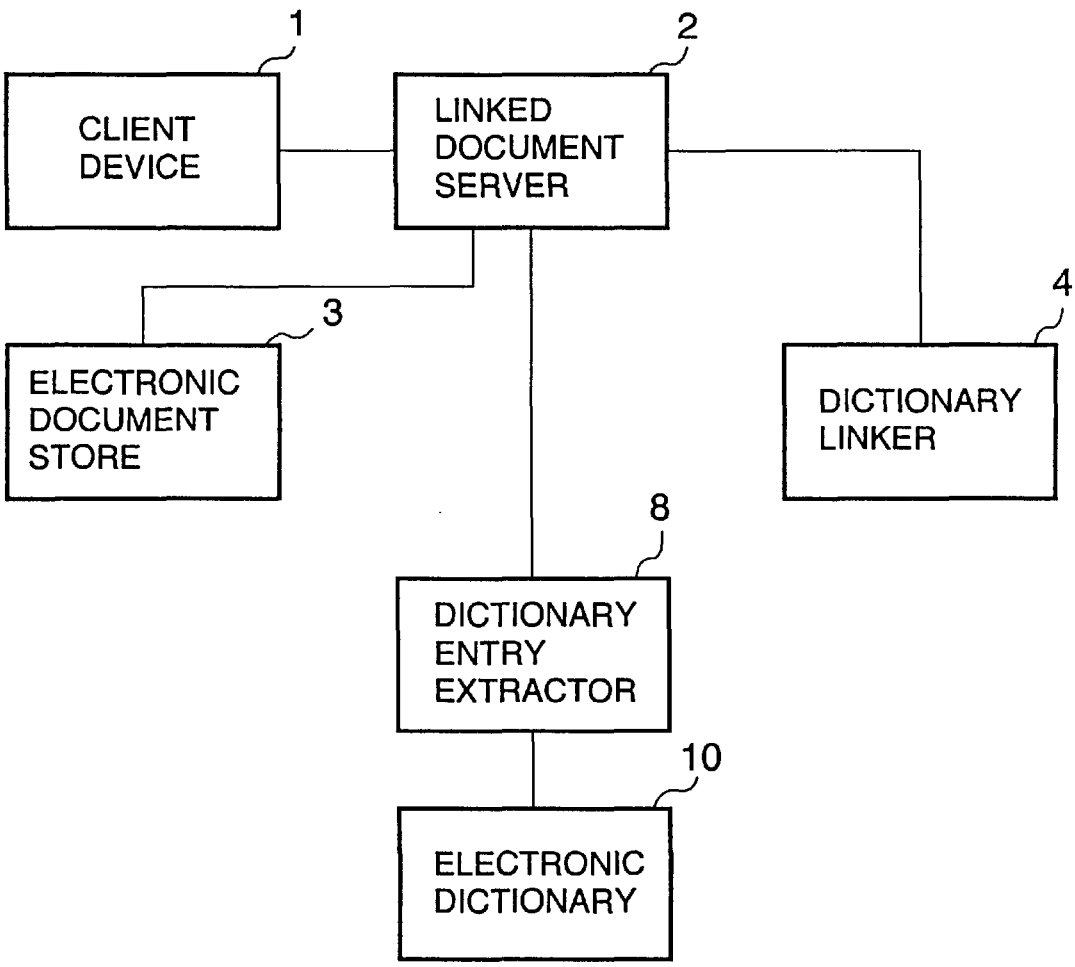


FIG.32

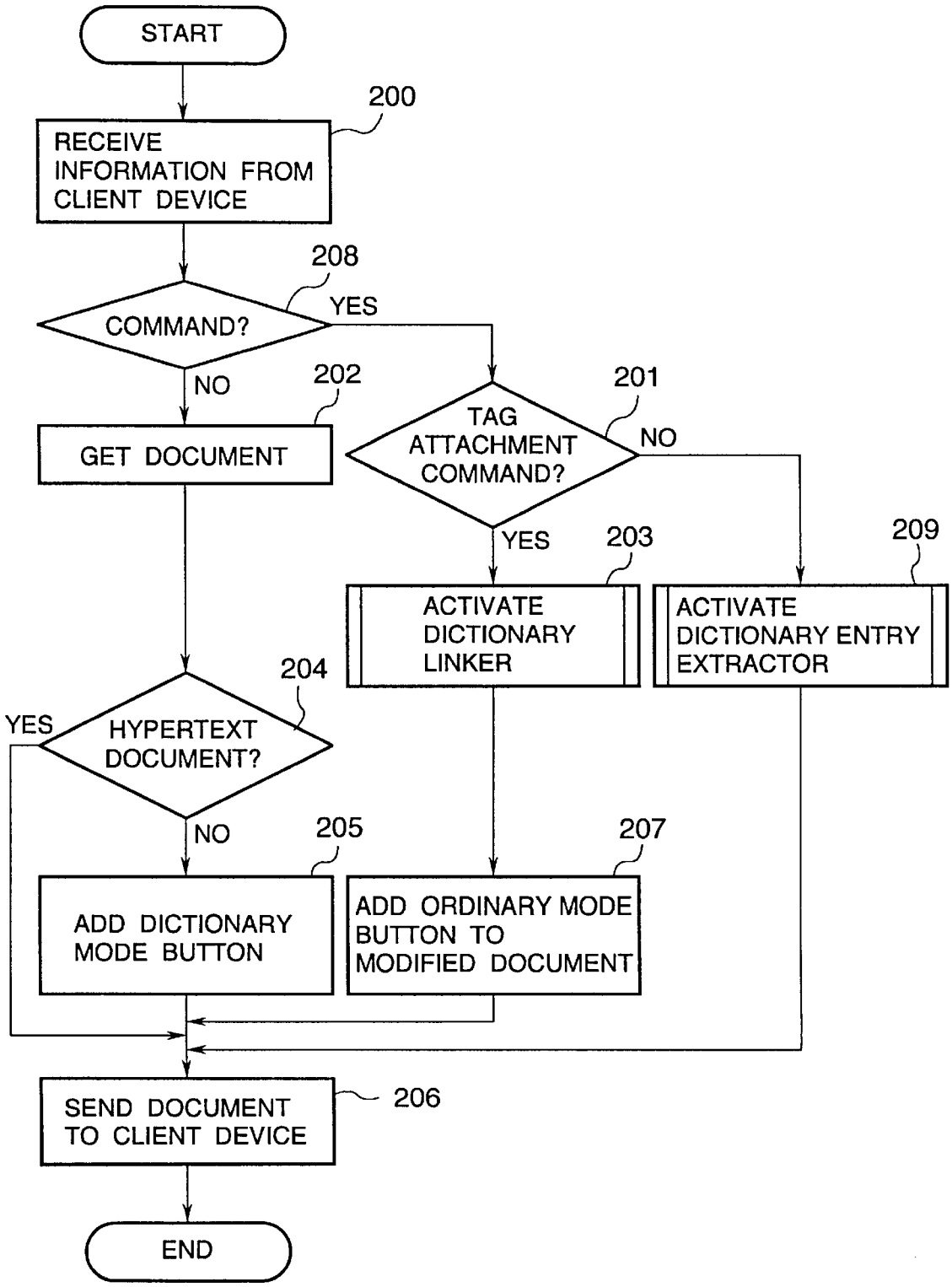

START
200
RECEIVE INFORMATION FROM CLIENT DEVICE
208
COMMAND?
YES
NO
202
GET DOCUMENT
201
TAG ATTACHMENT COMMAND?
NO
YES
203
ACTIVATE DICTIONARY LINKER
209
ACTIVATE DICTIONARY ENTRY EXTRACTOR
204
YES
HYPERTEXT DOCUMENT?
NO
205
ADD DICTIONARY MODE BUTTON
207
ADD ORDINARY MODE BUTTON TO MODIFIED DOCUMENT
206
SEND DOCUMENT TO CLIENT DEVICE
END

FIG.33

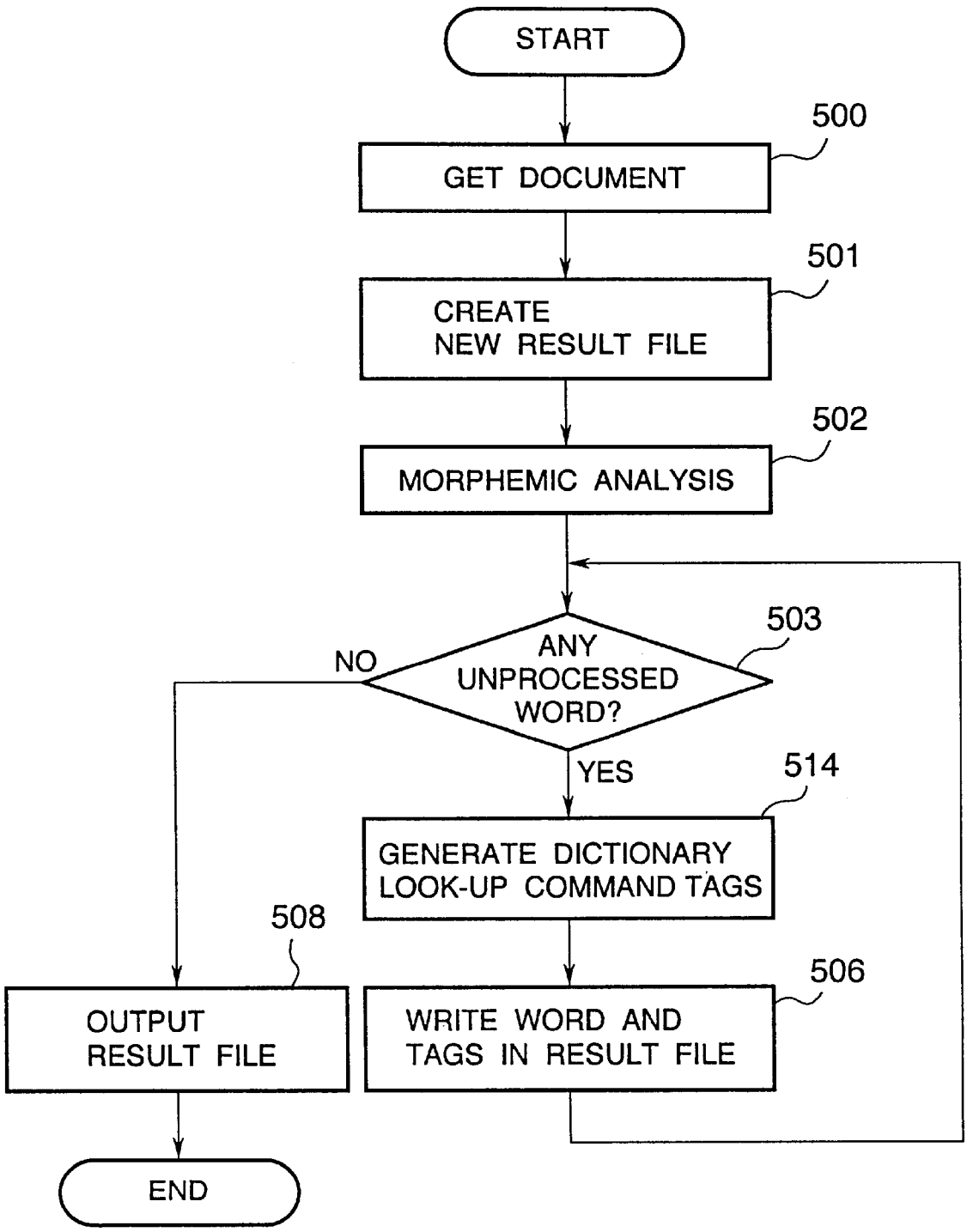

START
500
GET DOCUMENT
501
CREATE NEW RESULT FILE
502
MORPHEMIC ANALYSIS
503
ANY UNPROCESSED WORD?
NO
YES
514
GENERATE DICTIONARY LOOK-UP COMMAND TAGS
506
WRITE WORD AND TAGS IN RESULT FILE
508
OUTPUT RESULT FILE
END

FIG.34

```
< A HREF = "/cgi-bin/pick_dic?we" > We < /A >
< A HREF = "/cgi-bin/pick_dic?draw" > draw < /A >
< A HREF = "/cgi-bin/pick_dic?on" > on < /A >
< A HREF = "/cgi-bin/pick_dic?vast" > vast < /A >
< A HREF = "/cgi-bin/pick_dic?storehouse" > storehouses < /A >
                    .
                    .
                    .
```

FIG.35

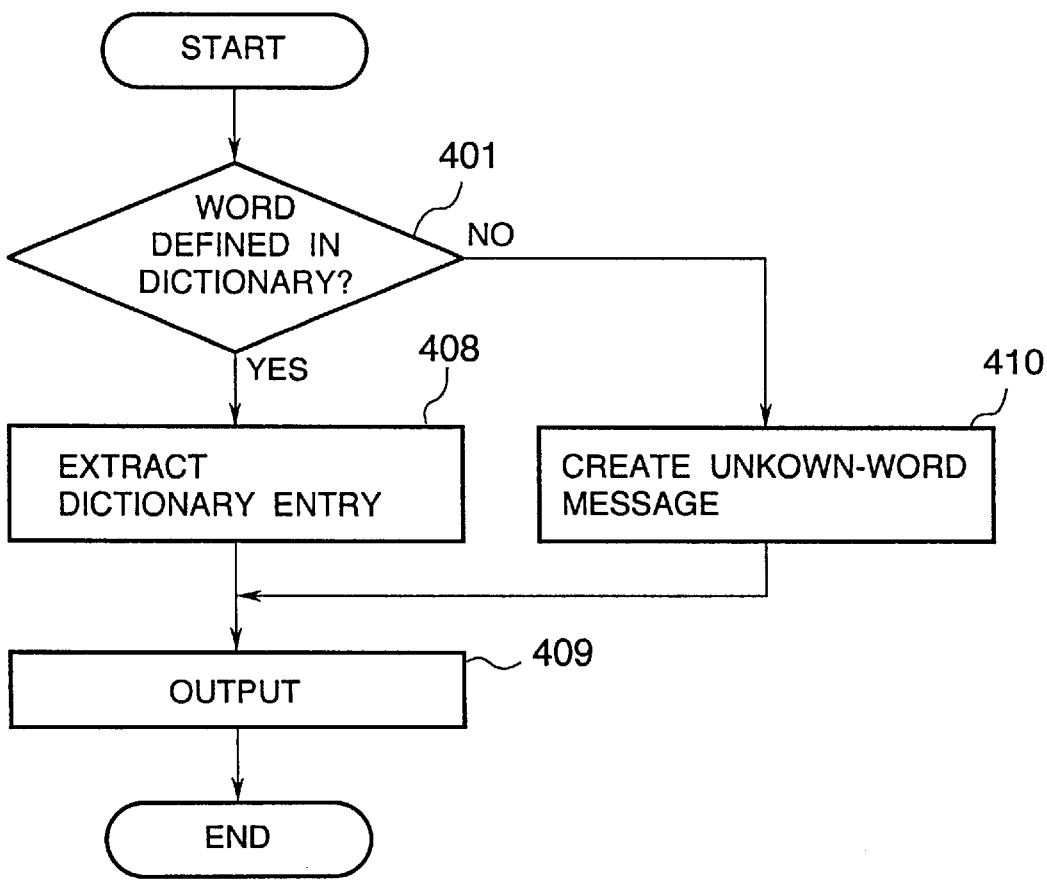


FIG.36

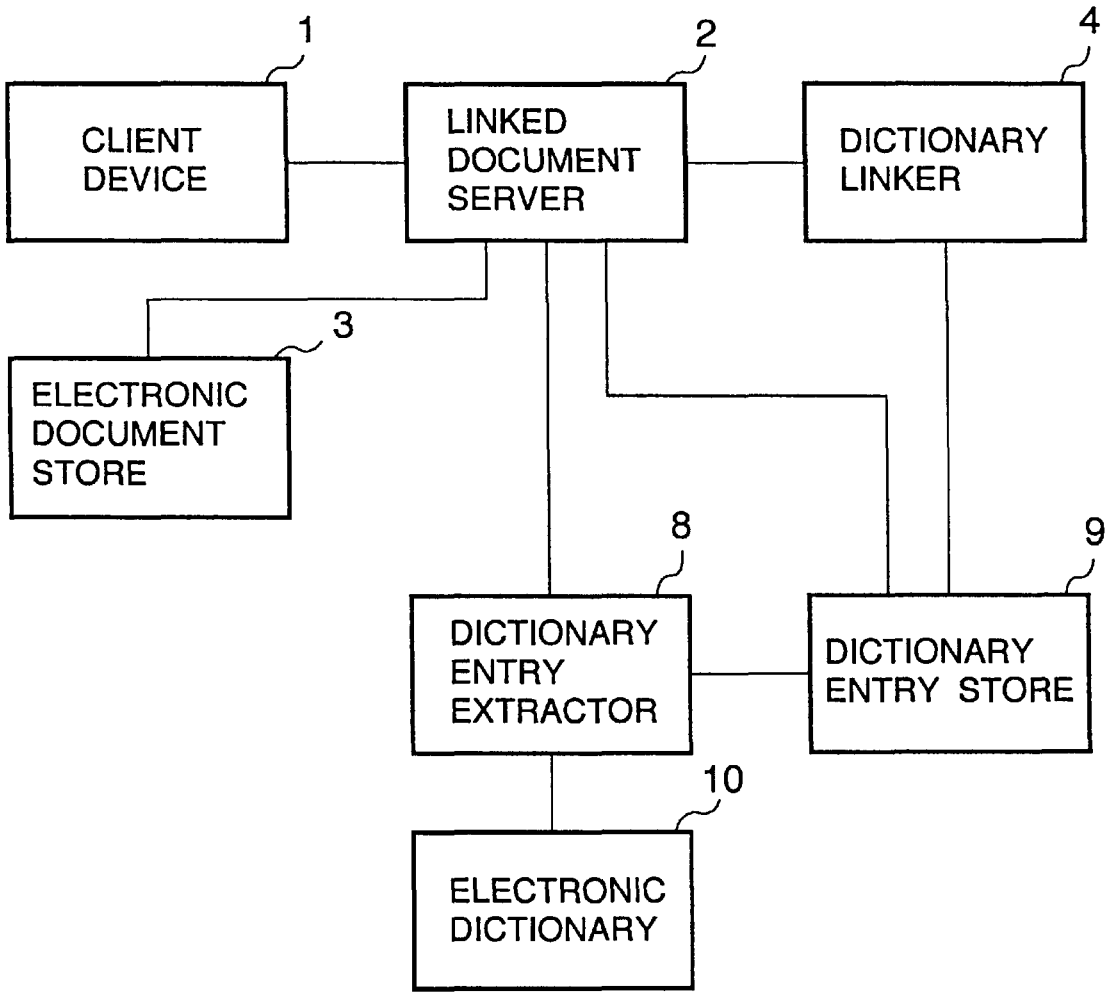

1
2
4
CLIENT DEVICE
LINKED DOCUMENT SERVER
DICTIONARY LINKER
3
ELECTRONIC DOCUMENT STORE
8
9
DICTIONARY ENTRY EXTRACTOR
DICTIONARY ENTRY STORE
10
ELECTRONIC DICTIONARY

FIG.37

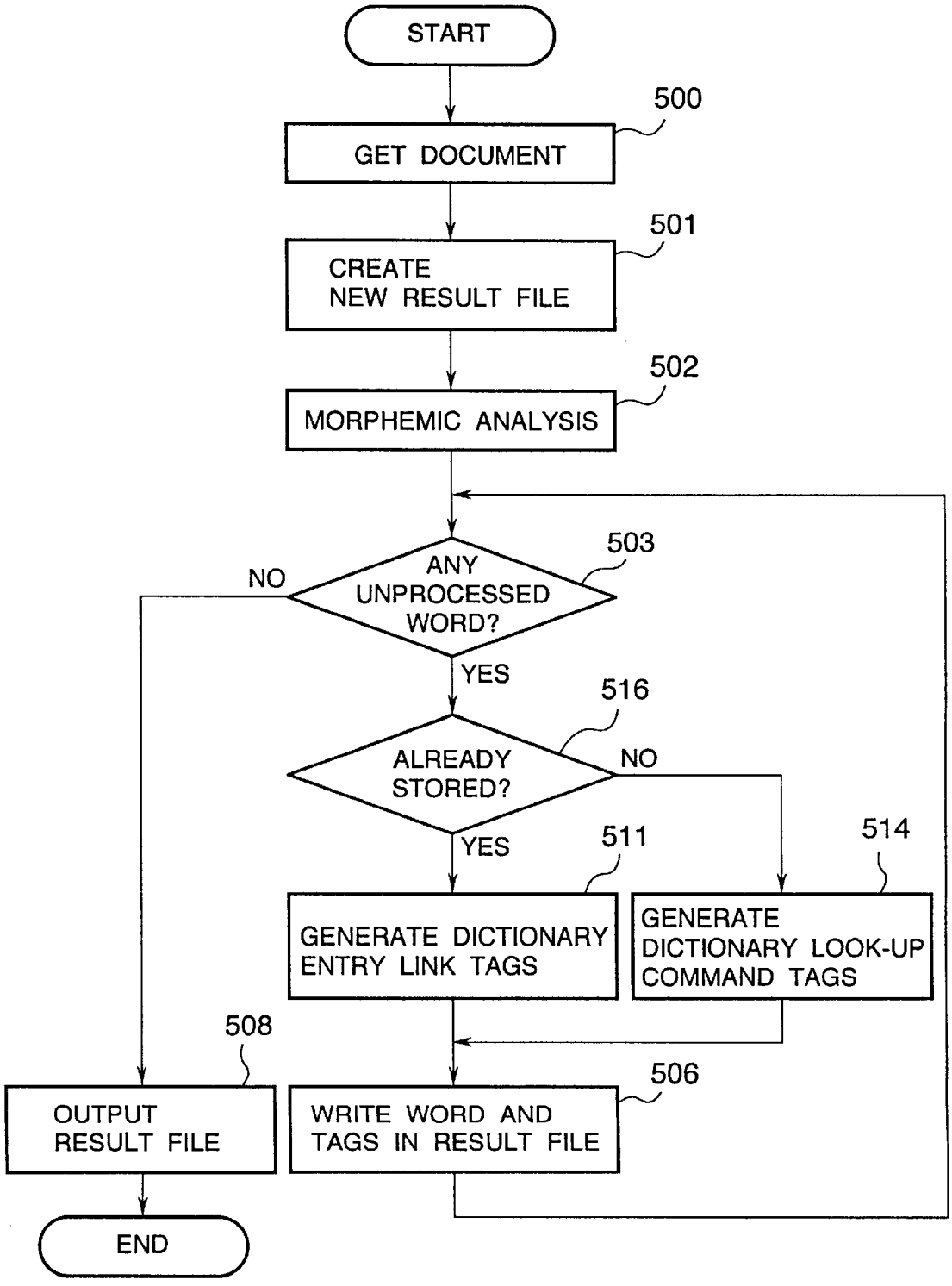

START
500
GET DOCUMENT
501
CREATE NEW RESULT FILE
502
MORPHEMIC ANALYSIS
503
ANY UNPROCESSED WORD?
NO
YES
516
ALREADY STORED?
NO
YES
511
GENERATE DICTIONARY ENTRY LINK TAGS
514
GENERATE DICTIONARY LOOK-UP COMMAND TAGS
506
WRITE WORD AND TAGS IN RESULT FILE
508
OUTPUT RESULT FILE
END

FIG.38

```
< A HREF = "/dic/keep/we" > We < /A >
< A HREF = "/cgi-bin/pick_dic?draw" > draw < /A >
< A HREF = "/dic/keep/on">on < /A >
< A HREF = "/cgi-bin/pick_dic?vast" > vast < /A >
< A HREF = "/cgi-bin/pick_dic?storehouse" > storehouses < /A >
```

FIG.39

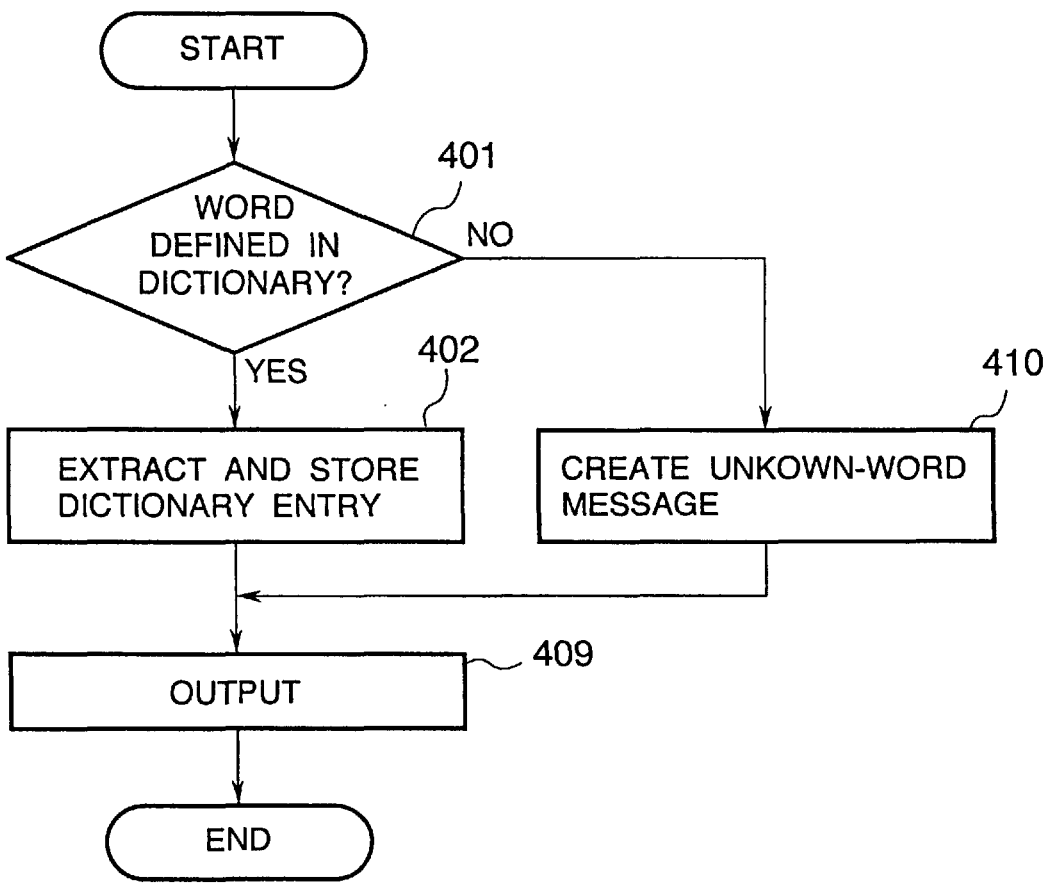


FIG.40

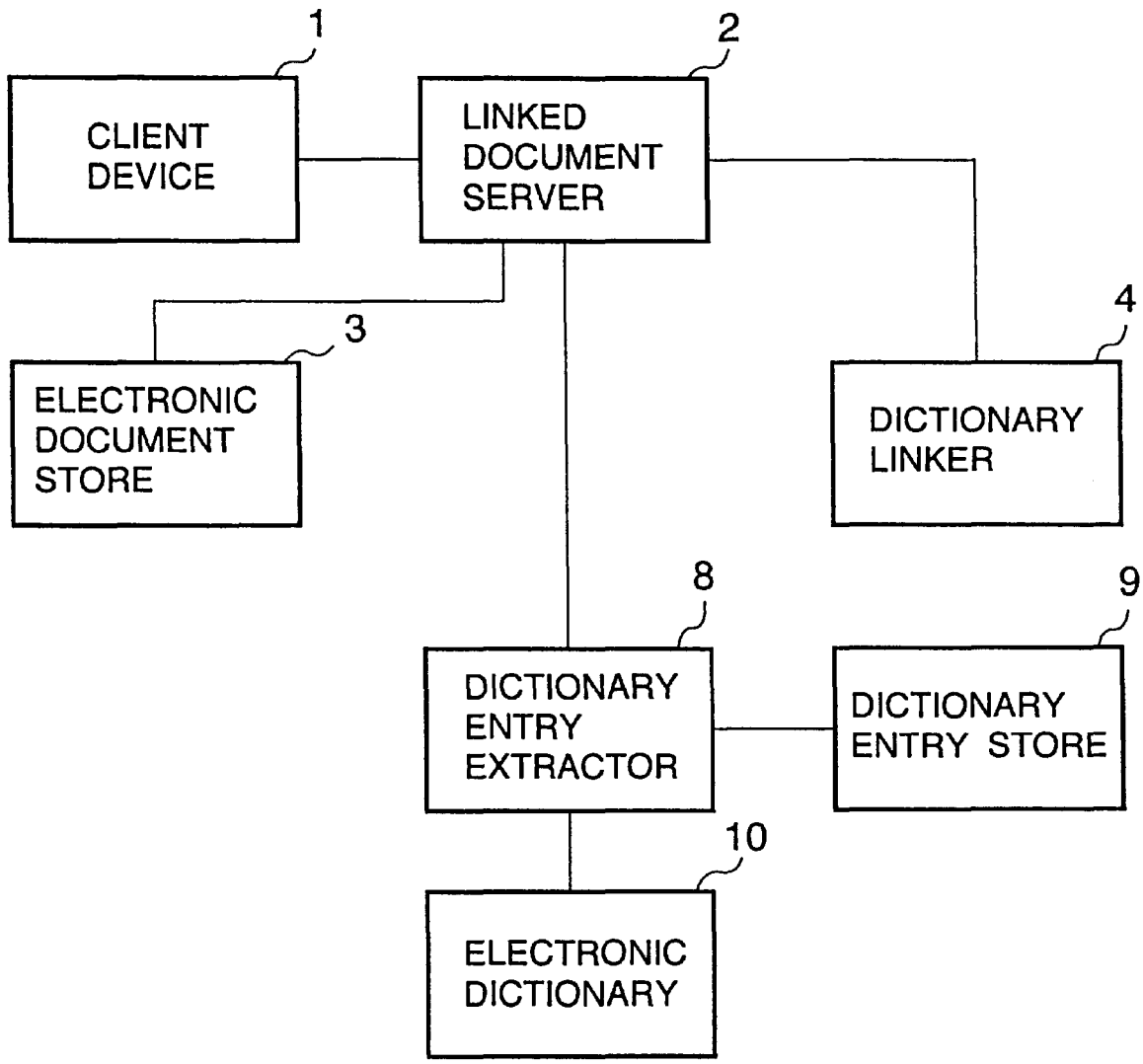

1
2
CLIENT DEVICE
LINKED DOCUMENT SERVER
3
4
ELECTRONIC DOCUMENT STORE
DICTIONARY LINKER
8
9
DICTIONARY ENTRY EXTRACTOR
DICTIONARY ENTRY STORE
10
ELECTRONIC DICTIONARY

FIG.41

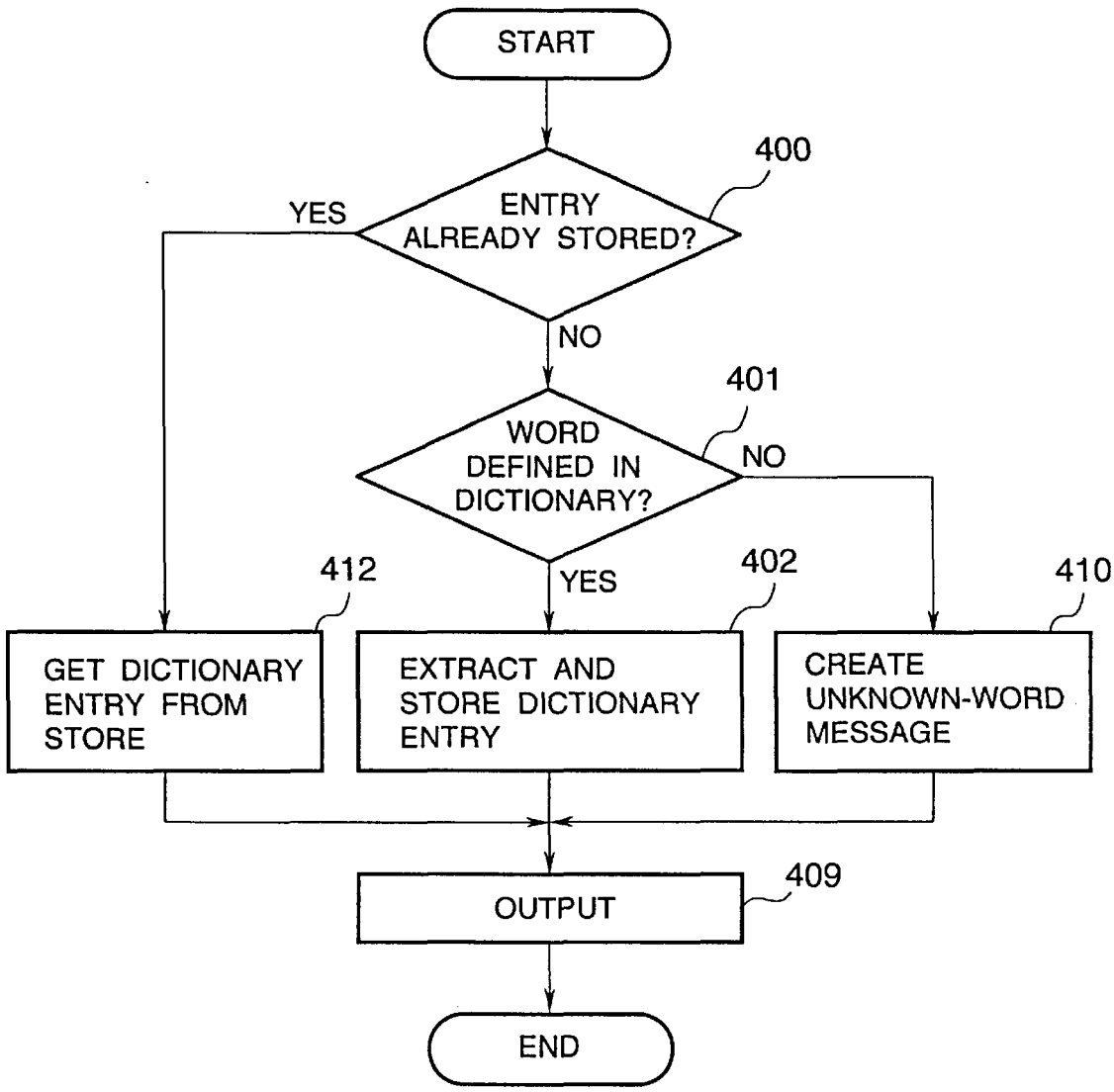

START
400
ENTRY ALREADY STORED?
YES
NO
401
WORD DEFINED IN DICTIONARY?
NO
YES
412
402
410
GET DICTIONARY ENTRY FROM STORE
EXTRACT AND STORE DICTIONARY ENTRY
CREATE UNKNOWN-WORD MESSAGE
409
OUTPUT
END

FIG.42

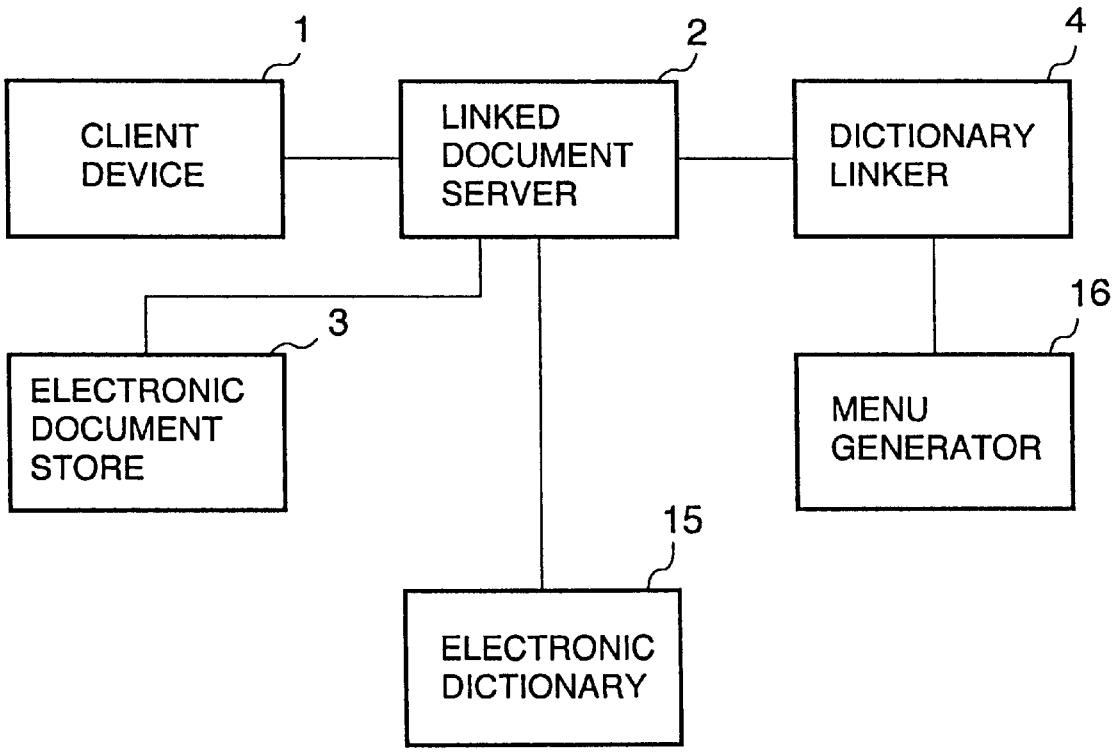

1
CLIENT DEVICE
2
LINKED DOCUMENT SERVER
4
DICTIONARY LINKER
3
ELECTRONIC DOCUMENT STORE
16
MENU GENERATOR
15
ELECTRONIC DICTIONARY

FIG.43

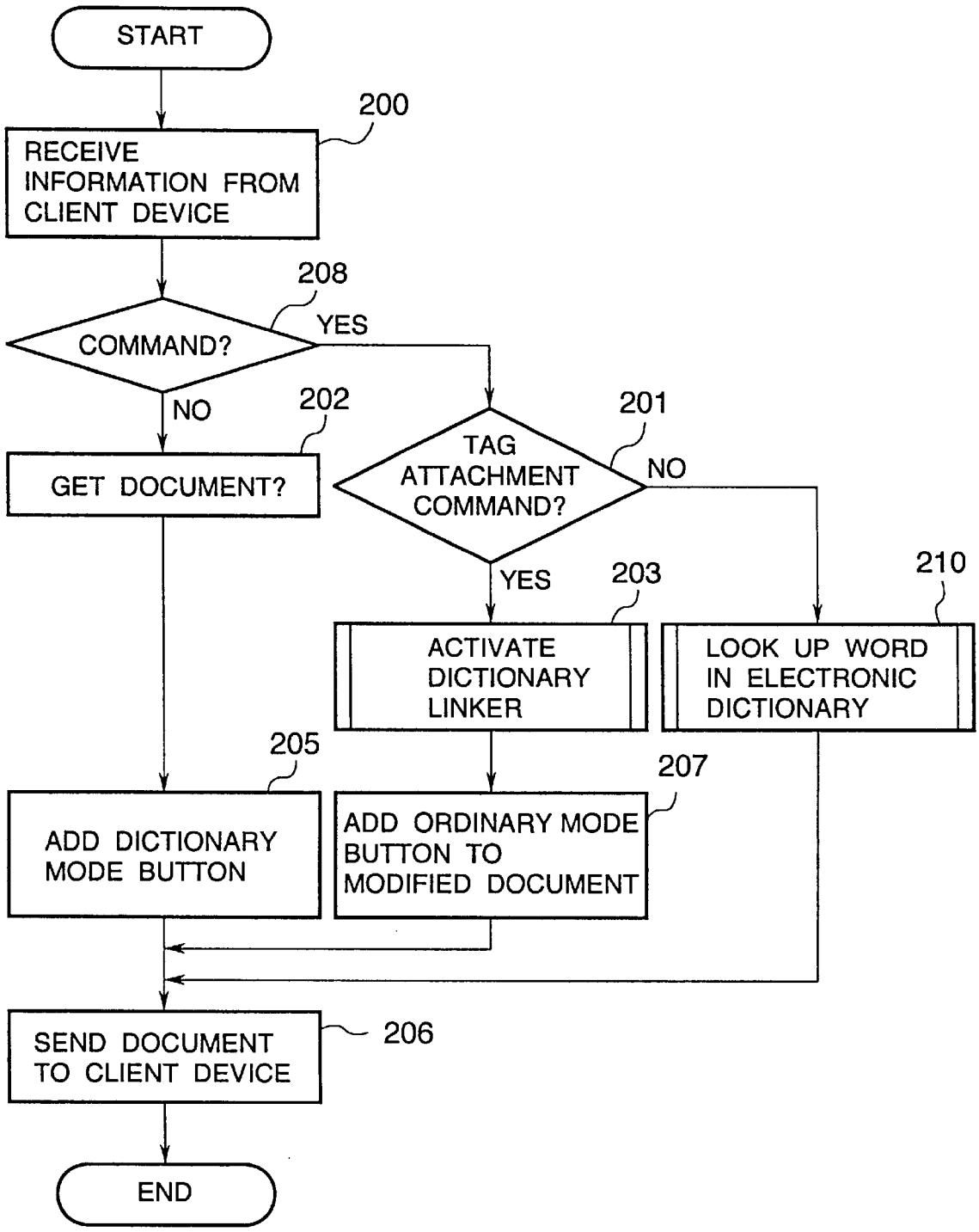

START
200
RECEIVE INFORMATION FROM CLIENT DEVICE
208
COMMAND?
YES
NO
202
GET DOCUMENT?
201
TAG ATTACHMENT COMMAND?
NO
YES
203
ACTIVATE DICTIONARY LINKER
210
LOOK UP WORD IN ELECTRONIC DICTIONARY
205
ADD DICTIONARY MODE BUTTON
207
ADD ORDINARY MODE BUTTON TO MODIFIED DOCUMENT
SEND DOCUMENT TO CLIENT DEVICE
206
END

FIG.44

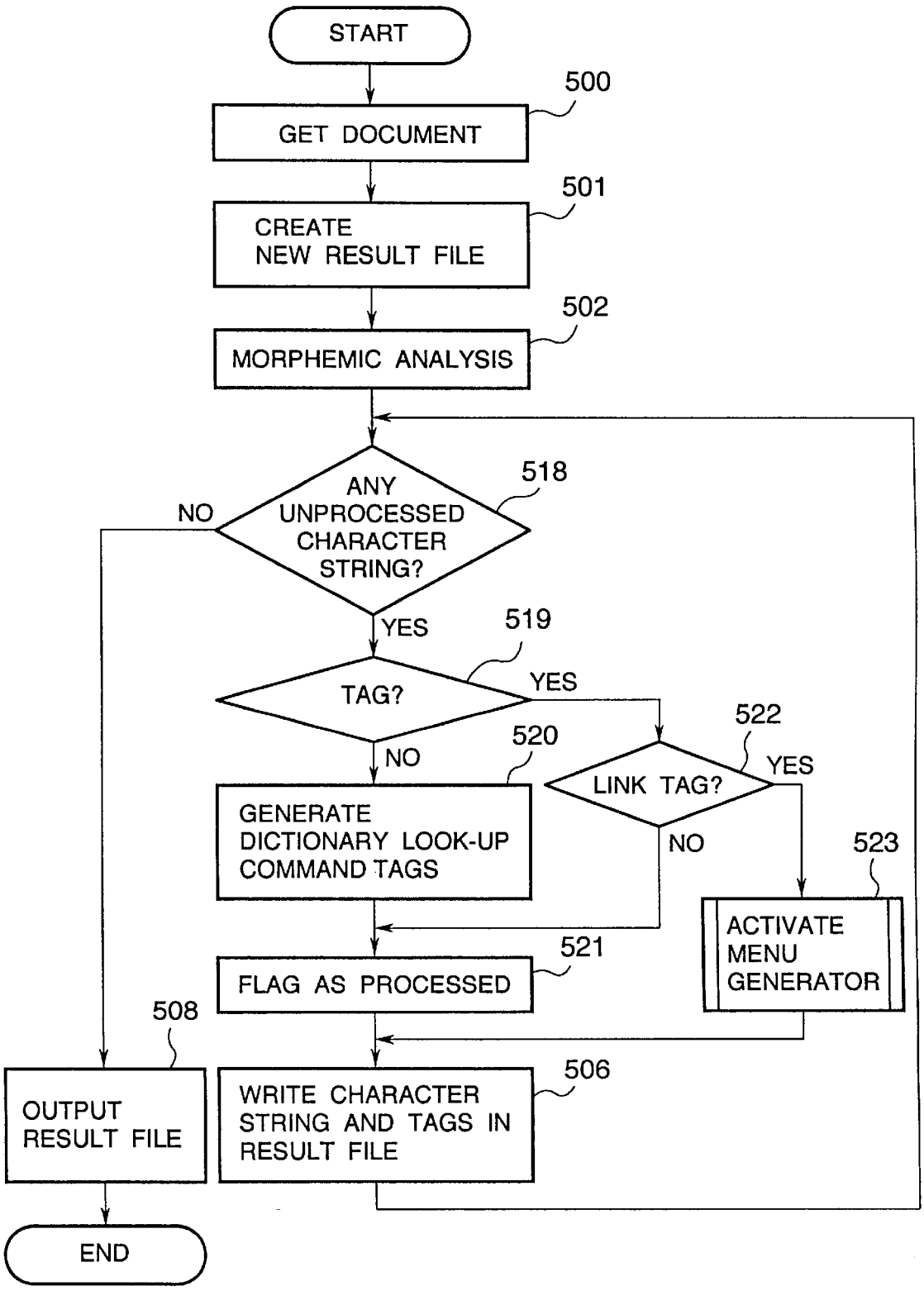

START
500
GET DOCUMENT
501
CREATE NEW RESULT FILE
502
MORPHEMIC ANALYSIS
518
ANY UNPROCESSED CHARACTER STRING?
NO
YES
519
TAG?
YES
NO
522
LINK TAG?
YES
NO
520
GENERATE DICTIONARY LOOK-UP COMMAND TAGS
523
ACTIVATE MENU GENERATOR
521
FLAG AS PROCESSED
508
OUTPUT RESULT FILE
506
WRITE CHARACTER STRING AND TAGS IN RESULT FILE
END

FIG.45

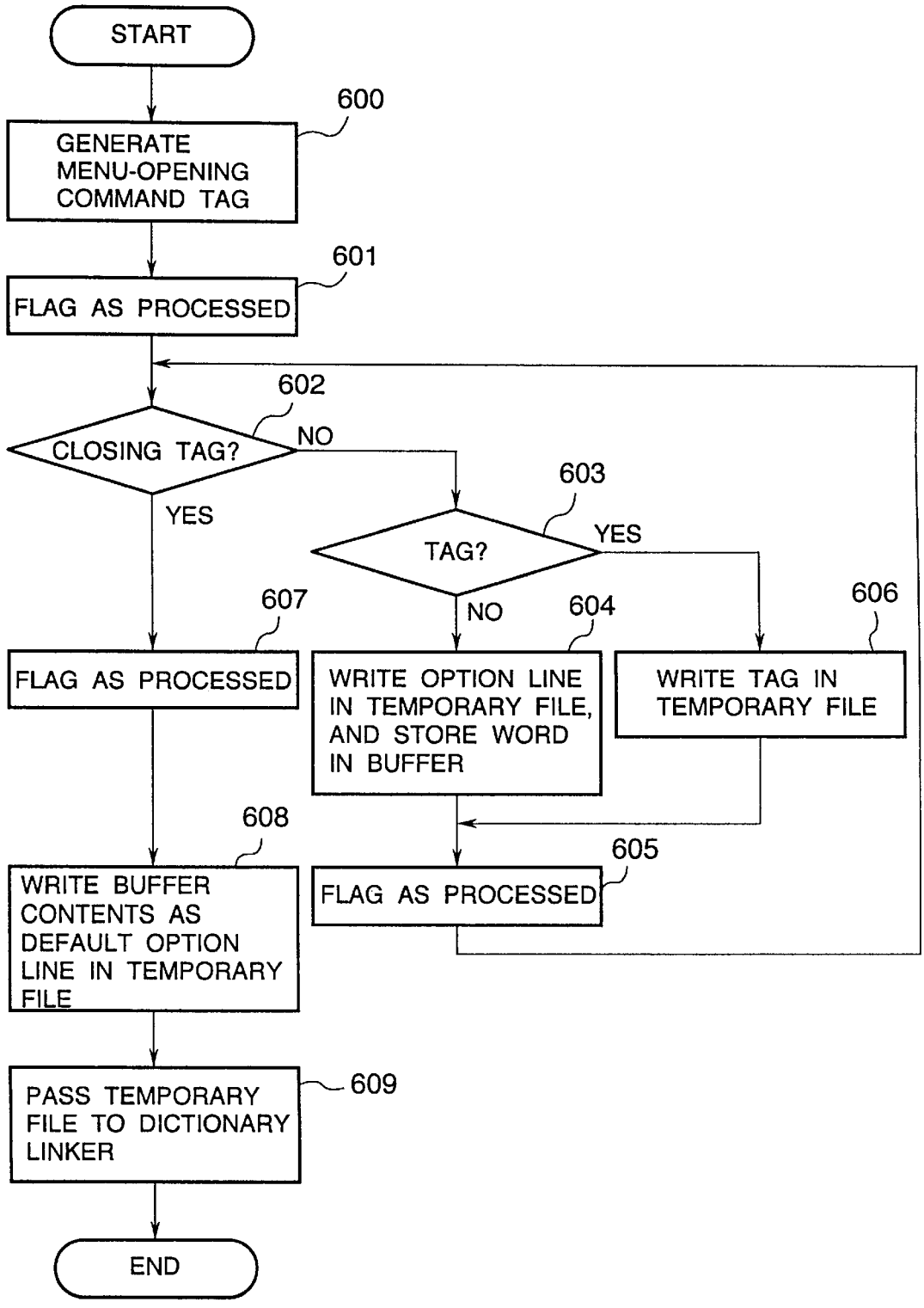

START
600
GENERATE MENU-OPENING COMMAND TAG
601
FLAG AS PROCESSED
602
CLOSING TAG?
NO
YES
603
TAG?
YES
NO
604
606
607
FLAG AS PROCESSED
WRITE OPTION LINE IN TEMPORARY FILE, AND STORE WORD IN BUFFER
WRITE TAG IN TEMPORARY FILE
608
605
WRITE BUFFER CONTENTS AS DEFAULT OPTION LINE IN TEMPORARY FILE
FLAG AS PROCESSED
609
PASS TEMPORARY FILE TO DICTIONARY LINKER
END

FIG.46

```
22
< A HREF = "/cgi-bin/into_the_dic" > DICTIONARY MODE < /A >

Corporate Glance < BR >

< DL >
< DD > < A HREF = "message" > Message from the President < /A >
< DD > < A HREF = "slogan" > Global Slogan < /A >
< DD > < A HREF = "glance" > OKI Electric at a Glance < /A >
< DL >
```

FIG.47

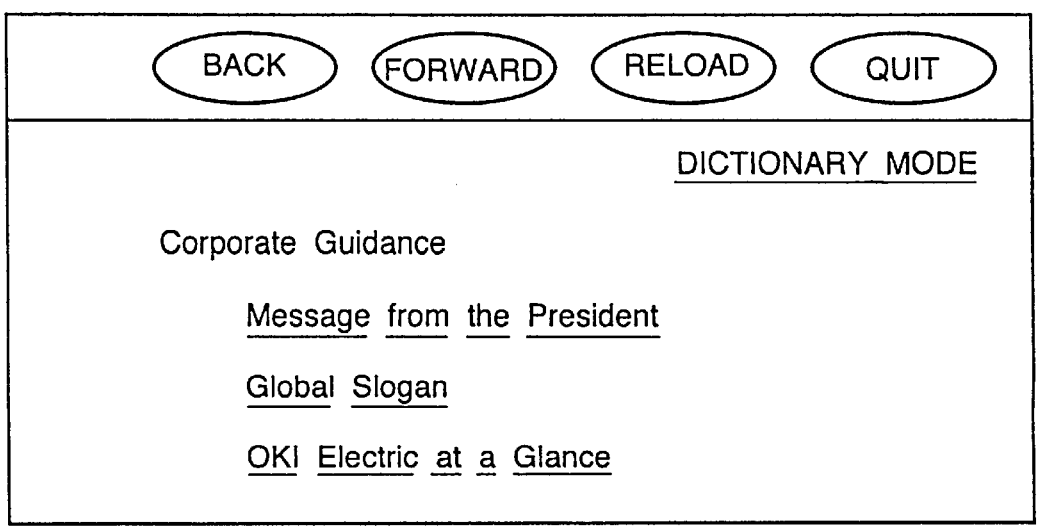


FIG.48

```
35
  < A  HREF = " guidance" > ORDINARY  MODE  < /A >

  < A  HREF = "/cgi-bin/look_up?corporate" > Corporate  < /A >
  < A  HREF = "/cgi-bin/look_up?guidance" >  Guidance  < /A > < BR >

  < DL >
  < DD >
  < SELECT  name  = " select1 " onFocus = 'ref(message)' onChange
                                          = 'lookdic(option)  >
34
  < OPTION > message (dic)
  < OPTION > from (dic)
  < OPTION > the (dic)
  < OPTION > President (dic)
  < OPTION SELECTED >  Message  from  the  President
  < /SELECT >
  < DD >
```

FIG.49

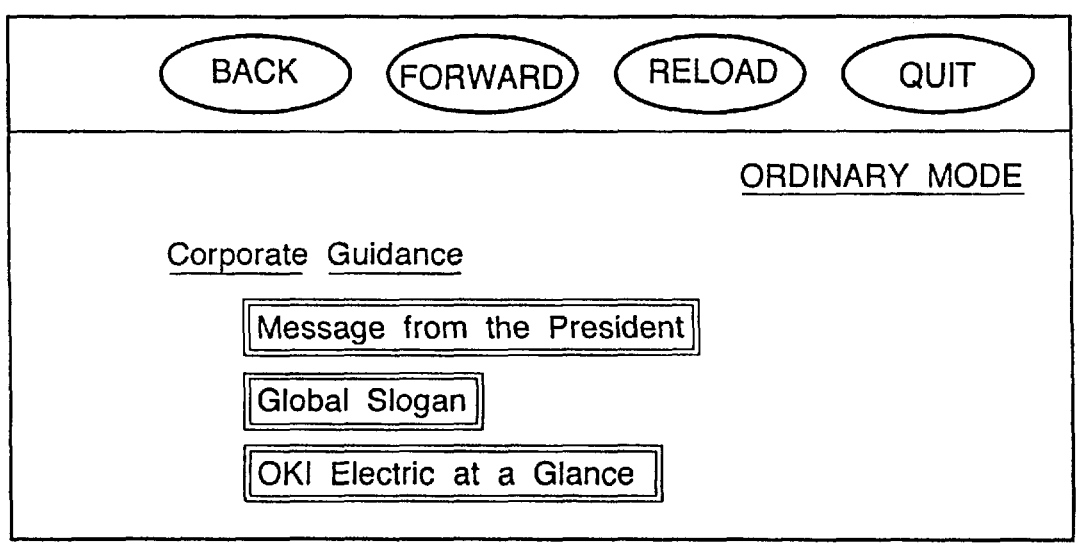


FIG.50

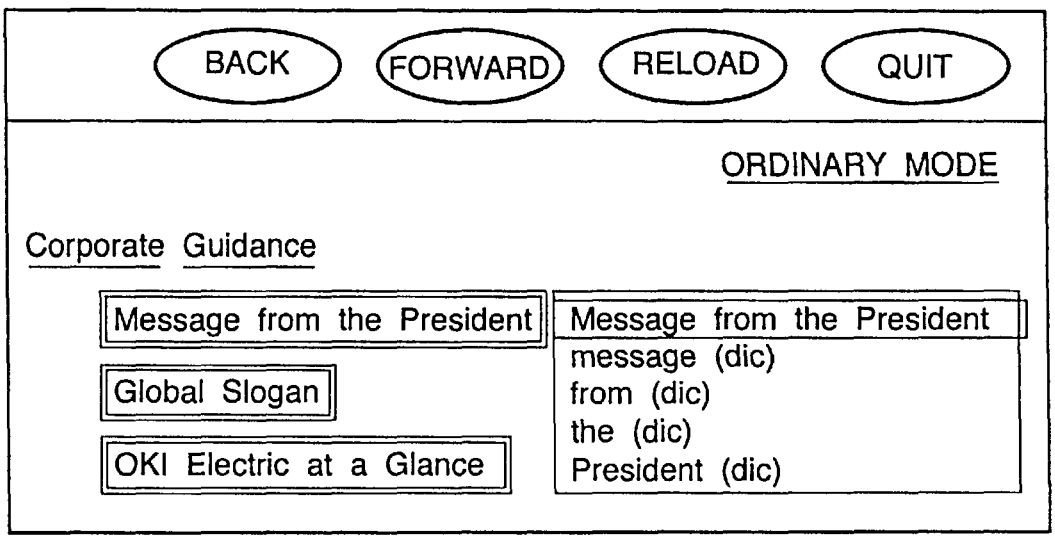

BACK
FORWARD
RELOAD
QUIT
ORDINARY MODE
Corporate Guidance
Message from the President
Global Slogan
OKI Electric at a Glance
Message from the President
message (dic)
from (dic)
the (dic)
President (dic)

FIG.51

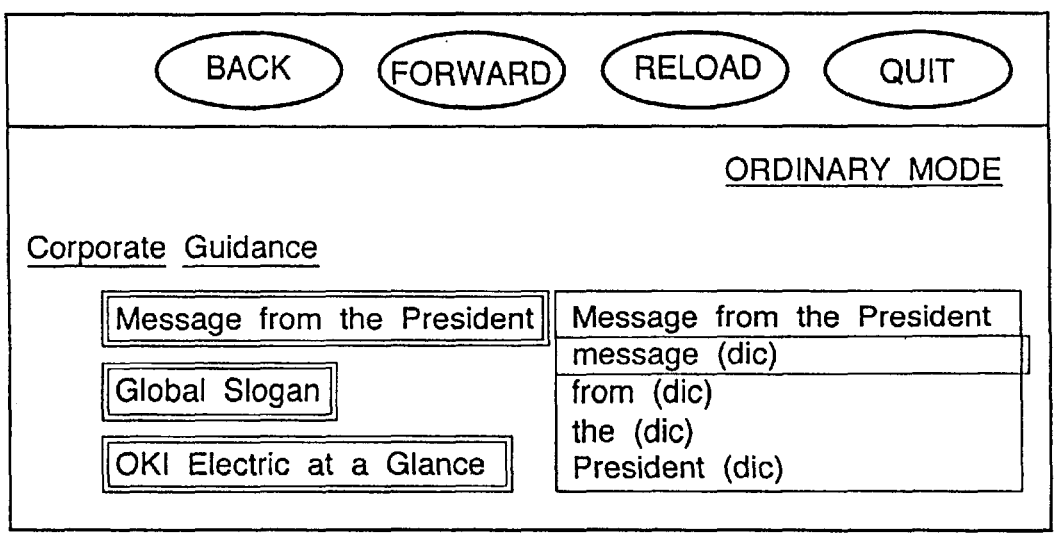

BACK
FORWARD
RELOAD
QUIT
ORDINARY MODE
Corporate Guidance
Message from the President
Global Slogan
OKI Electric at a Glance
Message from the President
message (dic)
from (dic)
the (dic)
President (dic)

FIG.52

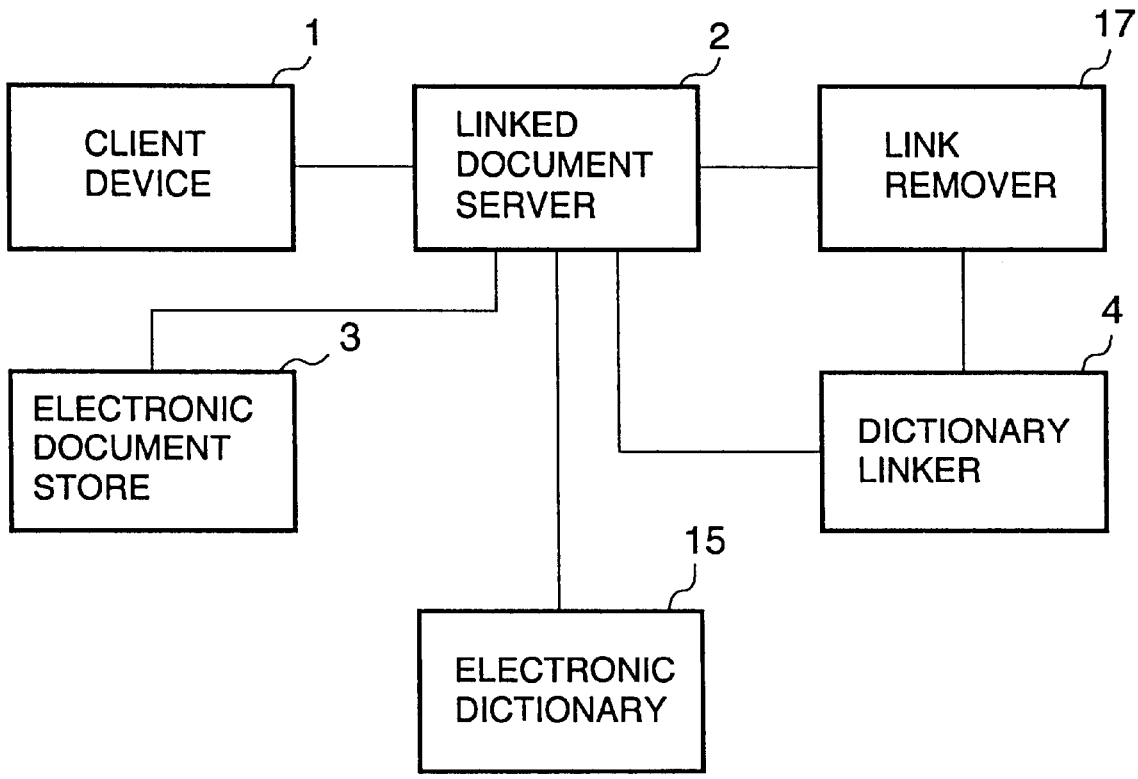

1
CLIENT DEVICE
2
LINKED DOCUMENT SERVER
17
LINK REMOVER
3
ELECTRONIC DOCUMENT STORE
4
DICTIONARY LINKER
15
ELECTRONIC DICTIONARY

FIG.53

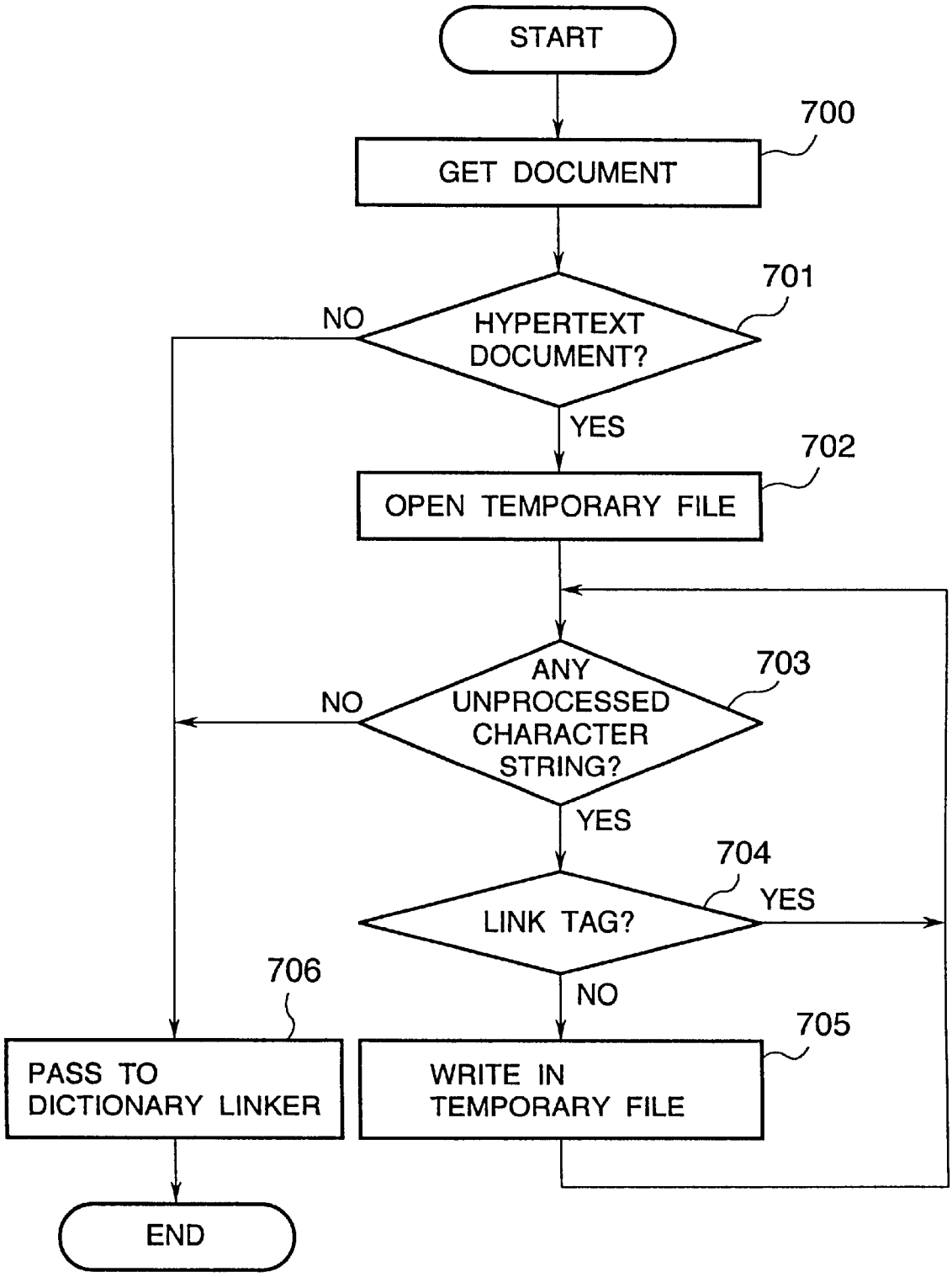

START
700
GET DOCUMENT
701
HYPERTEXT DOCUMENT?
NO
YES
702
OPEN TEMPORARY FILE
703
ANY UNPROCESSED CHARACTER STRING?
NO
YES
704
LINK TAG?
YES
NO
705
WRITE IN TEMPORARY FILE
706
PASS TO DICTIONARY LINKER
END

FIG.54

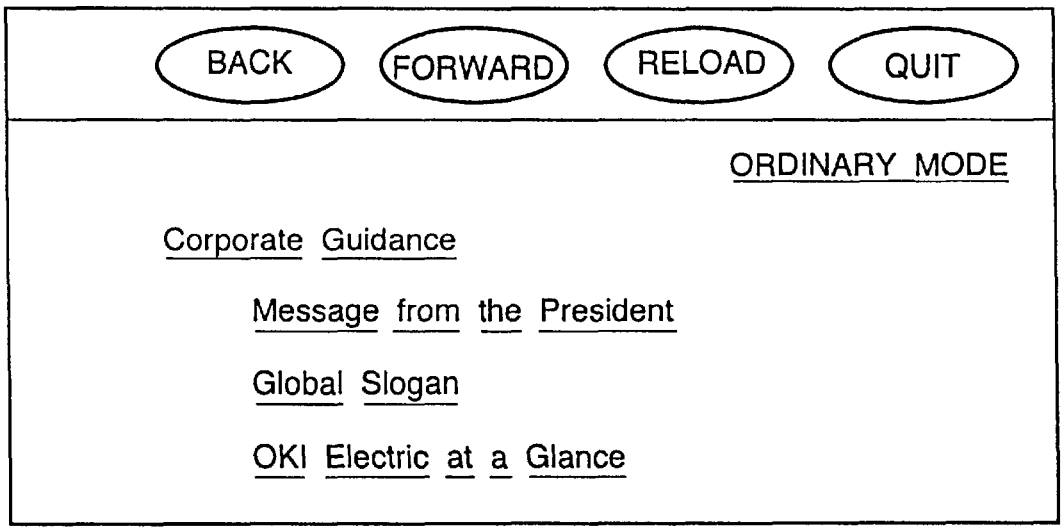

BACK
FORWARD
RELOAD
QUIT
ORDINARY MODE
Corporate Guidance
Message from the President
Global Slogan
OKI Electric at a Glance

FIG.55

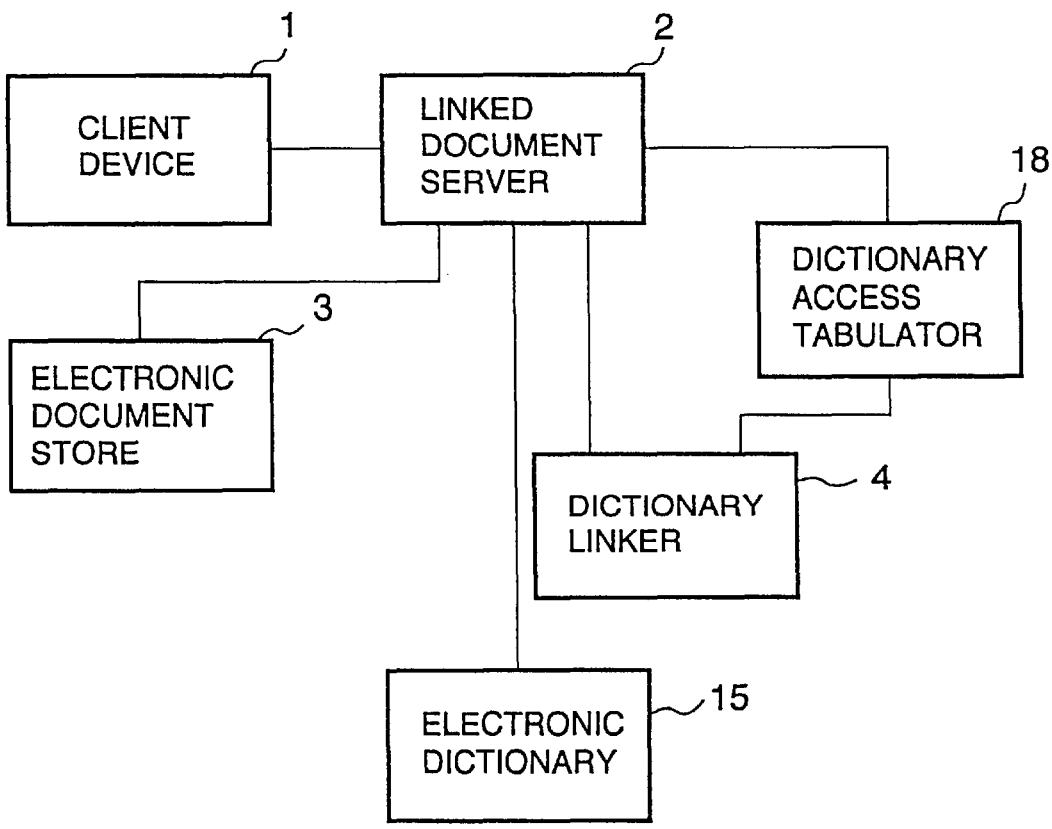

1
CLIENT DEVICE
2
LINKED DOCUMENT SERVER
18
DICTIONARY ACCESS TABULATOR
3
ELECTRONIC DOCUMENT STORE
4
DICTIONARY LINKER
15
ELECTRONIC DICTIONARY

FIG.56

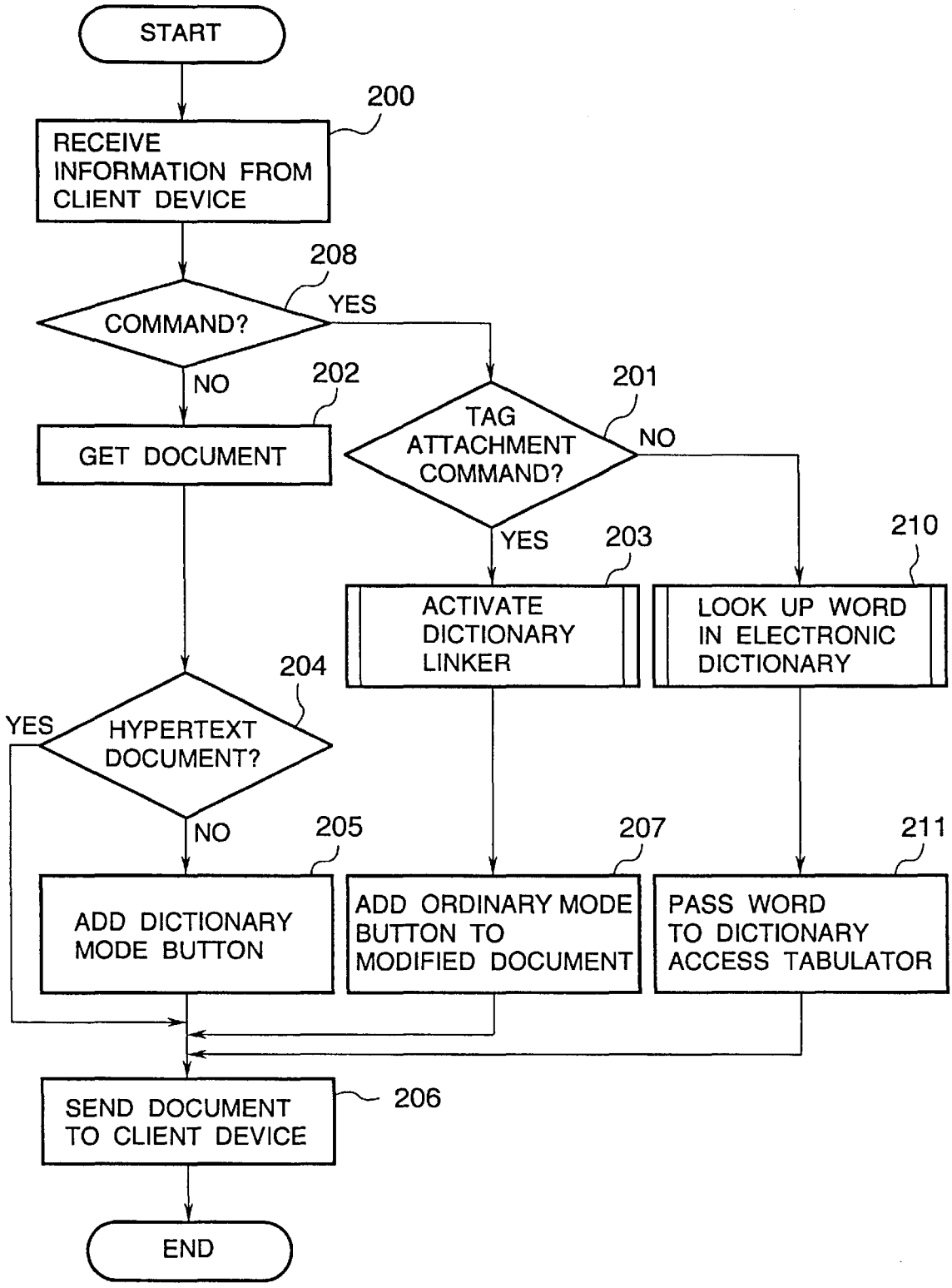

START
200
RECEIVE INFORMATION FROM CLIENT DEVICE
208
COMMAND?
YES
NO
202
GET DOCUMENT
201
TAG ATTACHMENT COMMAND?
NO
YES
203
ACTIVATE DICTIONARY LINKER
210
LOOK UP WORD IN ELECTRONIC DICTIONARY
204
YES
HYPERTEXT DOCUMENT?
NO
205
ADD DICTIONARY MODE BUTTON
207
ADD ORDINARY MODE BUTTON TO MODIFIED DOCUMENT
211
PASS WORD TO DICTIONARY ACCESS TABULATOR
206
SEND DOCUMENT TO CLIENT DEVICE
END

FIG.57

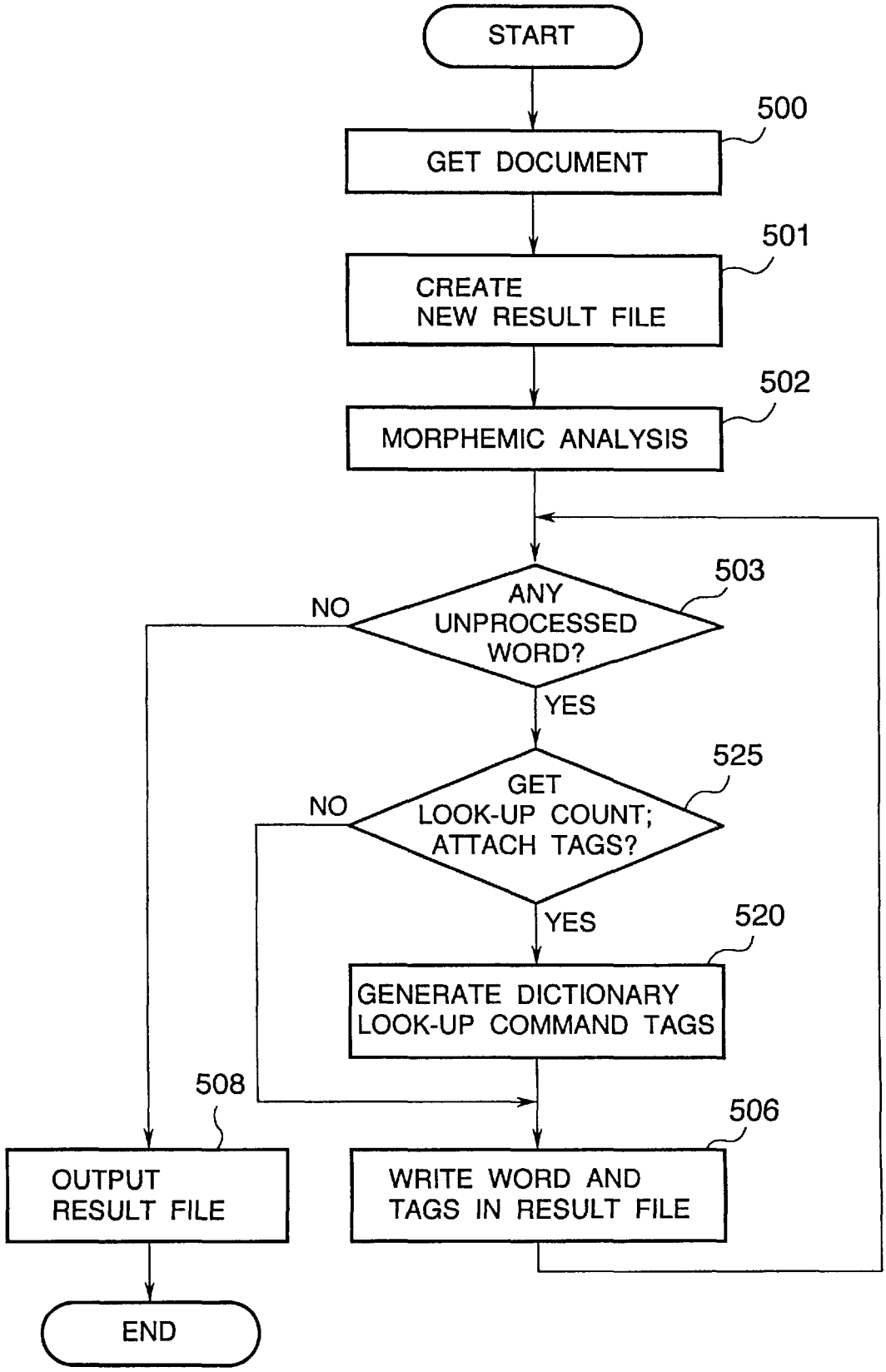

START
500
GET DOCUMENT
501
CREATE NEW RESULT FILE
502
MORPHEMIC ANALYSIS
503
ANY UNPROCESSED WORD?
NO
YES
525
GET LOOK-UP COUNT; ATTACH TAGS?
NO
YES
520
GENERATE DICTIONARY LOOK-UP COMMAND TAGS
508
506
OUTPUT RESULT FILE
WRITE WORD AND TAGS IN RESULT FILE
END

FIG.58

WORD	LOOK-UP COUNT
we	1
draw	3
on	1
vast	5
storehouse	10
⋮	⋮

FIG.59

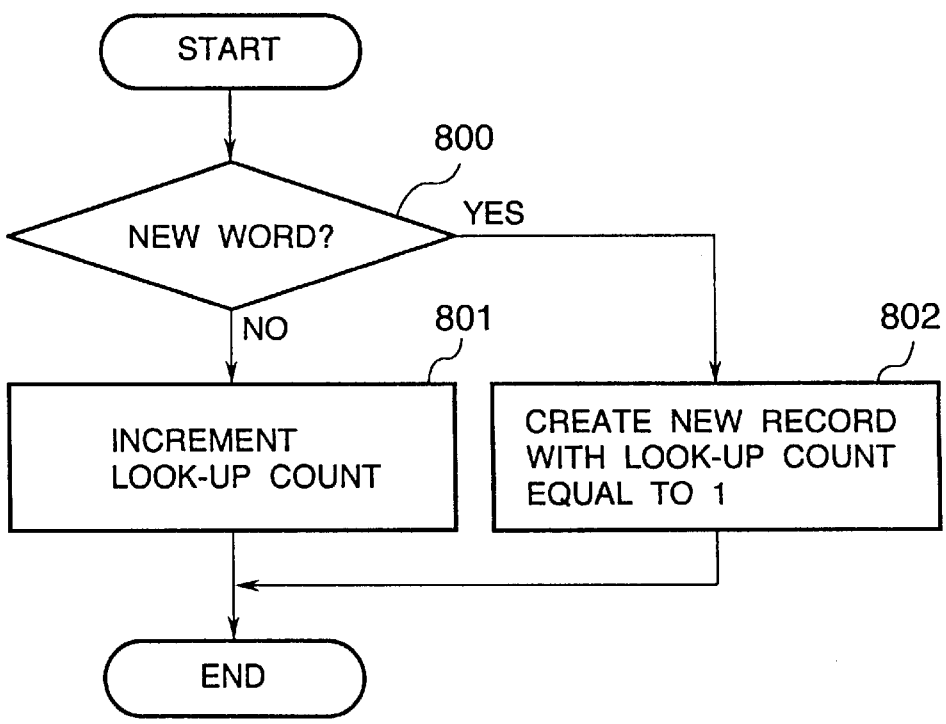


FIG.60

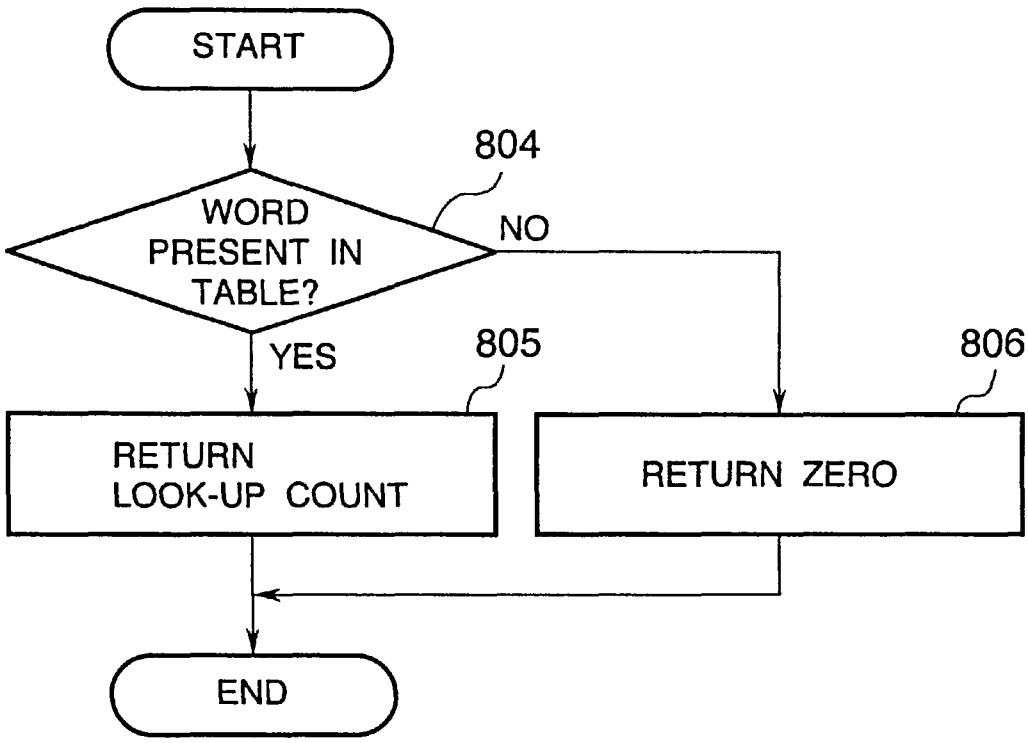


FIG.61

```
< A  HREF = "slogan" > ORDINARY  MODE  < /A >

We
< A  HREF = "/cgi-bin/look_up?draw" > draw  < /A >
on
< A  HREF = "/cgi-bin/look_up?vast" > vast < /A >
< A  HREF = "/cgi-bin/look_up?storehouse" > storehouses  < /A >
                                    .
                                    .
                                    .
                                    .
```

FIG.62

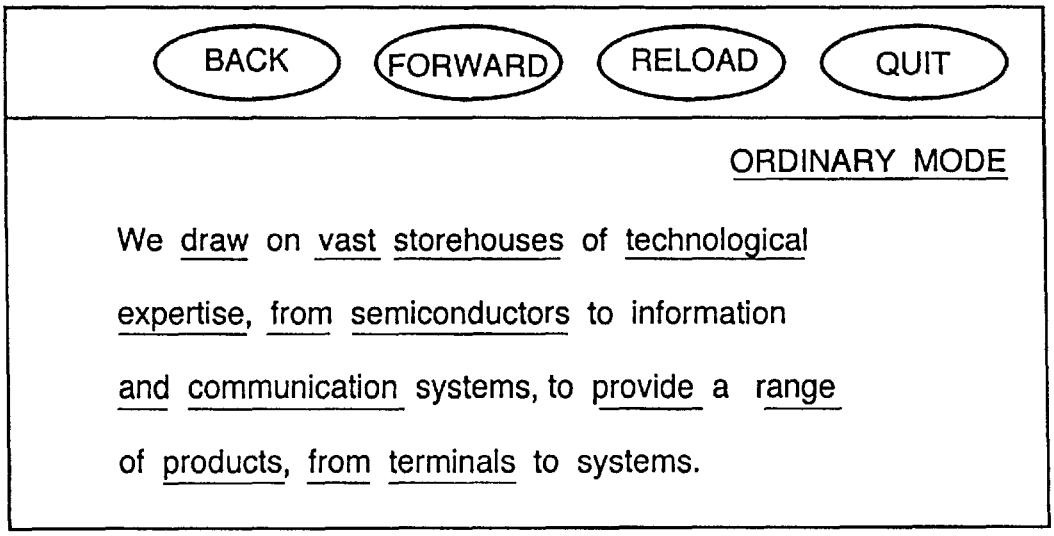


FIG.63

```
< FONT = BLUE >
< A HREF = "/cgi-bin/look_up?we" > We < /A >
< /FONT >
< FONT = GREEN >
< A HREF = "/cgi-bin/look_up?draw" > draw < /A >
< /FONT >
< FONT = BLUE >
< A HREF = "/cgi-bin/look_up?on" > on < /A >
< /FONT >
< FONT = RED >
< A HREF = "/cgi-bin/look_up?vast" > vast < /A >
< /FONT >
< FONT = GREEN >
< A HREF = "/cgi-bin/look_up?storehouse" > storehouses < /A >
< /FONT >
```

FIG.64

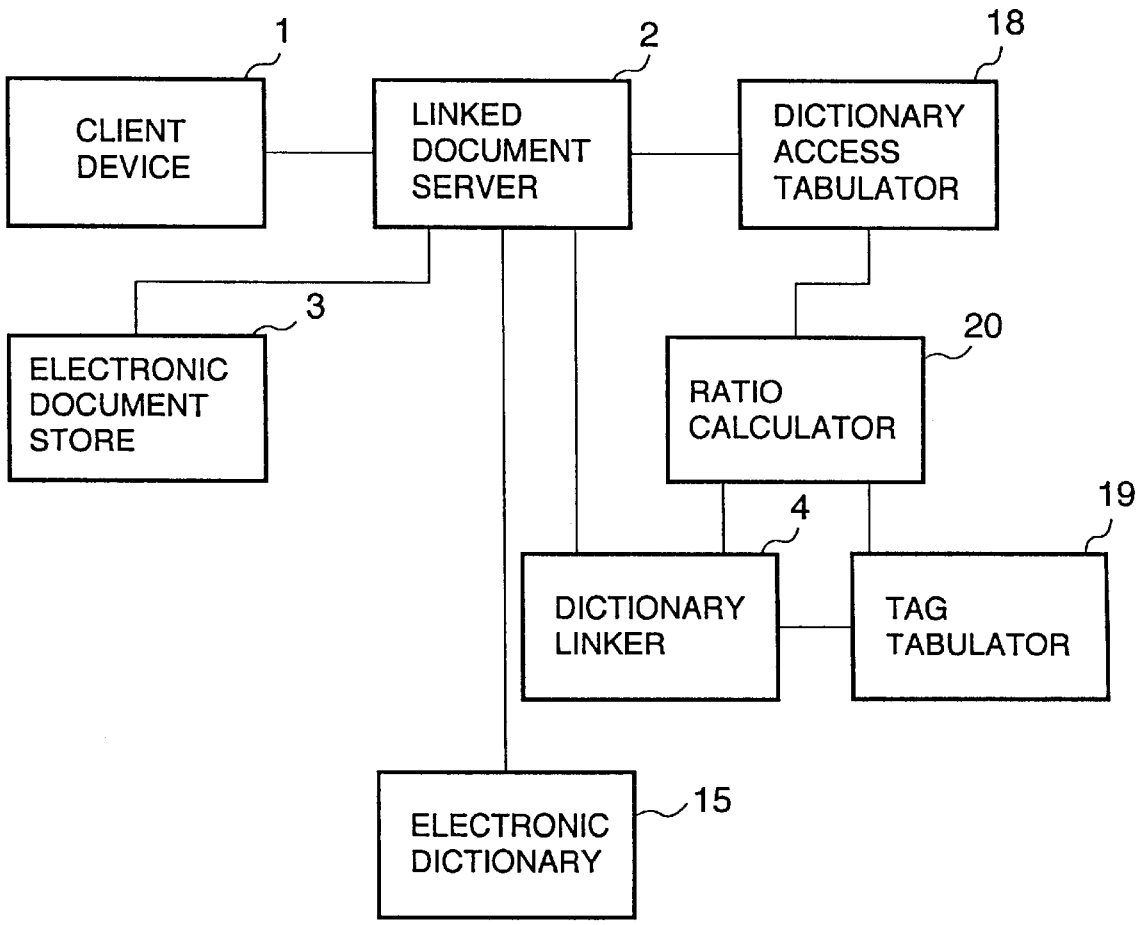

1
CLIENT DEVICE
2
LINKED DOCUMENT SERVER
18
DICTIONARY ACCESS TABULATOR
3
ELECTRONIC DOCUMENT STORE
20
RATIO CALCULATOR
4
19
DICTIONARY LINKER
TAG TABULATOR
15
ELECTRONIC DICTIONARY

FIG.65

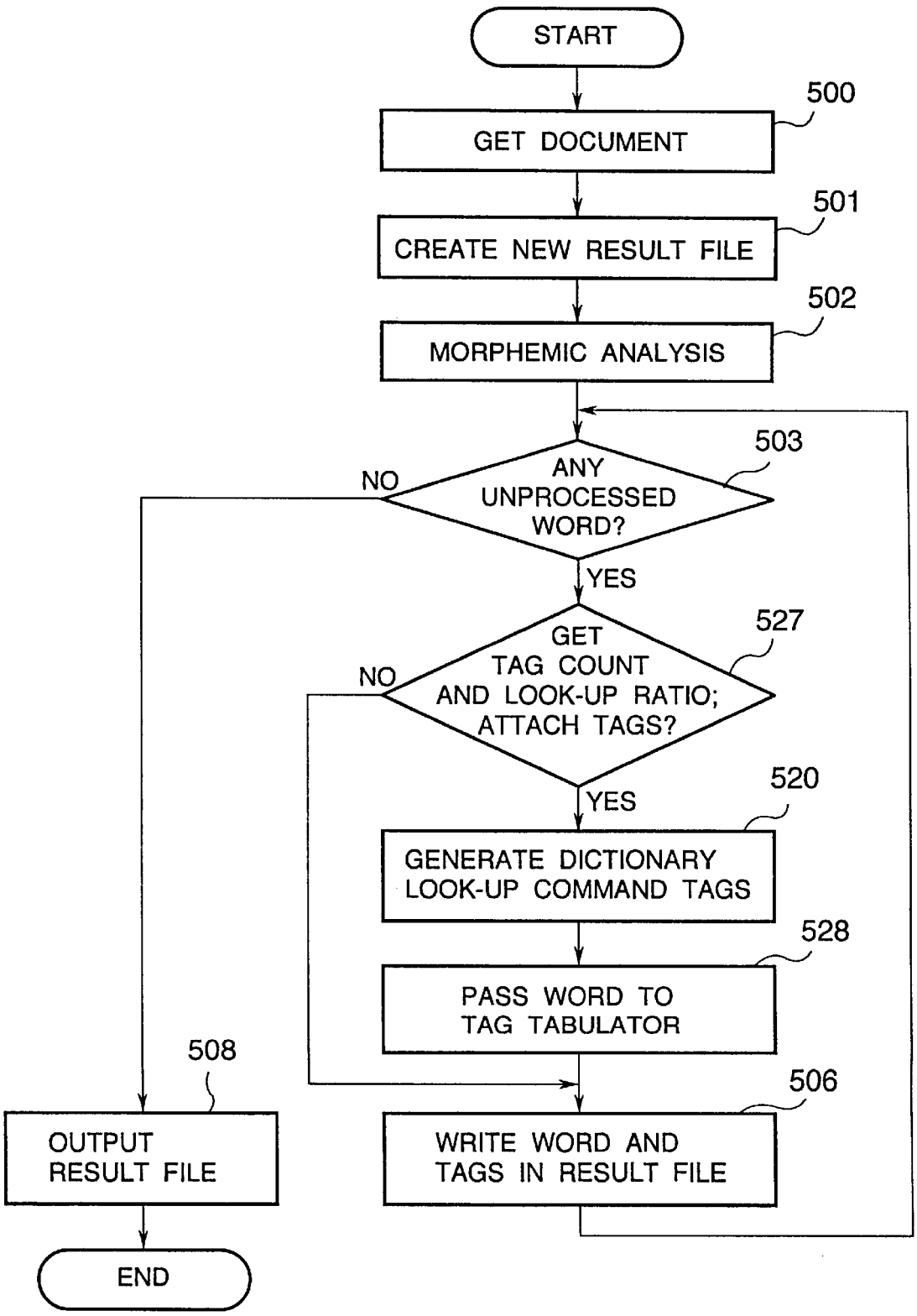
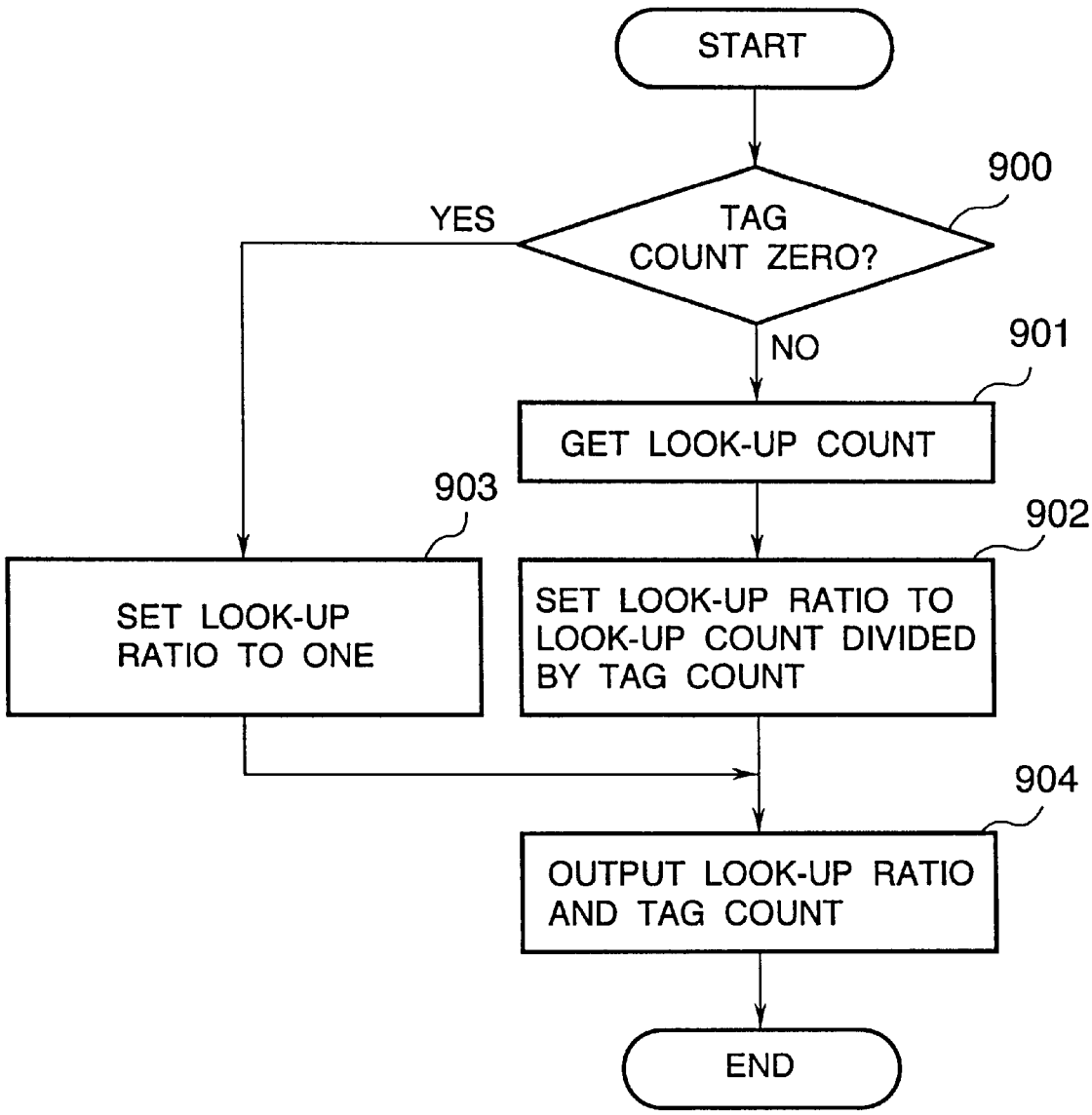

START
500
GET DOCUMENT
501
CREATE NEW RESULT FILE
502
MORPHEMIC ANALYSIS
503
ANY UNPROCESSED WORD?
NO
YES
527
GET TAG COUNT AND LOOK-UP RATIO; ATTACH TAGS?
NO
YES
520
GENERATE DICTIONARY LOOK-UP COMMAND TAGS
528
PASS WORD TO TAG TABULATOR
506
WRITE WORD AND TAGS IN RESULT FILE
508
OUTPUT RESULT FILE
END

FIG.66


START
900
TAG COUNT ZERO?
YES
NO
901
GET LOOK-UP COUNT
903
SET LOOK-UP RATIO TO ONE
902
SET LOOK-UP RATIO TO LOOK-UP COUNT DIVIDED BY TAG COUNT
904
OUTPUT LOOK-UP RATIO AND TAG COUNT
END

DOCUMENT DISPLAY SYSTEM AND ELECTRONIC DICTIONARY

BACKGROUND OF THE INVENTION

The present invention relates to a document display system and an electronic dictionary, more particularly to a document display system that enables words in an electronic document to be looked up easily in an electronic dictionary.

Dictionaries have recently become available on media, such as disks, that make the dictionary entries accessible to a personal computer or other electronic device. To look up a word in such an electronic dictionary, the user types the word on a keyboard; the meaning of the word then appears on a display screen. Various types of electronic dictionaries are available, including dictionaries of foreign languages.

Among the users of electronic dictionaries are people who retrieve electronic documents from computer networks. The retrieval of such documents has recently been greatly facilitated by the emergence of hypertext. In a hypertext document, certain items (e.g. words) are marked as being linked to other documents, and the user can proceed from one document to another simply by selecting the indicated items with a pointing device. The linking of computer resources throughout the world into a so-called world-wide web, and the commercial availability of software that facilitates browsing through hypertext documents, have brought many users into contact with documents in many languages, and documents on many specialized subjects.

Electronic dictionaries can be an extremely useful aid to the comprehension of such documents, but unfortunately, the ease with which a person can proceed from one document to another is not matched by the ease with which the person can look up an unknown word in a document. Conventional systems that display hypertext documents are not designed for displaying entries in electronic dictionaries, so to find the meaning of a word, the user must access the electronic dictionary in the usual manner, by typing the word separately. When there are many words to be looked up, this can become an irksome and time-consuming process.

The problem could be solved by browsing software that incorporated an electronic dictionary internally, but it is not practical to equip browsing software with all the dictionaries that might be required by all potential users, nor is it reasonable to expect a person who purchases browsing software also to pay for a large number of electronic dictionaries, or even for one electronic dictionary. It is furthermore inefficient for each user to have to store a large number of dictionaries on his or her own computer.

SUMMARY OF THE INVENTION

An object of the present invention is accordingly to provide an easy way for the reader of an electronic document to look up words in that document in an electronic dictionary.

A further object is to provide an easy way for the reader of a hypertext document to look up words in that document in an electronic dictionary.

Another object is to indicate to the user which words can be advantageously be looked up.

The invented document display system displays the contents of an electronic document together with a control item enabling the user to request access to an electronic dictionary. If the user requests dictionary access, the electronic document is modified by attaching dictionary access information to character strings such as individual words in the document, and the modified document is displayed. If the user then selects a character string to which such dictionary access information is attached, an electronic dictionary entry defining or describing the character string is automatically retrieved and displayed. If there is no entry in the electronic dictionary for the selected character string, a message to that effect is displayed.

According to one aspect of the invention, the electronic dictionary is provided with hypertext link information making the entries in the dictionary individually accessible via hypertext links from other documents. In this aspect of the invention, the above-mentioned dictionary access information comprises hypertext links pointing to entries in the electronic dictionary.

According to another aspect of the invention, the electronic dictionary is not provided with such hypertext link information. When attaching the above-mentioned dictionary access information, the document display system also extracts the corresponding entries from the electronic dictionary and stores each extracted entry as a separate document, which can be retrieved via a hypertext link. The dictionary access information comprises hypertext links to these separate documents.

According to yet another aspect of the invention, the dictionary access information comprises dictionary look-up commands. When the user selects a character string to which dictionary access information is attached, the document display system receives and executes a command to look up a corresponding entry in the electronic dictionary. If the corresponding entry is found in the electronic dictionary, it is displayed to the user.

According to a further aspect of the invention, when attaching dictionary access information to an item which already has a hypertext link to another document, the document display system attaches information for generating a menu that allows the user to choose whether to retrieve the other document or look up character strings in the item. If the user selects the item, first the menu is displayed; then, if the user chooses to look up a character string in the item, the dictionary entry of the character string is displayed by one of the methods described above.

According to a still further aspect of the invention, the document display system keeps records indicating how frequently different character strings have been looked up in the past, or how frequently dictionary access information has been attached to different character strings and how frequently the different character strings have been looked up, and decides whether to attach dictionary access information to character strings on the basis of these records.

According to a yet further aspect of the invention, the document display system varies the way in which character strings in the modified electronic document are displayed, according to the above records, to help the user decide which character strings to look up.

BRIEF DESCRIPTION OF THE DRAWINGS

In the attached drawings:

FIG. **1** is a block diagram of a first embodiment of the invented document display system;

FIG. **2** shows part of the contents of the tagged electronic dictionary in the first embodiment;

FIG. **3** is a flowchart illustrating the operation of the client device in the first embodiment;

FIG. **4** is a flowchart illustrating the operation of the linked document server in the first embodiment;

FIG. **5** illustrates the addition of a dictionary mode button to a document;

FIG. **6** illustrates the addition of an ordinary mode button to another document;

FIG. **7** is a flowchart illustrating the operation of the dictionary linker in the first embodiment;

FIG. **8** illustrates part of a result file output by the dictionary linker in the first embodiment;

FIG. **9** illustrates part of a result file output by the dictionary linker when a word cannot be found in the tagged electronic dictionary;

FIG. **10** illustrates an unknown-word entry provided in the tagged electronic dictionary as an alternate method of handling unknown words;

FIG. **11** illustrates an alternative result file output by the dictionary linker when a word cannot be found in the tagged electronic dictionary;

FIG. **12** illustrates an initial document retrieval display;

FIG. **13** illustrates a document retrieved from the display in FIG. **12**;

FIG. **14** illustrates a further document retrieved by a hypertext link from the document in FIG. **13**;

FIG. **15** illustrates the document in FIG. **14** displayed in dictionary-access mode;

FIG. **16** shows an example of information displayed when dictionary access is performed from the display in FIG. **15**;

FIG. **17** shows a display obtained from the result file in FIG. **9**, illustrating one method of handling unknown words;

FIG. **18** shows a message displayed when the alternative method of handling unknown words is employed;

FIG. **19** is a block diagram of a second embodiment of the invented document display system;

FIG. **20** is a flowchart illustrating the operation of the dictionary linker in the second embodiment;

FIG. **21** is a flowchart illustrating the tag attachment step in FIG. **20**;

FIG. **22** is a block diagram of a third embodiment of the invented document display system;

FIG. **23** illustrates part of the contents of the electronic dictionary in FIG. **22**;

FIG. **24** is a flowchart illustrating the operation of the dictionary linker in the third embodiment;

FIG. **25** illustrates part of a result file output by the dictionary linker in the third embodiment;

FIG. **26** is a flowchart illustrating the operation of the dictionary entry extractor in the third embodiment;

FIG. **27** shows an example of information displayed when dictionary access is performed in the third embodiment;

FIG. **28** is a flowchart illustrating the operation of the dictionary entry extractor in a variation of the third embodiment;

FIG. **29** is a flowchart illustrating the operation of the dictionary linker in this variation;

FIG. **30** illustrates part of a result file output by the dictionary linker in this variation;

FIG. **31** is a block diagram of a fourth embodiment of the invented document display system;

FIG. **32** is a flowchart illustrating the operation of the linked document server in the fourth embodiment;

FIG. **33** is a flowchart illustrating the operation of the dictionary linker in the fourth embodiment;

FIG. **34** illustrates part of a result file output by the dictionary linker in the fourth embodiment;

FIG. **35** is a flowchart illustrating the operation of the dictionary entry extractor in the fourth embodiment;

FIG. **36** is a block diagram of a fifth embodiment of the invented document display system;

FIG. **37** is a flowchart illustrating the operation of the dictionary tinker in the fifth embodiment;

FIG. **38** illustrates part of a result file output by the dictionary linker in the fifth embodiment;

FIG. **39** is a flowchart illustrating the operation of the dictionary entry extractor in the fifth embodiment;

FIG. **40** is a block diagram of a sixth embodiment of the invented document display system;

FIG. **41** is a flowchart illustrating the operation of the dictionary entry extractor in the sixth embodiment;

FIG. **42** is a block diagram of a seventh embodiment of the invented document display system;

FIG. **43** is a flowchart illustrating the operation of the linked document server in the seventh embodiment;

FIG. **44** is a flowchart illustrating the operation of the dictionary linker in the seventh embodiment;

FIG. **45** is a flowchart illustrating the operation of the menu generator in the seventh embodiment;

FIG. **46** illustrates part of a document file output by the linked document server in the seventh embodiment;

FIG. **47** shows how this document file is displayed by the client device;

FIG. **48** illustrates part of a result file output by the dictionary linker and linked document server in the seventh embodiment;

FIG. **49** shows how this result file is displayed by the client device;

FIG. **50** shows a menu displayed when the user selects a certain button in the display in FIG. **49**;

FIG. **51** shows the same menu with "message" selected for dictionary access;

FIG. **52** is a block diagram of an eighth embodiment of the invented document display system;

FIG. **53** is a flowchart illustrating the operation of the link remover in the eighth embodiment;

FIG. **54** illustrates a document displayed by the eighth embodiment in the dictionary-access mode;

FIG. **55** is a block diagram of a ninth embodiment of the invented document display system;

FIG. **56** is a flowchart illustrating the operation of the linked document server in the ninth embodiment;

FIG. **57** is a flowchart illustrating the operation of the dictionary linker in the ninth embodiment;

FIG. **58** illustrates part of a table maintained by the dictionary access tabulator in the ninth embodiment;

FIG. **59** is a flowchart illustrating the operation of the dictionary access tabulator in the ninth embodiment;

FIG. **60** is another flowchart illustrating the operation of the dictionary access tabulator in the ninth embodiment;

FIG. **61** illustrates part of a result file output by the dictionary linker **4** and linked document server **2** in the ninth embodiment;

FIG. **62** illustrates the document In FIG. **14** displayed in dictionary-access mode by the ninth embodiment;

FIG. **63** illustrates part of a result file output by the dictionary linker **4** in a variation of the ninth embodiment;

FIG. **64** is a block diagram of a tenth embodiment of the invented document display system;

FIG. **65** is a flowchart illustrating the operation of the dictionary linker in the tenth embodiment; and

FIG. **66** is a Flowchart illustrating the operation of the ratio calculator In the tenth embodiment.

DETAILED DESCRIPTION OF THE INVENTION

Various embodiments of the invented display system will be described below with reference to the attached drawings.

First Embodiment

Referring to FIG. **1**, the first embodiment comprises a client device **1**, a linked document server **2**, an electronic document store **3**, a dictionary linker **4**, and a tagged electronic dictionary **5**, which are interconnected as shown. The client device **1**, linked document server **2**, and electronic document store **3** are facilities provided in conventional document display systems. The dictionary linker **4** and tagged electronic dictionary **5** are novel features of the present invention.

Although only one linked document server **2** is shown in the drawing, there may be a plurality of linked document servers **2** located at different sites, each with its own electronic document store **3**, dictionary linker **4**, and tagged electronic dictionary **5**. Each linked document server **2** is accessible from the client device **1** through a communication channel, which may include part of a public telephone network. There may also be multiple client devices **1** disposed in different locations.

The client device **1** comprises, for example, a personal computer provided with a client program, such as a browsing program, that interacts with the linked document server **2**, thereby enabling the user to enter commands that retrieve arid display documents stored in the electronic document store **3**. The client device **1** also has a pointing device such as a so-called mouse, with which the user can select individual words, character strings, or other items in the displayed documents.

The client device **1** of course does not have to comprise a personal computer. Any device that enables the user to enter commands, display retrieved documents, and select character strings can serve as the client device **1**.

The linked document server **2** comprises, for example, a workstation or computer system equipped with communication functions and software for retrieving electronic document files from the electronic document store **3** and sending these files to the client device **1**, and for other functions that will be described below.

The electronic document store **3** comprises, for example, a magnetic or optical disk drive, which may be physically integrated with the linked document server **2**, and which stores a plurality of electronic documents in the form of computer-accessible files.

The dictionary linker **4** comprises, for example, a program running on the same workstation or computer dictionary linker **4** as the linked document server **2**. The function of the dictionary linker **4** in the first embodiment is to refer to the tagged electronic dictionary **5** and add hypertext links to a file provided by the linked document server **2** to the client device. The dictionary linker **4** is activated by the linked document server **2**.

The tagged electronic dictionary **5** comprises, for example, a file stored on a magnetic or optical disk, or in the memory of the workstation or computer on which the linked document server **2** and dictionary linker **4** operate. The tagged electronic dictionary **5** differs from conventional electronic dictionaries in being provided with tags that can anchor links from hypertext documents. In the following description, it will be assumed that the tagged electronic dictionary **5** is an English-to-Japanese dictionary.

FIG. **2** shows an example of part of the contents of the tagged electronic dictionary **5**. The first line in FIG. **2** is an opening tag **11** that anchors a dictionary entry named "storehouse." The next two lines constitute the heading and body of this entry **12**, the word "storehouse" appearing as the heading, followed by a Japanese definition of the meaning of the word. The next line is a closing tag **13** indicating the end of the dictionary entry. These tags **11** and **13**, incidentally, conform to a well-known hypertext markup language (HTML).

Although the dictionary linker **4** and tagged electronic dictionary **5** may reside in the same workstation or computer system as the linked document server **2** and electronic document store **3**, this is not necessary. The dictionary linker **4** can reside in a separate computer system or a specialized system coupled to the system in which the linked document server **2** and electronic document store **3** reside. The dictionary linker **4** and tagged electronic dictionary **5** may also reside in an independent system that is coupled by telecommunication lines to the linked document server **2** and to other linked document servers, making the facilities of the dictionary linker **4** and tagged electronic dictionary **5** available to a plurality of server systems.

Next, the operation of the first embodiment will be described. In the description, the term "electronic document" wilt sometimes be shortened to "document." The term "hypertext document" will refer to an electronic document having embedded hypertext links pointing either to the tagged electronic dictionary **5** or to another document. The tagged electronic dictionary **5** will also be referred to as a hypertext document. Except for the tagged electronic dictionary **5**, documents that do not contain embedded links pointing to other documents will not be described as hypertext documents, even though these documents may be the targets of links from hypertext documents.

First, the operation of the client device **1** will be described. Referring to FIG. **3**, when the user accesses the document display system, the client device **1** first displays an initial input screen (step **100**), then waits for input from the user (step **101**). The input may be a file descriptor requesting an electronic document, or a command such as "Quit." If the "Quit" command is given, access to the document display system is terminated (step **102**). File descriptors and commands other than "Quit" generally cause the client device **1** to send information to the linked document server (step **103**). The linked document server **2** generally responds by sending back an electronic document, which the client device **1** receives and displays (step **104**). After step **104**, the process returns to step **101** to wait for further input: for example, for the user to request a further document linked to the document now on display.

Although the user may give various commands other than "Quit," it will be assumed in the first embodiment that the information sent to the linked document server **2** is always either a file descriptor or one particular command, specifically, a tag attachment command. Details will be given later.

Next, the operation of the linked document server **2** will be described. The linked document server **2** repeatedly executes the process shown in FIG. **4**. The process starts with the reception of information from the client device **1**

(step **200**). The linked document server **2** identifies the information as a file descriptor or tag attachment command (step **201**). If the information is a file descriptor, the linked document server **2** reads the requested document from the electronic document store **3** (step **202**). If the information is a tag attachment command, the linked document server **2** activates the dictionary linker **4** (step **203**), at the same time providing the dictionary linker **4** with a copy of the document currently being displayed by the client device **1**.

If the information received in step **200** is a file descriptor, after obtaining the document in step **202**, the linked document server **2** decides whether the document is a hypertext document (step **204**). If the document is not a hypertext document, the linked document server **2** adds a dictionary-mode control item, referred to below as a dictionary mode button, to the document (step **205**), then sends the document to the client device **1** (step **206**). If the document is a hypertext document, the linked document server **2** sends the document to the client device **1** (step **206**) without adding a dictionary mode button.

If the information received in step **200** is a tag attachment command, the dictionary linker **4** activated in step **203** modifies the provided document by adding dictionary access information. The dictionary access information comprises tags which will be described later. The linked document server **2** then adds an ordinary-mode control item, referred to below as an ordinary mode button, to the resulting modified document (step **207**), and sends the document to the client device **1** (step **206**).

The dictionary mode button and ordinary mode button are items that the user can select with the pointing device to switch between a dictionary-access mode and an ordinary mode, at the user's own convenience. These control items can be added to the document in various ways, one of which is illustrated in FIG. **5**, which shows part of a document **21** that begins with the words "We draw on vast storehouses . . . . " To add a dictionary mode button to this document, the linked document server **2** adds a line **21** containing the command "/cgi-bin/into_the_dic" in a hypertext reference (HREF) tag, followed by the words "Dictionary mode," then a closing tag (</A>). The user sees the words "Dictionary mode." When the user selects these words, the client device **1** sends the character string "/cgi-bin/into_the_dic" to the linked document server **2**, "/cgi-bin/into_the_dic" being the tag attachment command mentioned above.

FIG. **6** illustrates the similar addition of an ordinary mode button to a modified document output by the dictionary linker **4**. Although the body **31** of the document has much embedded dictionary access information, the body **31** is displayed in nearly the same way as the body of the document in FIG. **5**, as will be shown later. The linked document server **2** adds a line **32** in which the hypertext reference "slogan" is the file descriptor of the document in FIG. **5**, which was on display at the elient device **1** when the dictionary linker **4** was activated. The user can select the words "Ordinary mode" in line **32** with the pointing device, causing the client device **1** to send the file descriptor "slogan" to the linked document server **2**.

Next, the operation of the dictionary linker **4** will be described. Referring to FIG. **7**, the dictionary linker **4** begins by receiving from the linked document server **3** a copy of the document currently on display at the client device **1**, or (equivalently) receives permission to read this document from the electronic document store **3** (step **500**). Next, the dictionary linker **4** creates a result file (step **501**). The result file is created as a new file, and is initially empty.

The dictionary linker **4** now performs a morphemic analysis of the document obtained in step **500**. By means of this analysis, the dictionary linker **4** identifies the words appearing in the document, and the dictionary forms of these words. If the word "storehouses" appears, for example, the dictionary linker **4** identifies "storehouse" as the dictionary form.

The subsequent steps from step **503** to step **507** form a loop that is repeated as long as any unprocessed word remains in the document, each repetition adding one line to the result file. Step **503** is a loop control step in which the dictionary linker **4** determines whether any unprocessed word remains. When no unprocessed word remains, the result file is output to the linked document server **2** as the modified document mentioned above (step **508**).

When an unprocessed word is found in step **503**, the dictionary linker **4** looks for a definition of the word (i.e. for an entry headed by the dictionary form of the word) in the tagged electronic dictionary **5** (step **504**). If the definition is found, the dictionary linker **4** generates dictionary access information comprising tags establishing a link from the word to the dictionary entry defining the word (step **505**), writes these tags and the word itself in the result file (step **506**), and returns to step **503** to process the next word. If the word is not defined in the tagged electronic dictionary **5**, the dictionary linker **4** carries out appropriate unknown-word processing (step **506**), two examples of which will be given later. The dictionary linker **4** then writes the unknown word in the result file, together with any tags that may have been generated by the unknown-word processing (step **506**), and returns to step **503**.

FIG. **8** shows the first five lines of the result file output by the dictionary linker **4** for a document beginning with "We draw on vast storehouses . . . " when all five of these words are defined in the tagged electronic dictionary **5**. Each line begins with an opening tag such as <A HREF= "tagED#we">, which is a hypertext reference to the entry for the word "we" in the tagged electronic dictionary **5**. This is followed by the word as it appeared in the document (e.g. "We"), then a closing tag (</A>).

FIG. **9** illustrates one possible type of unknown-word processing, taking as an example the case in which the word "vast" is not defined in the tagged electronic dictionary **5**. The unknown-word processing in this example consists of doing nothing at all; the dictionary linker **4** writes "vast" in the result file without any opening or closing tags. If this type of unknown-word processing is adopted, step **507** in FIG. **7** is a no-operation step, and can be omitted.

FIGS. **10** and **11** show an example of another possible type of unknown-word processing. Tn this example, the tagged electronic dictionary **5** has a special entry for unknown words, shown in FIG. **10**. The body of this entry is a Japanese sentence, read as "Sono tango wa jisho ni arimasen," meaning "That word is not in the dictionary." When the dictionary linker **4** cannot find a word (e.g. "vast") in the tagged electronic dictionary **5**, tags linking that word to this special unknown-word entry are generated and written in the result file around the unknown word, as illustrated in the fourth line in FIG. **11**.

The result file created by the dictionary linker **4** is a temporary file. After the result file has been output to the linked document server **2** and transferred to the client device **1**, the linked document server **2** may delete this file from the memory of the computer system or workstation in which the linked document server **2** resides, so that a copy of the result file remains only at the client device **1**. Similarly, if the

linked document server **2** and dictionary linker **4** reside in different systems, the dictionary linker **4** may delete the result file from the memory of its own system after the result file has been transferred to the system of the linked document server **2**.

Next, the overall operation of the first embodiment will be described, with reference to FIGS. **12** to **18**. Three cases of dictionary access will be illustrated: one in which there are no unknown words; another in which an unknown word is processed as in FIG. **9**; and another in which an unknown word is processed as in FIGS. **10** and **11**.

When first activated by the user, the client device **1** sends certain initial information to the linked document server **2**. The linked document server **2** responds by sending back the contents of an initial screen like the one in FIG. **12**, which the client device **1** displays. At the top of this screen are four buttons marked "Back," "Forward," "Reload," and "Quit." By selecting the "Quit" button with a pointing device, the user enters the "Quit" command described above. When the "Reload" button is selected, the client device **1** sends the linked document server **2** the file descriptor of the document currently on display, causing the linked document server **2** to send the document again. The "Forward" and "Back" buttons move forward and backward in a series of documents that the user has accessed by means of hypertext links. These four buttons are displayed at the top of all screens in the first embodiment.

Below these buttons, the initial screen has a line on which the user can enter the file descriptor of a desired document. The client device **1** sends the entered file descriptor to the linked document server **2**, which sends back the requested document file.

In FIG. **13**, the user has entered the file descriptor of a hypertext document entitled "Corporate Guidance," which the linked document server **2** has retrieved from the electronic document store **3** and sent to the client device **1**, and which is now displayed on the screen of the client device **1**. The three lines below the title are hypertext links, and are marked as such with underlines. Since this document is a hypertext document, no dictionary mode button is attached or displayed.

If the user selects, for example, the "Global Slogan" line with the pointing device, the client device **1** sends the linked document server **2** the file descriptor of a further document. This file descriptor is contained in a hypertext reference tag that precedes the "Global Slogan" line but is not visible to the user. The linked document server **2** retrieves the specified further document from the electronic document store **3** and sends the document back to the client device **1**. The client device **1** displays this document as shown in FIG. **14**. This document has no links to further documents, so it is not a hypertext document in the sense in which the term is being used herein, and the linked document server **2** attaches a dictionary mode button.

Incidentally, the user could also retrieve the "Global Slogan" document directly, by entering the file descriptor of this document on the initial screen in FIG. **12**, instead of first retrieving the "Corporate Guidance" document.

The operations so far have been carried out in the ordinary mode. If the user selects the dictionary mode button with the pointing device, the client device **1** sends the linked document server **2** the tag attachment command ("/cgi-bin/into_the_dic") contained in the tag preceding the words "Dictionary mode." The linked document server **2** executes this command, thereby activating the dictionary linker **4**.

The dictionary linker **4** has the linked document server **2** retrieve the "Global Slogan" document from the electronic document store **3** again, looks up the words in this document in the tagged electronic dictionary **5**, and attaches tags to create a result file as was partially shown, for example, in FIG. **8**. When the dictionary linker **4** has looked up all words and completed the result file, the linked document server **2** adds an ordinary mode button to the result file, and sends the resulting modified document to the client device **1**. The tag preceding the ordinary mode button contains the file descriptor of the "Global Slogan" document, shown as the word "slogan" in the first line **32** of FIG. **6**.

If all words in the "Global Slogan" document are defined in the tagged electronic dictionary **5**, the user now sees the display shown in FIG. **15**. Each word in the document is underlined, the underlines indicating the presence of hypertext links to the tagged electronic dictionary **5** that were added by the dictionary linker **4**.

To find the Japanese meaning of, for example, the word "storehouses," the user selects this word with the pointing device. The client device **1** sends the information "tagED#storehouse" contained in the tag attached to this word to the linked document server **2**. The linked document server **2** accesses the tagged electronic dictionary **5** at the entry indicated by this information, and sends that entry, together with a certain number of preceding and following entries, back to the client device **1**. The client device **1** then displays the screen depicted in FIG. **16**, which gives Japanese definitions for "storehouse" and other words. If he wants to, the user can scroll this screen up or down to display the Japanese meanings of other words in alphabetical sequence with "storehouse."

To return from this display of dictionary meanings to the modified document display, the user selects the "Back" button in FIG. **16**. The client device **1** then displays the screen shown in FIG. **15** again, enabling the user to took up other words in the same way that "storehouses" was looked up, by selecting the desired word with the pointing device.

When the user has finished looking up words, he can select the ordinary mode button in FIG. **15** to have the "Global Slogan" document displayed again in the ordinary mode, as shown in FIG. **14**. If the client device **1** retains a copy of this document, the client device **1** displays the retained copy; otherwise, the client device **1** sends the file descriptor of the "Global Slogan" document to the linked document server **2** again, receives the unmodified "Global Slogan" document again from the linked document server **2**, and displays the received document.

Next, examples in which unknown words appear in the document in FIG. **15** will be shown.

If the word "vast" is not defined in the tagged electronic dictionary **5**, and the dictionary linker **4** adopts the method shown in FIG. **9** of dealing with unknown words, then when the user selects the dictionary-access mode, instead of the display in FIG. **15**, he sees the display in FIG. **17**, in which "vast" is not underlined. If the user selects "vast" with the pointing device, nothing happens, because this word has no dictionary access information attached. This method of processing unknown words has the advantage of indicating to the user which words are and which words are not defined in the tagged electronic dictionary **5**, so that the user need riot waste time in attempting to look up words that are not defined.

If the method illustrated in FIGS. **10** and **11** of dealing with unknown words is adopted, then even if "vast" is not defined in the tagged electronic dictionary **5**, after selecting the dictionary mode button, the user sees the display shown FIG. **15**. If the user tries to look up the word "vast," the client

device **1** sends the information "tagED#UNKNOWN WORDS" to the linked document server **2**, which accordingly accesses the special unknown-word entry in the tagged electronic dictionary **5**. As a result, the user sees the screen displayed in FIG. **18**, with a Japanese message meaning "That word is not in the dictionary." This method has the advantage of explaining to the user why the word cannot be looked up. The user can use the "Back" button to return from this screen to the display in FIG. **15**.

When a user follows a series of hypertext links to a document written in a foreign language, the first embodiment assists the user in reading the document in a natural and intuitive way: to look up words, the user points to the words "Dictionary mode," then simply points to the words to be looked up. The user does riot have to learn any special operations or do any extra typing.

The first embodiment is efficient in that it displays only the definitions the user wants to see, and does not attempt to display the meaning of every word in a document, or translate the entire document. The amount of information transferred between the client device **1** and linked document server **2** can thus be held to a minimum, and needless delays can be avoided. If the tagged electronic dictionary **5** is stored in a semiconductor memory, or is loaded from a disk memory into a semiconductor memory when the linked document server **2** is started up, the tagged electronic dictionary **5** can be accessed by means of internal memory pointers, enabling the user to obtain definitions very quickly.

The first embodiment is also efficient in that it enables dictionary definitions to be accessed without requiring tags with links to dictionary entries to be embedded in the documents stored in the electronic document store **3**. Thus users who do not need to use the dictionary will not be distracted by unnecessary underlines in the documents they are reading.

Since the tagged electronic dictionary **5** is located at the site of the linked document server **2**, it is not necessary for each user to purchase his own copy of the tagged electronic dictionary **5**. Moreover, this site can be provided with a large number of tagged electronic dictionaries of different types, in different languages, for example, and the system can be adapted to provide the user with a choice of dictionaries.

In this case a third type of unknown-word processing is available. If a word is not found in one dictionary, the dictionary linker **4** can link the word to a menu screen offering the user a selection of other dictionaries in which the user might wish to try to look up the word, this screen being provided with links to commands that access the other dictionaries. This menu screen can be built into the tagged electronic dictionary **5**, so that both known words and unknown words are initially processed in the same way: by displaying a page from the tagged electronic dictionary **5**. The system can then be expanded by adding dictionary-access commands, without having to modify the basic operations of the linked document server **2** and dictionary linker **4**.

Second Embodiment

The second embodiment differs from the first embodiment in that the result files generated by the dictionary linker **4** are saved for possible future use.

Referring to FIG. **19**, the second embodiment has the same client device **1**, linked document server **2**, electronic document store **3**, dictionary linker **4**, and tagged electronic dictionary **5** as the first embodiment, arid an additional dictionary-access-ready document store **6**, which stores the result files generated by the dictionary linker **4**. The dictionary-access-ready document store **6** is, for example, a magnetic or optical memory device coupled to the linked document server **2** and dictionary linker **4**. This device may be an independent device such as an external disk drive. Alternatively, the (dictionary-access-ready document store **6** may be incorporated into the workstation or computer system in which the linked document server **2** resides, or the system in which the dictionary linker **4** resides, or the system in which both the linked document server **2** and dictionary linker **4** reside.

Next, the operation of the second embodiment will be described.

The linked document server **2** operates as in the first embodiment, following the flowchart in FIG. **4**, except that under certain conditions, the modified document (result file) to which an ordinary mode button is added in step **207** is obtained from the dictionary-access-ready document store **6** instead of from the dictionary linker **4**.

The operation of the dictionary linker **4** differs from the first embodiment, and is illustrated in FIGS. **20** and **21**. Referring to FIG. **20**, when activated, the FIG. **4** begins by obtaining a copy of the unmodified document currently on display at the client device **1** (step **300**). This step is the same as step **500** in FIG. **7**.

Next, the dictionary linker **4** searches the directory of the dictionary-access-ready document store **6** to see if the dictionary-access-ready document store **6** already contains a result file for this document (step **301**). If it does, the dictionary linker **4** compares the time stamp on the unmodified document with the time stamp on the result file to determine whether the document stored in the electronic document store **3** has been updated since the result file in the dictionary-access-ready document store **6** was created (step **302**).

If the document in the electronic document store **3** has not been updated since the result file in the dictionary-access-ready document store **6** was created, the dictionary linker **4** checks whether the result file in the dictionary-access-ready document store **6** is locked (step **303**). The locked state occurs if the result file is currently being tagged in response to a request from a different client device. The check in step **303** is repeated until the document is found not to be locked, at which time the dictionary linker **4** notifies the linked document server **2** (step **304**), and the linked document server **2** transfers the result file from the dictionary-access-ready document store **6** to the client device **1**.

If there is no corresponding result file in the dictionary-access-ready document store **6**, giving a negative result in step **301**, or if the result file is present in the dictionary-access-ready document store **6** but the original document in the electronic document store **3** has been updated, giving an affirmative result in step **302**, the dictionary linker **4** creates a new result file with dictionary access information (step **305**), then leaves this new result file in the dictionary-access-ready document store **6** and notifies the linked document server **2** that the result file is ready (step **306**). If an old result file for the same document was present in the dictionary-access-ready document store **6**, the new result file replaces the old result file.

FIG. **21** illustrates the result file creation step **305** in FIG. **20**. The dictionary linker **4** begins by creating and locking the new result file (step **508**). The locking and unlocking of the result file can be carried out by, for example, manipulating an access permission bit in the directory information of the result file. When the new result file is created, the old

result file, if present, is deleted. The same steps **502**, **503**, **504**, **505**, **506**, and **507** as described in the first embodiment (FIG. **7**) are then carried out to write words and tags into the result file. When no unprocessed words remain, giving a negative result in step **503**, the dictionary linker **4** unlocks the result file (step **509**), making it available to the linked document server **2**.

From the user's point of view, the operation of the second embodiment is identical to the operation of the first embodiment, except that dictionary access is often faster. More precisely, when dictionary access from the same document is requested repeatedly, either by the same user or by different users, the response to the request is speeded up the second and subsequent times, because the result file is already available in the dictionary-access-ready document store **6**.

From the system's point of view, additional file storage space is required for the dictionary-access-ready document store **6**, but the processing load is reduced, because as long as a document is not updated, a result file is created for that document only once. This advantage becomes particularly significant if words in the document are likely to be looked up by a large number of users.

Third Embodiment

The third embodiment differs from the preceding embodiments in regard to the format of the electronic dictionary.

Referring to FIG. **22**, the third embodiment has the same client device **1**, linked document server **2**, and electronic document store **3** as the first embodiment, a generally similar dictionary linker **4**, and a dictionary entry extractor **8**, a dictionary entry store **9**, and an electronic dictionary **10**.

The electronic dictionary **10** is an English-to-Japanese dictionary similar to the tagged electronic dictionary **5** of the first embodiment, but has no embedded anchoring tags. A commercially available electronic dictionary can be used as the electronic dictionary **10** FIG. **23** shows an example of part of the electronic dictionary **10**, in which each entry comprises an English word, then a Japanese definition. The Japanese definition is terminated by a special code represented in the drawing by a square, followed by a new-line code, then the next entry. The special code indicates that the word on the next line is the heading of a new dictionary entry, and the information on the subsequent line or lines, up to the next special code, is the definition of the word given in the heading.

The electronic dictionary **10** is not limited to the format shown in FIG. **23**. electronic dictionaries in other formats can be used, as long as the format enables entry headings and definitions to be recognized.

The dictionary entry extractor **8** receives a word from the dictionary linker **4**, looks this word up in the electronic dictionary **10**, stores the entry of this word, comprising the word and its definition, as a separate document in the dictionary entry store **9**, and provides the dictionary linker **4** with information giving the storage location of the entry in the dictionary entry store **9**. For example, the dictionary entry extractor **8** can store each retrieved entry in a separate file in the dictionary entry store **9**, and provide the dictionary entry store **9** with the file names. The contents of the dictionary entry store **9** can be read by the linked document server **2**. Files stored in the dictionary entry store **9** are treated as hypertext documents, so the linked document server **2** does not add a dictionary mode button.

The electronic dictionary **10** can be stored as a single file on the same type of storage device as the electronic document store **3**. The dictionary linker **4**, dictionary entry extractor **8**, dictionary entry store **9**, and electronic dictionary **10** can all be incorporated into the same computer system or workstation as the linked document server **2** and electronic document store **3**. Alternatively, the dictionary linker **4**, dictionary entry extractor **8**, dictionary entry store **9**, and electronic dictionary **10** can reside in another computer system, workstation, or special device coupled to the system or workstation in which the linked document server **2** and electronic document store **3** reside, or linked to that system or workstation by a telecommunication line.

The client device **1** and linked document server **2** operate as in the first embodiment, following the flowcharts in FIGS. **3** and **4**. Upon retrieving a document that is not a hypertext document from the electronic document store **3**, the linked document server **2** adds a dictionary mode button as shown in FIG. **5**.

The dictionary linker **4** now operates according to the flowchart in FIG. **24**. Steps **500**, **501**, **502**, and **503** are the same as in the first embodiment, but when an unprocessed word remains in step **503**, the dictionary linker **4** commands the dictionary entry extractor **8** to process the word (step **510**). This process normally results in the storage of a dictionary entry for the word in the dictionary entry store **9**. The dictionary linker **4** then generates hypertext tags linking the word to the dictionary entry in the dictionary entry store **9** (step **511**), and writes the word and these tags in the result file (step **506**). When no more unprocessed words remain, the result file is output (step **508**) as in the first embodiment.

FIG. **25** shows part of the result file output by the dictionary linker **4** for the document in FIG. **14**. This result file is similar to the one in FIG. **6**, except that the hypertext references are the names of files in the dictionary entry store **9**, such as "/dic/keep/storehouse" in the last line in FIG. **25**. In this reference, "/dic/keep/" is the name of the directory or the dictionary entry store **9**, and "storehouse" is the name of a file in which the dictionary entry for the word "storehouse" has been stored.

FIG. **26** illustrates the operation of the dictionary entry extractor **8**. When given a word to process, the dictionary entry extractor **8** first determines if the entry for this word is already stored in the dictionary entry store **9** (step **400**). If the entry is not already stored, the dictionary entry extractor **8** attempts to look the word up in the electronic dictionary **10** (step **401**). If the word is defined in the electronic dictionary **10**, the dictionary entry extractor **8** reads the entry headed by the word from the electronic dictionary **10**, and stores the entry in a file in the dictionary entry store **9** (step **402**). The file name of this file is the word looked up, and the file contents are the entry read from the electronic dictionary **10**. The dictionary entry extractor **8** then passes the file name to the dictionary linker **4** (step **403**). This completes the processing of the word.

If the word is not defined in the dictionary, giving a negative result in step **401**, the dictionary entry extractor **8** checks whether an unknown-word file is present in the dictionary entry store **9** (step **404**). The unknown-word file has a predetermined name, such as "unknown_words," for example. Step **404** is carried out by searching for this file name in the directory "/dic/keep/."

If no unknown-word file is present in the dictionary entry store **9**, the dictionary entry extractor **8** now creates one (step **405**), giving the unknown-word file the above-mentioned predetermined name. The file contents are, for example, a Japanese message stating that the word is riot in the dictionary: more specifically, the message illustrated in FIG. **18**.

The dictionary entry extractor **8** then passes the name of the unknown-word file to the dictionary linker **4** in step **403** to complete the processing of the unknown word.

If an unknown-word file already exists, giving an affirmative result in step **404**, the dictionary entry extractor **8** passes the name of this file to the dictionary entry store **9** (step **403**) without creating a new unknown-word file. If the entry of the word is already stored in the dictionary entry store **9**, giving an affirmative result in step **400**, the dictionary entry extractor **8** proceeds immediately to step **403** and passes the file name of the entry to the dictionary linker **4**, skipping the intermediate steps **401** and **402**.

The overall operation of the third embodiment will be described next, focusing on the operations that occur when the user retrieves the "Global Slogan" document shown in FIG. **14** and selects the dictionary mode button on this document with the pointing device. Other operations are carried out as described in the first embodiment. It will be assumed that the dictionary entry store **9** is initially empty. It will also be assumed that the electronic dictionary **10** contains entries for all words in the "Global Slogan" document except the word "vast."

As in the first embodiment, selection of the dictionary mode button causes the client device **1** to send the command "/cgi-bin/into_the_dic" to the linked document server **2**, and the linked document server **2** to activate the dictionary linker **4**. After obtaining a copy of the "Global Slogan" document from the linked document server **2**, creating a new result file, and performing a morphemic analysis, the dictionary linker **4** passes the first word ("we") of this document to the dictionary entry extractor **8**.

The dictionary entry extractor **8** checks to see whether an entry for "we" is already stored in the dictionary entry store **9**. Specifically, the dictionary linker **4** searches for a file named "we" in the directory "/dic/keep/" of the dictionary entry store **9**. By the assumption above, no such file exists, so the dictionary linker **4** next looks up the word "we" in the electronic dictionary **10**, reads the entire entry for this word, stores the entry as a new file named "/dic/keep/we" in the dictionary entry store **9**, and passes the file descriptor "/dic/keep/we" to the dictionary linker **4**. As the first line in the result file, the dictionary linker **4** writes the line <A HREF="/dic/keep/we">We</A>.

The next two words, "draw" and "on," are processed similarly, creating files named "/dic/keep/draw" and "dic/keep/on" in the dictionary entry store **9**. The word "vast," however, is not defined in the electronic dictionary **10**, so the dictionary entry extractor **8** checks to see whether an unknown-word file is already present in the dictionary entry store **9**. By the assumption above, no unknown-word file is present, so the dictionary entry extractor **8** creates a file named "dic/keep/unknown_words" containing the message shown in FIG. **18**, and passes the file name to the dictionary linker **4**. The dictionary linker **4** writes the line <A HREF= "/dic/keep/unknown_words">vast</A> in the result file.

Further words in the "Global Slogan" document are processed in the same way. The word "from" is encountered twice. The first time, the dictionary entry extractor **8** reads the entry for this word from the electronic dictionary **10**, creates a new file named "/dic/keep/from" in the dictionary entry store **9**, and passes the file name "/dic/keep/from" to the dictionary linker **4**. The second time, the dictionary entry extractor **8** only passes the file name "/dic/keep/from" to the dictionary linker **4**, without creating a new file. Each time, the dictionary linker **4** writes a new <A HREF="/dic/keep/from">from</A> line in the result file.

When all words have been processed, the result file is transferred from the dictionary linker **4** to the linked document server **2**, then to the client device **1**, and the user sees the screen shown in FIG. **15**. If the user selects the word "storehouses" with the pointing device, the client device **1** sends the file descriptor "/dic/keep/storehouse" contained in the invisible hypertext tag preceding this word to the linked document server **2**. The linked document server **2** retrieves the file named "/dic/keep/storehouse" from the dictionary entry store **9**, and sends this file to the client device **1**. The client device **1** then displays the screen in FIG. **27**, showing the Japanese meanings or "storehouse."

If the user selects the "Back" button on the screen in FIG. **27**, the client device **1** again displays the screen shown in FIG. **15**. If the user next selects the word "vast," the client device **1** sends the file descriptor "/dic/keep/unknown_words" to the linked document server **2**. The linked document server **2** retrieves the file named "/dic/keep/unknown_words" from the dictionary entry store **9**, and sends this file to the client device **1**, which now displays the screen shown in FIG. **18**, informing the user that the selected word is not in the dictionary.

To the user, the third embodiment appears to operate like the first embodiment, except that when the user looks up words, only one definition appears on the screen at a time. One advantage of this is that less information has to be transferred from the linked document server **2** to the client device **1**. Another advantage is that the user is shown only what he want to see, and can quickly read the desired definition without being distracted by definitions of other words. The first embodiment could also be adapted to operate in this way.

In terms of speed and efficiency, the third embodiment is generally intermediate between the first and second embodiments. When the dictionary-access mode is selected repeatedly for the same document, the result file is returned to the client device **1** faster than in the first embodiment, because no actual dictionary look-up is necessary the second time and subsequent times, but not as quickly as in the linked document server **2**, because the dictionary linker **4** still has to create a result file each time.

If the dictionary-access mode is selected for a series of different documents, however, the third embodiment may outperform the second embodiment in the second and subsequent documents, because common words will already be stored in the dictionary entry store **9**. The electronic document store **3** may also improve on the second embodiment in terms of memory efficiency, because the size of the dictionary entry store **9** in the third embodiment is limited by the size of the electronic dictionary **10**, whereas the dictionary-access-ready document store **6** in the second embodiment can grow very large if result files for many documents are stored.

The main advantage of the third embodiment, however, is that the electronic dictionary **10** does not, have to have embedded tags. Commercially available electronic dictionaries can be used in their existing form, greatly increasing the number of dictionaries that can be accessed.

FIGS. **28** to **30** illustrate a variation of the third embodiment that does not create an unknown-word file in the dictionary entry store **9**.

Referring to FIG. **28**, upon encountering an unknown word in step **401**, the dictionary entry extractor **8** sets an unknown-word flag (step **406**), then terminates processing without creating any file in the dictionary entry store **9**. Step **406** replaces steps **404** and **405** in FIG. **26**. Steps **400**, **401**, **402**, and **403** are the same as in FIG. **26**.

Referring to FIG. **29**, after having a word processed by the dictionary entry extractor **8** in step **510**, the dictionary linker **4** checks the unknown-word flag (step **512**). If the unknown-word flag is set, the dictionary linker **4** skips step **511**, and writes the word in the result file without attached tags. If the unknown-word flag is not set, the dictionary linker **4** executes both steps **511** and **506**, as in FIG. **24**. The other steps in FIG. **29** are identical to the corresponding steps in FIG. **24**. The unknown-word flag is cleared each time the dictionary linker **4** activates the dictionary entry extractor **8**, although this is riot explicitly shown in FIGS. **28** and **29**.

FIG. **30** shows the result file output by the dictionary linker **4** in this variation of the third embodiment when the dictionary mode button is selected in FIG. **14** and the word "vast" does not appear in the electronic dictionary **10**. The first three lines and the fifth line in this result file are the same as in FIG. **25**, but the word "vast" appears by itself on the fourth line.

When this result file is transferred to the client device **1**, the user sees the display shown in FIG. **17**, in which the word "vast" is riot underlined. As explained in the first embodiment, this variation has the advantage of warning the user in advance that the word cannot be looked up.

Fourth Embodiment

Referring to FIG. **31**, the fourth embodiment has the same client device **1**, electronic document store **3**, and electronic dictionary **10** as the third embodiment, and a generally similar linked document server **2**, dictionary linker **4**, and dictionary entry extractor **8**, but has no dictionary entry store. The dictionary entry extractor **8** communicates directly with the linked document server **2**, and does not communicate with the dictionary linker **4**.

Referring to FIG. **32**, the linked document server **2** in the fourth embodiment can receive two types of commands: tag attachment commands and dictionary look-up commands. The command processing is modified accordingly. After receiving information from the client device **1** in step **200**, the linked document server **2** first determines whether the received information is a command (step **208**). If the received information is not a command, the linked document server **2** proceeds with the processing already described in the first embodiment (steps **202**, **204**, **205**, and **206**). If the received information is a command, the linked document server **2** proceeds to step **201** to decide whether the command is a tag attachment command.

If the command is a tag attachment command, the linked document server **2** proceeds as described in the first embodiment (steps **203** and **207**). If the command is not a tag attachment command, then the command is a dictionary look-up command, which the linked document server **2** processes by activating the dictionary entry extractor **8** (step **209**).

Referring to FIG. **33**, upon being activated by the linked document server **2**, the dictionary linker **4** obtains a copy of the relevant document file, creates a result file, and performs a morphemic analysis as in the first and third embodiments (steps **500**, **501**, and **502**). For each unprocessed word found in step **503**, however, the dictionary linker **4** now attaches a dictionary look-up command tag and a closing tag (step **514**), in place of the hypertext reference tags that were attached in the preceding embodiments. The word and attached dictionary look-up command tag and closing tag are then written in the result file (step **506**). This process continues until dictionary look-up command tags have been attached to all words in the document, at which point the result file is output to the linked document server **2** (step **508**) and the processing of the dictionary linker **4** ends.

FIG. **34** shows an example of the result file output by the dictionary linker **4** in the fourth embodiment for the "Global Slogan" document in FIG. **14**. Each dictionary look-up command begins with "/cgi-bin/pic_dic?" The question mark is followed by the word to be looked up, e.g. "we" in the first line in FIG. **34**. When the command is executed, the dictionary entry extractor **8** is activated as a command processor, and the word following the question mark is passed to the dictionary entry extractor **8** as a parameter.

FIG. **35** shows the processing carried out by the dictionary entry extractor **8**. Upon activation, the dictionary entry extractor **8** looks up the supplied word in the electronic dictionary **10** (step **401**). If this word is defined in the electronic dictionary **10**, the dictionary entry extractor **8** extracts the entire dictionary entry for the word (step **408**), then outputs the entry to the linked document server **2**. If the word is not defined in the dictionary, the dictionary entry extractor **8** prepares an unknown-word message (step **410**), and outputs this message to the linked document server **2** (step **409**). The content of the unknown-word message is, for example, a Japanese sentence stating that the particular word is not found in the dictionary. In the course of this processing, the dictionary entry extractor **8** may create a temporary file to hold the dictionary entry or unknown-word message.

Next, the overall operation of the fourth embodiment will be described, again focusing on the operations that take place when the dictionary mode button is selected in FIG. **14**.

Selection of the dictionary mode button sends the same tag attachment command to the linked document server **2** as in the preceding embodiments. The linked document server **2** activates the dictionary linker **4**, which quickly generates a result file by attaching dictionary look-up command tags to all of the words in the document, as illustrated in FIG. **34**. This result file is displayed at the client device **1** as shown in FIG. **15**, all words being underlined to indicate the presence of hypertext links.

If the user now selects the word "storehouses," for example, the client device **1** sends the linked document server **2** the attached dictionary look-up) command "/cgi-bin/pic_dic?storehouse." The linked document server **2** executes this command by activating the dictionary entry extractor **8**, passing the word "storehouse" to the dictionary entry extractor **8** as a parameter. The dictionary entry extractor **8** looks up the word "storehouse" in the electronic dictionary **10** and returns the dictionary entry for this word, which is transferred to the client device **1** and displayed as in FIG. **27**.

If the user selects a word such as "vast" which is not defined in the electronic dictionary **10**, the dictionary entry extractor **8** creates a message such as "vast wa jisho ni arimasen," in which "vast" is the undefined word, and "wa jisho ni arimasen" are Japanese words meaning "is not in the dictionary." This message is returned to the linked document server **2**, transferred to the client device **1**, and displayed.

To the user, the fourth embodiment appears to operate much like the third embodiment, with the slight difference that the unknown-word message names the word that could not be found in the dictionary. In terms of speed, the display in FIG. **15** is generated more quickly than in the third embodiment, because no dictionary look-up is required. When the user selects a word in this display, however, the definition (or unknown-word message) is returned more

slowly than in the third embodiment, because the dictionary entry extractor **8** must search for the word in the electronic dictionary **10**. Thus the third embodiment is advantageous for users who look up a large number of words in a short document, while the fourth embodiment is advantageous for users who look up fewer words in a longer document.

From the system's point of view, the fourth embodiment has the advantage of requiring less memory, because there is no dictionary entry store, and the further advantage that no time is spent in extracting words that the user will not select for dictionary look-up. When the same word is looked up repeatedly, however, the dictionary entry extractor **8** must be activated each time, instead of only once as in the third embodiment.

Fifth Embodiment

The fifth embodiment combines the advantages of the third and fourth embodiments.

Referring to FIG. **36**, the fifth embodiment comprises the same client device **1**, electronic document store **3**, and electronic dictionary **10** as the third embodiment, and a generally similar linked document server **2**, dictionary linker **4**, dictionary entry extractor **8**, and dictionary entry store **9**. The interrelations among these elements differ from the third embodiment in that the dictionary entry extractor **8** communicates with the linked document server **2** instead of with the dictionary linker **4**, and the dictionary linker **4** can access the dictionary entry store **9**.

As in all of the preceding embodiments, the linked document server **2** receives information from the client device **1** and sends back documents retrieved from the electronic document store **3**. As in the third embodiment, the linked document server **2** can also obtain result files from the dictionary linker **4** and dictionary entries from the dictionary entry store **9**, and send these to the client device **1**. As in the fourth embodiment, the linked document server **2** can activate both the dictionary linker **4** and dictionary entry extractor **8** in response to commands received from the client device **1**, and can obtain dictionary entries from the dictionary entry extractor **8**. These operations of the linked document server **2** in the fifth embodiment can be understood from the preceding embodiments, so further description will be omitted.

The dictionary linker **4** is activated when the linked document server **2** receives a tag attachment command. Referring to FIG. **37**, the dictionary linker **4** obtains a copy of the relevant document file, creates a result file, performs a morphemic analysis, processes the words in the document file one by one, and outputs the result file when all words have been processed. These steps (steps **500**, **501**, **502**, **503**, and **508**) are the same as in the third and fourth embodiments, but the processing of each word differs from the processing in those embodiments.

When an unprocessed word is found in step **503**, the dictionary linker **4** first decides whether a dictionary entry for that word is already stored in the dictionary entry store **9** (step **516**). If the dictionary entry of that word is already stored in the dictionary entry store **9**, the dictionary linker **4** generates tags linking the word to the dictionary entry stored in the dictionary entry store **9** (step **511**), and writes the word and these tags in the result file (step **506**). The lines written in the result file when steps **516** and **511** are followed resemble the lines written in the third embodiment, shown in FIG. **25**.

If the dictionary entry of the word is not already stored in the dictionary entry store **9**, the dictionary linker **4** generates a dictionary look-up command tag (step **514**), and writes the word with this dictionary look-up tag and a closing tag in the result file (step **506**). The lines written in the result file when steps **516** and **514** are followed resemble the lines written in the fourth embodiment, shown in FIG. **34**.

FIG. **38** shows an example of the result file output by the dictionary linker **4** for a document beginning "We draw on vast storehouses . . . " when dictionary entries for the words "we" and "on" have already been stored in the dictionary entry store **9**, and dictionary entries for the words "draw," "vast," and "storehouses" have not been stored. The meanings of the tags are the same as in the third and fourth embodiments.

The dictionary entry extractor **8** is activated when the linked document server **2** receives a dictionary look-up command. As in the fourth embodiment, the dictionary entry extractor **8** receives a word as a command parameter. Referring to FIG. **39**, the dictionary entry extractor **8** begins as in the fourth embodiment by determining whether the word is defined in the electronic dictionary **10** (step **401**). If the word is defined, the dictionary entry extractor **8** extracts the dictionary entry of the word from the electronic dictionary **10** arid stores a copy of this entry as a file in the dictionary entry store **9**, as in the third embodiment (step **402**), then outputs the entry to the linked document server **2** (step **409**). If the word is riot defined, the dictionary entry extractor **8** creates an unknown-word message as in the fourth embodiment (step **410**), and outputs this message to the linked document server **2** (step **409**).

The overall operation of the tagged electronic dictionary **5** can be understood from the description given above and the descriptions of the preceding embodiments. When the user selects the dictionary mode button, the dictionary linker **4** generates a result file, without performing any actual dictionary look-up. Words are looked up in the electronic dictionary **10** one by one, when their meanings are requested by the user, as in the fourth embodiment. The dictionary entries are saved in the dictionary entry store **9**, however, as in the third embodiment, so that the same word will not have to be looked up in the electronic dictionary **10** again the next time the definition of the word is requested.

The result file in the fifth embodiment is output more quickly than in the third embodiment, because the electronic dictionary **10** is not, accessed and no dictionary entries are copied from the electronic dictionary **10** to the dictionary entry store **9**. Output of the result file is not as fast as in the fourth embodiment, however, because the dictionary linker **4** must check the dictionary entry store **9** before generating each pair of tags.

The first time the definition of a word is requested, the fifth embodiment returns the definition at the same speed as the fourth embodiment. The definition is not returned as quickly as in the third embodiment, because a dictionary look-up command must be executed.

When the definition of the same word is requested repeatedly, however, the fifth embodiment retrieves the definition from the dictionary entry store **9**, without executing a dictionary look-up command, so the definition is returned just as quickly as in the third embodiment, and faster than in the fourth embodiment.

From the point of view of system efficiency, the fifth embodiment has advantages over both the third and fourth embodiments, in that words are never looked up in the electronic dictionary **10** unnecessarily, the same word is not looked up in the electronic dictionary **10** more than once, and dictionary entries that no one wants to see are not stored unnecessarily in the dictionary entry store **9**.

Sixth Embodiment

The sixth embodiment also combines the advantages of the third and fourth embodiments.

Referring to FIG. **40**, the sixth embodiment comprises the same client device **1**, electronic document store **3**, dictionary entry store **9**, and electronic dictionary **10** as the third embodiment, the same dictionary linker **4** as the fourth embodiment, and a linked document server **2** and dictionary entry extractor **8** that are generally similar to the corresponding elements in the fifth embodiment.

The only difference between the linked document server **2** in the sixth embodiment and the linked document server **2** in the fifth embodiment is that the linked document server **2** in the sixth embodiment does not access the dictionary entry store **9** directly.

Referring to FIG. **41**, when activated by the linked document server **2**, the dictionary entry extractor **8** starts by checking whether a dictionary entry for the word supplied by the linked document server **2** as a command parameter is already stored in the dictionary entry store **9** (step **400**). This step is identical to the corresponding step in the third embodiment. If the dictionary entry is already stored, the dictionary entry extractor **8** reads this dictionary entry from the dictionary entry store **9** (step **412**), and outputs the dictionary entry thus read to the linked document server **2** (step **409**).

If the dictionary entry is not already stored in the dictionary entry store **9**, the dictionary entry extractor **8** proceeds as in the fifth embodiment to look up the word in the electronic dictionary **10** (step **401**), copy its dictionary entry (if found) to the dictionary entry store **9** (step **402**) or create an unknown-word message (step **410**), and output the dictionary entry or unknown-word message to the linked document server **2** (step **409**).

Next the overall operation of the sixth embodiment will be briefly described, starting from the point at which the user selects the dictionary mode button, thereby sending a tag attachment command to the linked document server **2**.

The linked document server **2** activates the dictionary linker **4**, which attaches dictionary look-up command tags to all words in the document, as in the fourth embodiment. The dictionary linker **4** operates according to the flowchart in FIG. **33**, producing a result file like the one shown in FIG. **34**. This file is sent to the client device **1** and displayed as in FIG. **15**.

If the user now selects, for example, the word "storehouses" on the display in FIG. **15**, the command "cgi-bin/pick_dic?storehouse" is sent to the linked document server **2** and executed by the dictionary entry extractor **8** according to the flowchart in FIG. **41**. If the dictionary entry for "storehouse" has already been stored in the dictionary entry store **9**, the dictionary entry extractor **8** quickly returns the stored entry to the linked document server **2**, which sends it to the client device **1**. If the dictionary entry for "storehouse" has not already been stored in the dictionary entry store **9**, the dictionary entry extractor **8** obtains this entry from the electronic dictionary **10**, returns the obtained entry to the linked document server **2**, and also stores this entry in the dictionary entry store **9** for possible futures use.

If the user selects a word that is not defined in the electronic dictionary **10**, the dictionary entry extractor **8** creates a message stating that the selected word is not in the dictionary, and returns this message instead of a dictionary entry.

The sixth embodiment thus has the same advantage as the fourth embodiment in returning a result file quickly, because the dictionary linker **4** attaches tags without accessing either the dictionary entry store **9** or the electronic dictionary **10**. In this regard, the sixth embodiment is faster than the third and fifth embodiments.

In returning the definitions of individual words, the dictionary-access-ready document store **6** is generally faster than the fourth embodiment if the word has been looked up before, because the dictionary entry can be obtained from the dictionary entry store **9** instead of the electronic dictionary **10**, but slower than the fourth embodiment if the word has not been looked up before, because the dictionary entry must be searched for in the dictionary entry store **9** before being obtained from the electronic dictionary **10**, and stored in the dictionary entry store **9** after being obtained from the electronic dictionary **10**. In both cases, the sixth embodiment is slower than the fifth embodiment. When the word has been looked up before, the sixth embodiment is also slower than the fifth embodiment, since the fifth embodiment does not require command execution in this case.

The sixth embodiment resembles the fifth embodiment in that dictionary entries are not stored in the dictionary entry store **9** until specifically requested by the user, thereby avoiding the unnecessary storage of dictionary entries that no one wants to see.

Seventh Embodiment

The seventh embodiment differs from the preceding embodiments by attaching dictionary access information to hypertext documents, as well as to other documents.

Referring to FIG. **42**, the seventh embodiment comprises the same client device **1** and electronic document store **3** as the first embodiment, a generally similar linked document server **2** and dictionary linker **4**, an electronic dictionary **15**, and a menu generator **16**.

The electronic dictionary **15** comprises, for example, a commercially available electronic dictionary, together with dictionary access software equivalent to the dictionary entry extractor **8** in the fourth embodiment. When the linked document server **2** issues a dictionary look-up command and supplies a word as a command parameter, the electronic dictionary **15** returns the dictionary entry for the supplied word. A single dictionary entry can be returned, as illustrated in FIG. **27**, or the electronic dictionary **15** can return dictionary entries for the supplied word and several alphabetically adjacent words, as illustrated in FIG. **16**. If the supplied word is not defined in the electronic dictionary **15**, the electronic dictionary **15** returns a message to that effect.

The linked document server **2** operates as shown in FIG. **43**. Upon receiving information from the client device **1** (step **200**), the linked document server **2** determines whether the information is a command or a file descriptor (step **208**). If the information is a file descriptor, the linked document server **2** gets the described document file from the electronic document store **3** (step **202**), attaches a dictionary mode button (step **205**), and sends the document back to the client device **1**. Differing from the linked document server **2** in the previous embodiments, the linked document server **2** in the seventh embodiment attaches a "Dictionary mode" tag even if the document is a hypertext document, containing links to other documents.

If the information received from the client device **1** is a command, the linked document server **2** determines whether the command is a tag attachment command or a dictionary look-up command (step **201**). If the command is a tag attachment command, the linked document server **2** activates the dictionary linker **4** (step **203**), then receives the

result file output by the dictionary linker **4**, adds an ordinary mode button to the result file (step **207**), and sends the result file back to the client device **1** (step **206**).

If the command is a dictionary look-up command, the linked document server **2** executes the command, thereby activating the software that looks up words in the electronic dictionary **15**, and obtaining the contents of a dictionary entry, or an unknown-word message, from the electronic dictionary **15** (step **210**). The information obtained from the electronic dictionary **15** is then sent as a document to the client device **1** (step **206**).

When activated by the linked document server **2**, the dictionary linker **4** operates as shown in FIG. **44**. After obtaining the relevant document file from the linked document server **2** (step **500**) and creating a result file (step **501**), the dictionary linker **4** performs a morphemic analysis (step **502**). If the document is a hypertext document, the morphemic analysis identifies both character strings that represent words and character strings that represent tags in the document. These character strings are then processed one by one until none are left (until a negative result is obtained in step **518**), at which point the result file is output (step **508**).

When an affirmative result is obtained in step **518**, indicating the presence of an unprocessed character string in the document, the next step is to determine whether the character string is a word or a tag (step **519**). If the character string is not a tag, i.e. if the character string is a word, the dictionary linker **4** generates a dictionary look-up command tag for the character string (step **520**). For the character string "Corporate," the generated tag is, for example, <A HREF="/cgi-bin/look_up?corporate">. In this tag, "/cgi-bin/look_up" is the command that activates the dictionary access software in the electronic dictionary **15**, and "corporate" is the word to be looked up, which is passed to the electronic dictionary **15** as a command parameter. The dictionary linker **4** also generates a closing tag (</A>). Next, the dictionary linker **4** flags the character string (the word "Corporate" in the example above) as having been processed (step **521**), and writes the generated tags and the character string in the result file (step **506**).

If the character string is a tag, giving an affirmative result in step **519**, the dictionary linker **4** proceeds to determine whether the tag is a link tag, that is, an opening tag specifying a hypertext reference to another document (step **522**). If the tag is not a link tag, the dictionary linker **4** simply flags the tag character string as having been processed (step **521**) and writes the tag character string in the result file (step **506**). If the tag is a link tag, however, the dictionary linker **4** activates the menu generator **16** (step **523**), then writes the information returned by the menu generator **16** in the result file (step **506**).

When activated by the dictionary linker **4**, the menu generator **16** receives a pointer to the opening tag found by the dictionary linker **4** to be a link tag. Referring to FIG. **45**, the menu generator **16** begins by generating a menu-opening command tag that makes the hypertext reference specified in the link tag into the default menu selection, and specifies dictionary look-up for the other selections (step **600**). An example will be shown later. The menu generator **16** writes the menu-opening command tag into a temporary file that will be passed back to the dictionary linker **4**, then flags the opening tag as having been processed (step **601**).

The menu generator **16** then examines the next character string following the opening tag, and determines whether this character string is the closing tag of the link (step **602**). If the character string is not the closing tag, the menu generator **16** determines whether the character string is any type of tag (step **603**). If the character string is not a tag, i.e. if the character string is a word, the menu generator **16** stores the character string in a buffer, also writes the dictionary form of the character string as an option line in the temporary file to be passed back to the dictionary linker **4** (step **604**), and flags the character string as having been processed (step **605**). If the character string is a tag, the dictionary linker **4** writes the character string into the temporary file (step **606**) and flags the character string as having been processed (step **605**). After step **605**, the menu generator **16** returns to step **602** to process the next character string.

When the closing tag is encountered in step **602**, the menu generator **16** flags this tag as having been processed (step **607**), then writes the contents of the above-mentioned buffer as a selected-option line, describing the default option, in the temporary file (step **608**), adds a menu closing tag, and passes the temporary file to the dictionary linker **4** (step **609**).

Next, the overall operation of the seventh embodiment will be described in relation to the "Corporate Guidance" hypertext document shown earlier. The file descriptor of this document will be assumed to be the word "guidance."

When the user enters "guidance" on the screen shown in FIG. **12**, for example, the linked document server **2** obtains the "Corporate Guidance" file from the electronic document store **3**, adds a dictionary mode button, and sends the resulting document to the client device **1** in the form shown in FIG. **46**. The first line **22** is the line added by the linked document server **2**, comprising an opening tag containing the "/cgi-bin/into_the_dic" command, then the words "Dictionary mode," then a closing tag. The other lines contain the body of the document, including formatting tags and link tags. For example, the link tag <A HREF= "message"> is a hypertext reference to a document having the file descriptor "message" and containing a presidential message. The client device **1** displays the document as shown in FIG. **47**.

If the user selects the dictionary mode button in FIG. **46**, the linked document server **2** receives and executes the "/cgi-bin/into_the_dic" command, thereby activating the dictionary linker **4**, which in turn activates the menu generator **16**. The dictionary linker **4** and menu generator **16** together generate a result file with the contents **34** shown in FIG. **48**.

The first line in the result file contents **34** comprises the tag <A HREF="/cgi-bin/look_up?corporate"> described above, followed by the word "corporate" and a closing tag (</A>). This is followed by a similar line for the word "Guidance." After a pair of formatting tags, there then appears a menu-opening command tag generated by the menu generator **16** from the tag <A HREF="message">.

The word SELECT identifies this command tag as a menu-opening tag. The name of the menu is given as "select**1**." Arbitrary names such as "select**1**," "select**2**," and so on can be assigned. Next, "onFocus='ref(message)'" indicates that the default menu selection is a hypertext reference to a document with the file descriptor "message." The following "onChange='lookdic(option)" indicates that if the user changes the menu selection from the default selection, the menu option selected by the user is to be looked up in the electronic dictionary **15**. Specifically, "lookdic" is a function executed by the client device **1**, and the word "option" indicates that a selected option is to be supplied as an argument of the function.

The next four lines are option lines written by the menu generator **16**, comprising the dictionary forms of the words

"Message," "from," "the," and "President," each preceded by an <OPTION> tag and followed by the notation "(dic)," indicating that this option selects the dictionary entry of the given word.

These lines are followed by a </SELECT> tag, which is a closing tag indicating the end of the "select1" menu. The result file continues with similar menus for "Global Slogan" and "OKI Electric at a Glance."

The linked document server **2** adds an ordinary mode button to this result file by adding an initial line **35** giving "guidance" as a hypertext reference, and sends the resulting document to the client device **1**. The client device **1** displays the received document as shown in FIG. **49**. The underlines under the words "Corporate" and "Guidance" indicate that these words can be looked up in the dictionary. The three items "Message from the President," "Global Slogan," and "OKI Electric at a Glance" are presented as buttons that call forth menus.

If the user selects the word "Corporate" with the pointing device, the linked document server **2** receives and executes the command "/cgi-bin/look_up?corporate," causing the electronic dictionary **15** to return the dictionary entry for this word. The user then sees a display similar to FIG. **16** or **27**, except that the Japanese definition of "corporate" is given instead of the definition of "storehouse."

If the user selects the "Message from the President" button, however, the tags and other information shown in FIG. **48** cause the client device **1** to display a menu beside the selected button, as shown in FIG. **50**. The top line "Message from the President" in this menu is highlighted to indicate that this is the default selection. If the user chooses this selection, by pressing a button on the pointing device, for example, the file descriptor "message" will be sent to the linked document server **2**, which will return the corresponding document, and the user will be able to read a presidential message, as if he had selected "Message from the President" on the display in FIG. **47**.

If the user wants to know the Japanese meaning of the word "message," he can use the pointing device to change the menu selection as shown in FIG. **51**. When this menu item is selected, the client device **1** executes the "lookdic" function shown in FIG. **48** with the word "message" as an argument. This function generates and sends to the linked document server **2** a "/cgi-bin/look_dic?message" command. By executing this command, the linked document server **2** obtains the dictionary entry for "message" from the electronic dictionary **15**, and sends this dictionary entry back to the client device **1**. The user then sees the Japanese definition of "message."

The seventh embodiment is similar to the fourth embodiment in that dictionary entries are looked up when specifically requested, instead of when the dictionary mode button is selected, and in that the dictionary entries and result file are not stored. By using menus as described above, however, the seventh embodiment is able to provide dictionary access from hypertext documents as well as from other documents, which is a considerable benefit for the user.

As a variation of the seventh embodiment, menu-opening tags and option tags like the ones shown in FIG. **48** can be placed in the document files stored in the electronic document store **3**, thereby providing the user with quicker dictionary access to the words appearing in hypertext links. This feature helps users to move from one hypertext document to another when the documents are in a foreign language. When the user reaches a document he wants to read more thoroughly, he can select the dictionary mode button on the document to have dictionary look-up command tags added to all words in the document, in this variation, the menu generator **16** is used to generate the menu-tagged documents stored in the electronic document store **3**.

Eighth Embodiment

The eighth embodiment also provides dictionary access from hypertext documents, but does so by removing the hypertext links to other documents in the dictionary-access mode.

Referring to FIG. **52**, the eighth embodiment comprises the same client device **1**, electronic document store **3**, and electronic dictionary **15** as the seventh embodiment, a generally similar linked document server **2** and dictionary linker **4**, and a link remover **17**. The link remover **17** communicates with both the linked document server **2** and the dictionary linker **4**.

The linked document server **2** in the eighth embodiment differs from the linked document server **2** in the seventh embodiment in the following regard: upon receiving a tag attachment command, instead of activating the dictionary linker **4**, the linked document server **2** activates the link remover **17**. The link remover **17** subsequently activates the dictionary linker **4**, and the linked document server **2** obtains a result file from the dictionary linker **4** as in the seventh embodiment.

When activated by the linked document server **2**, the link remover **17** operates as shown in FIG. **53**. The link remover **17** first obtains a copy of the document for which the tag attachment command was issued, by having the linked document server **2** transfer the relevant document file from the electronic document store **3** (step **700**). Next, the link remover **17** determines whether this document is a hypertext document (step **701**).

If the document is a hypertext document, the link remover **17** opens a temporary file (step **702**), then starts reading character strings from the top of the document, continuing as long as any character strings remain to be read (step **703**). While reading character strings, the link remover **17** looks for a character string that is the opening or closing tag of a hypertext link to another file (step **704**). When such a tag is read, the character string constituting the tag is discarded, and processing returns to step **703**. Other character strings are written in the temporary file (step **705**), after which processing returns to step **703**.

When all character strings have been processed in this way, the temporary file consists of the entire contents of the hypertext document, except for the hypertext links. The dictionary linker **4** is then activated, and the temporary file is passed to the dictionary linker **4** (step **706**).

If the document is found not to be a hypertext document in step **701**, the document is passed without alteration to the dictionary linker **4** (step **706**).

Upon receiving a temporary file or unaltered document from the link remover **17**, the dictionary linker **4** operates essentially as in the fourth embodiment, following the flowchart in FIG. **33**. The only differences between the dictionary linkers **4** in the fourth embodiment and eighth embodiment are that the dictionary linker **4** in the eighth embodiment receives the document file from the link remover **17** instead of from the linked document server **2** in step **500**, and the dictionary look-up command tags generated in step **514** invoke the electronic dictionary **15**, instead of the dictionary entry extractor. After generating such command tags for all words and writing the words and

command tags in the result file, the dictionary linker **4** passes the result file to the linked document server **2**.

Next, the overall operation of the eighth embodiment will be described, again in relation to the "Corporate Guidance" hypertext document.

When the user retrieves this document, the linked document server **2** attaches a dictionary mode button as in the seventh embodiment, and the user sees the display shown in FIG. **47**. If the user selects the dictionary mode button, the link remover **17** obtains a copy of the "Corporate Guidance" document from the linked document server **2** and removes the hypertext links. Tags such as the <A HREF="message"> and </A> shown in FIG. **46** are removed, for example. The dictionary linker **4** next inserts dictionary look-up command tags. For example, the tags <A HREF="/cgi-bin/look_up?corporate"> and </A> are inserted before and after the word "Corporate," and <A HREF="/cgi-bin/look_up?message"> and </A> are inserted before and after the word "Message." As a result, the user sees the display in FIG. **54**, in which each individual word in the "Corporate Guidance" document is underlined to indicate that the word can be looked up in the electronic dictionary **15**.

The user can look up words by selecting them on the display in FIG. **54** with the pointing device. When all necessary words have been looked up, the user can select the ordinary mode button to return to the display in FIG. **47**. From FIG. **47**, the user can retrieve other documents by selecting the underlined items with the pointing device.

Compared with the seventh embodiment, the eighth embodiment is more convenient for the user to operate, in that dictionary definitions can be obtained without going through a menu selection process. The eighth embodiment is less convenient, however, in that the user cannot proceed directly from the display in FIG. **54** to another linked document.

Ninth Embodiment

The ninth embodiment attaches dictionary access information only to selected words in a non-hypertext document, instead of attaching such information to all words.

Referring to FIG. **55**, the ninth embodiment comprises the same client device **1**, electronic document store **3**, and electronic dictionary **15** as the eighth embodiment, a generally similar linked document server **2** and dictionary linker **4**, and a dictionary access tabulator **18**, which is coupled to the linked document server **2** and dictionary linker **4**. The dictionary access tabulator **18** keeps records indicating the frequency with which dictionary definitions of different words are requested, obtaining this information from the linked document server **2**, and supplies this information on request to the dictionary linker **4**.

The linked document server **2**, dictionary linker **4**, and dictionary access tabulator **18** may all reside in the same computer system or workstation, for example, or they may reside in two or more separate systems. The functions of the dictionary access tabulator **18** may also be built into the electronic dictionary **15**, in which case a separate dictionary access tabulator **18** is unnecessary.

The linked document server **2** in the ninth embodiment operates as shown in FIG. **56**. When the linked document server **2** receives a file descriptor, it obtains the requested document and attaches a dictionary mode button if the document is not a hypertext document (steps **200**, **208**, **202**, **204**, and **205**). When the linked document server **2** receives a tag attachment command, it activates the dictionary linker **4**, and attaches an ordinary mode button to the result file output by the dictionary linker **4** (steps **200**, **208**, **201**, **203**, and **207**). These steps are identical to the corresponding steps performed by the linked document server **2** in the fourth embodiment, shown in FIG. **32**.

When the linked document server **2** receives a dictionary look-up command, it executes the command (step **210**) as in the eighth embodiment, obtaining the dictionary entry for the requested word from the electronic dictionary **15**, or an unknown-word message if the word is not in the dictionary. In addition, the linked document server **2** passes the word to the dictionary access tabulator **18** (step **211**).

After step **205**, **207**, or **211** in FIG. **56**, the linked document server **2** sends the document or dictionary entry obtained from the above processing to the client device **1** (step **206**).

When activated by the linked document server **2**, the dictionary linker **4** operates as shown in FIG. **57**. As in the first embodiment, after obtaining a copy of the document to be tagged (step **500**), creating a result file (step **501**), and performing a morphemic analysis (step **502**), the dictionary linker **4** processes words one by one until no unprocessed words remain (giving a negative result in step **503**), then outputs the result file to the linked document server **2** (step **508**).

When an affirmative result is obtained in step **503**, indicating the presence of an unprocessed word, the dictionary linker **4** sends the word to the dictionary access tabulator **18** together with a command asking the dictionary access tabulator **18** to indicate the number of times the word has been looked up in the past, receives this information from the dictionary access tabulator **18**, and decides whether to attach dictionary access information to the word (step **525**). This decision can be made, for example, according to a threshold: dictionary access information is attached to words that have been looked up at most N times in the past, and is not attached to words that have been looked up more than N times in the past, where N is a non-negative integer. The threshold value N may vary according to the part of speech. For example, the value of N could be set equal to zero for articles, conjunctions, pronouns, and prepositions, and to ten for nouns, verbs, adjectives, and adverbs, in this case, the dictionary linker **4** will stop attaching dictionary access information to articles, conjunctions, pronouns, and prepositions after these words have been looked up once, but will continue attaching dictionary access information to nouns, verbs, adjectives, and adverbs until these words have been looked up eleven times.

When the dictionary linker **4** decides in step **525** to attach dictionary access information to a word, it generates a dictionary look-up command tag and a corresponding closing tag (step **520**), and writes the word and these tags in the result file (step **506**). When the dictionary linker **4** decides not to attach dictionary access information, step **520** is skipped, and only the word is written in the result file in step **506**. After step **506**, the dictionary linker **4** returns to step **503** to process the next word.

Next, the operation of the dictionary access tabulator **18** will be described with reference to FIGS. **58**, **59**, and **60**.

The dictionary access tabulator **18** maintains a table of word look-up records as illustrated in FIG. **58**. Each record comprises a word and the number of times the word has been sent to the dictionary access tabulator **18** from the linked document server **2**; that is, the number of times the linked document server **2** has received a dictionary look-up command for the word. This number of times will be referred to as the look-up count. The table in FIG. **58** indicates that the

word "we" has been looked up once, for example, and the word "draw" three times.

When supplied with a word by the linked document server **2**, the dictionary access tabulator **18** operates as shown in FIG. **59**. First, the dictionary access tabulator **18** determines whether the word is a new word, meaning a word that does not already appear in the table of word look-up counts (step **800**). If the supplied word already appears, the dictionary access tabulator **18** increments its look-count by one (step **801**). If the supplied word does not already appear, the dictionary access tabulator **18** creates a new record in the table of word look-up counts, listing the supplied word with a look-up count of one (step **802**).

When supplied with a word by the dictionary linker **4**, the dictionary access tabulator **18** operates as shown in FIG. **60**. First, the dictionary access tabulator **18** determines whether the word appears in the table of word look-up counts (step **804**). If the supplied word appears, the dictionary access tabulator **18** returns its look-up count to the dictionary linker **4** (step **805**). If the word does not appear, the dictionary access tabulator **18** returns a look-up count of zero to the dictionary linker **4** (step **806**).

Next, the overall operation of the ninth embodiment will be described.

Starting from the initial screen shown in FIG. **12**, if the user enters the file descriptor of the "Corporate Guidance" document, then selects the hypertext link to the "Global Slogan" document, he will see first the display in FIG. **13** (without a dictionary mode button, because the "Corporate Guidance" document is a hypertext document), then the display in FIG. **14** (with a dictionary mode button, because the "Global Slogan" document is not a hypertext document).

If the user selects the dictionary mode button on the display in FIG. **14**, the linked document server **2** activates the dictionary linker **4**, which generates a result file according to the look-up counts maintained by the dictionary access tabulator **18**. The linked document server **2** adds an ordinary mode button, and sends this file back to the client device **1**.

FIG. **61** shows an example of the result file returned by the linked document server **2** to the client device **1** when the word look-up count table has the contents shown in FIG. **58**, and the threshold values of zero and ten are used as described above. The first line in FIG. **61** is the ordinary mode button added by the linked document server **2**. The next line contains only the word "We," because the look-up count (one) for this word exceeds the threshold value (zero) for pronouns. The next line contains the word "draw" together with a dictionary look-up command tag and closing tag, because the look-up count (three) for this word does not exceed the threshold value (ten) for verbs. Other lines are generated similarly by the dictionary linker **4**. The client device **1** displays this result file as shown in FIG. **62**.

The user can now look up the underlined words in FIG. **62** by selecting them with the pointing device. If the user selects the word "storehouses," for example, he will obtain a Japanese definition as shown in FIG. **16** or **27**. If the user selects the word "We," however, no definition will be returned and the display in FIG. **62** will remain unchanged, because no dictionary access tag is attached to this word.

By not underlining words that the user has already looked up a certain number of times, the system reminds the user that he (presumably) already knows these words. More significantly, as the user looks up more and more words, the amount of tag attachment processing that must be carried out by the dictionary linker **4** gradually decreases, and the system's response to selection of the dictionary mode button becomes faster.

Next, a variation of the ninth embodiment will be described. In this variation, the dictionary linker **4** attaches dictionary access information to all words in the document, regardless of their look-up counts, but also attaches tags that vary the way in which each word is displayed, depending on the look-up count of the word. For example, words that have not been looked up before can be displayed in the color blue, words that have been looked up from one to five times in the color green, and words that have been looked up more than five times in the color red.

FIG. **63** shows a hypothetical result file generated by the dictionary linker **4** in this case. The tag <FONT=BLUE> in the first line indicates that the words between this tag and the following closing tag </FONT> are to be displayed in the color blue. The word "We" is accordingly displayed in blue. Green and red are specified similarly. In this case the color red informs the user that the word "vast" has been looked at least six times in the past, suggesting that this might be a word worth learning. The display colors thus provide the user with information that can help the user to decide which words to look up.

Instead of changing the colors in which the words are displayed, the dictionary linker **4** can indicate look-up counts in various other ways. For example, the colors of the underlines below the words can be altered, or italic and bold fonts can be used.

Tenth Embodiment

The tenth embodiment adds a learning function to the ninth embodiment. The tenth embodiment keeps track of both the number of times a word is tagged for dictionary access and the number of times the word has been looked up, and stops tagging words that have been frequently tagged but rarely looked up.

Referring to FIG. **64**, the tenth embodiment comprises the same client device **1**, linked document server **2**, electronic document store **3**, electronic dictionary **15**, and dictionary access tabulator **18** as the ninth embodiment, and a generally similar dictionary linker **4**. The tenth embodiment also comprises a tag tabulator **19**, which is linked to the dictionary linker **4**, and a ratio calculator **20**, which is linked to the dictionary linker **4**, dictionary access tabulator **18**, and tag tabulator **19**.

The dictionary linker **4** in the tenth embodiment operates like the dictionary linker **4** in the ninth embodiment, but with two differences. Referring to FIG. **65**, one difference is that the dictionary linker **4** in the tenth embodiment decides whether to attach dictionary access information to each word on the basis of a tag count and look-up ratio supplied from the ratio calculator **20** (step **527**), instead of on the basis of a look-up count supplied from the dictionary access tabulator **18**. The tag count indicates the number of times the word, or a word with the same dictionary form, has been tagged in the past. The look-up ratio indicates what proportion or those times have resulted in actual look-up of the word by the user.

Specifically, the dictionary linker **4** compares the tag count with a first threshold value and the look-up ratio with a second threshold value. If the tag count is equal to or less than the first threshold value, or the look-up ratio is greater than the second threshold, the dictionary linker **4** decides to attach dictionary access information. If the tag count is greater than the first threshold value and the look-up ratio is equal to or less than the second threshold value, the dictio-

nary linker **4** decides not to attach dictionary access information. For example, the dictionary linker **4** can decide to attach dictionary access information unless the word has been tagged more than five times already, but has not been looked up more than three-tenths of the time.

The other difference is that, when the dictionary linker **4** decides to attach dictionary access information, after generating the necessary tags in step **520**, the dictionary linker **4** passes the dictionary form of the word being tagged to the tag tabulator **19** (step **528**).

The other steps shown in FIG. **65** are identical to the corresponding steps performed by the dictionary linker **4** in the ninth embodiment, shown in FIG. **57**, so a step-by-step description will be omitted.

The operation of the tag tabulator **19** is analogous to the operation of the dictionary access tabulator **18**, described in the ninth embodiment, so explanatory drawings will be omitted. The tag tabulator **19** maintains a table of records indicating the number of times the dictionary sinker **4** has attached dictionary access information to different words. The table is similar to the table of look-up counts shown in FIG. **58**, except that the values indicate tag counts instead of look-up counts. When sent a word by the dictionary linker **4**, the tag tabulator **19** searches for the word in the table of tag counts, increments the tag count of the word if the word already appears in the table, and enters the word in a new record with a tag count of one if the word does not appear. When sent a word by the ratio calculator **20**, the tag tabulator **19** returns the tag count of the word if the word appears in the table, and returns a tag count of zero of the word does not appear.

FIG. **66** illustrates the operation of the ratio calculator **20** when supplied with a word from the dictionary linker **4**. The ratio calculator **20** begins by sending this word to the tag tabulator **19**, receiving the tag count of the word, and deciding whether the tag count is zero (step **900**). If the tag count is not zero, the ratio calculator **20** sends the word to the dictionary access tabulator **18** and receives the look-up count of the word (step **901**). The look-up count is then divided by the tag count to obtain the look-up ratio (step **902**). The look-up ratio is a number between zero and one, inclusive. For example, if a word has been tagged five times and never looked up, its look-up ratio is zero. If the word has been tagged five times and looked up five times, its look-up ratio is one.

If the word has been looked up more often than tagged, which may occur if a user looks up a word repeatedly in the same place in the same document, the look-up ratio is arbitrarily set to one. The look-up ratio is also set to one if the tag count is zero (step **903**). After step **902** or step **903**, the ratio calcuator **20** sends both the tag count and the look-up ratio to the dictionary linker **4** (step **904**).

The overall operation of the tenth embodiment is similar to the overall operation of the ninth embodiment, but more responsive to the user's look-up behavior. If the user does not look up a word even though the word is tagged repeatedly, presumably because the user already knows the word, the system will stop tagging the word as soon as its tag count passes the first threshold value. If the user looks the word up a few times, then stops looking it up, presumably because he has learned the meaning of the word, the system will again stop tagging the word, as soon as the tag count exceeds the first threshold value and the look-up ratio falls to the second threshold value. If the user keeps looking the word up from time to time, however, the system will continue to tag the word as long as the look-up ratio remains above the second threshold value.

As a variation of the tenth embodiment, the dictionary access tabulator **18** and tag tabulator **19** can be adapted to clear the tables of look-up counts and dictionary access tag counts periodically, thereby re-initializing both tables to an empty state. The reason for doing so is that if the tables are not cleared, then once the dictionary linker **4** decides not to tag a given word, it is likely to continue deciding not to tag that word indefinitely, even if in the meantime the user forgets the meaning of the word and would like to look it up again.

Alternatively, the tenth embodiment can be adapted to clear the look-up and tag counts of individual words that have not been looked up for a certain period of time to zero, or to adjust the tag count to a value that makes the look-up ratio greater than the second threshold value, so that the word can be tagged again when it appears in a document. If the user still does not look the word up, the intervals at which the look-up and tag counts are cleared or adjusted can be gradually lengthened.

The tenth embodiment can also be adapted to display tagged words in different colors according to their look-up ratios, or otherwise alter the display of the words to indicate how frequently they have been looked up, as in the variation of the ninth embodiment described above.

When the ninth embodiment or tenth embodiment is practiced in a system that serves multiple users, the dictionary access tabulator **18** and tag tabulator **19** can be adapted to maintain separate tables for each user, so that the words made available for dictionary look-up by one user will not be affected by the past look-up behavior of other users. This feature can easily be implemented in systems that require a user to present a user name or other identifying information when accessing the system.

Alternatively, a single table of look-up counts and a single table of tag counts can be maintained for all users. In this case, in deciding which words to tag for a given user, the system can make use of information gained from the look-up behavior of other users. For example, the system can quickly learn to omit the tagging of the English articles ("a," "an," and "the") and other common words that everyone knows.

As a further variation, the invented document display system can maintain a fixed list of words that are not to be tagged, in place of the tables of look-up counts and tag counts. More generally, the system can maintain a table of conditions specifying how individual words are to be tagged for dictionary access, and how these words are to be displayed. The system may have various specialized dictionaries, for example, and may be adapted to tag specialized terminology for access to the appropriate specialized dictionary, using different colors to indicate to the user that the tags lead to different dictionaries.

The embodiments described above do not exhaust the ways in which the invention can be practiced. The features of different embodiments can be combined to obtain new embodiments. For example, the second and third embodiments can be combined, and many other combinations are possible.

Needless to say, the invention is riot limited to giving Japanese definitions of English words. Definitions of words in any language can be given in any other language. Definitions of words can also be given in the same language as the words themselves.

The electronic dictionaries employed in the invention are not limited to dictionaries that simply give words arid their definitions. The dictionary entries may also indicate the pronunciation of the word, possibly by means of synthesized

speech, in which case the client device **1** should be equipped with facilities for audio output. Dictionary entries may also be illustrated with pictures, in which case the client device **1** should be adapted to display such pictures. Dictionaries that give, for example, short biographies of famous people, may also be used, in addition to dictionaries giving the meanings of words.

When the invention is practiced using a plurality of electronic dictionaries, various means can be used to select a particular dictionary. One known method assigns certain key words to each dictionary, and selects the dictionary having the most key words in the document for which the dictionary mode button was pressed. Another possible method is to select the dictionary on the basis of the contents of tags appearing in the document. Thus a tag indicating an author's name could be used to select a biographical dictionary.

The description of the present invention has referred to words as being defined in an electronic dictionary. The term "defined" should be interpreted broadly enough to include the descriptions given in biographical dictionaries and other such dictionaries.

The ordinary mode button and dictionary mode button do not have to be displayed in the positions shown in the drawings, and do not have to be added by the linked document server **2**. For example, the client device **1** can be adapted to display these buttons beside the "Forward" and "Back" and other buttons at the top of the screen. The ordinary mode button and dictionary mode button can be displayed as underlined words, as shown in the drawings, or they can be made to resemble physical buttons, or they can be displayed as icons or any other recognizable control items.

The ordinary mode button can be omitted. The user can return to the ordinary mode by selecting the "Back" button.

As noted earlier, the invention can be practiced in a computer communication network in which there are multiple client devices, multiple linked document servers, and multiple electronic document stores located at different sites. In this case, a linked document server can attach dictionary mode buttons to documents retrieved from other sites, enabling words in a document to be looked up regardless of the location from which the document is obtained.

The invention can also be practiced in a system that is not networked, or a system in which documents are not linked to one another by hypertext references. For example, the invention can be used to enable words in arbitrary text files to be looked up, by adding dictionary look-up commands or pointers to those files. These commands or pointers need not be contained in hypertext tags, but can be embedded in other forms, such as attribute information.

The dictionary entries returned to the client device do not have to give both the accessed word and its definition, as shown in the drawings. The definition alone can be given.

The invention can also be adapted to generate a result file with embedded dictionary access information automatically when a document is first retrieved, so that the user does not have to select the dictionary mode button each time he needs to look up a word. For example, the client device can be adapted to send a tag attachment command to the linked document server together with the file descriptor of each document to be retrieved.

Those skilled in the art will recognized that further modifications are possible within the scope claimed below.

What is claimed is:

1. A method of displaying an electronic document to a user on a device enabling the user to select items such as character strings in the electronic document, comprising:

attaching dictionary access information to character strings in said electronic document;

displaying said electronic document with said dictionary access information attached, said character strings being visibly marked to indicate presence of said dictionary access information without displaying said dictionary access information itself; and

displaying an entry from an electronic dictionary, if the user selects a character string to which said dictionary access information is attached, said entry defining said charater string; and

removing, from said electronic document, hypertext links to other electronic documents, before said step of attaching dictionary access information.

2. The method of claim **1**, wherein said electronic dictionary comprises hypertext link information enabling entries in said electronic dictionary to be individually retrieved by hypertext links from other documents, and said dictionary access information comprises hypertext links pointing to said entries in said electronic dictionary.

3. The method of claim **1**, wherein said dictionary access information comprises commands for looking up said character strings in said electronic dictionary, and said displaying an entry from said electronic dictionary is carried out by executing one of said commands.

4. The method of claim **1**, further comprising

extracting entries from said electronic dictionary; and

storing the entries thus extracted in a dictionary entry store separate from said electronic dictionary.

5. The method of claim **4**, wherein extracting entries and storing the entries are carried out when said dictionary access information is attached, and said dictionary access information comprises hypertext links pointing to individual entries in said dictionary entry store.

6. The method of claim **4**, wherein extracting entries and storing the entries are carried out when the user selects character strings defined by the entries.

7. The method of claim **6**, further comprising checking said dictionary entry store when dictionary access information is attached, wherein said dictionary access information comprises hypertext links pointing to entries already present in said dictionary entry store, and dictionary look-up commands for obtaining entries not already present in said dictionary entry store from said electronic dictionary.

8. The method of claim **1**, further comprising:

attaching menu information to items in said electronic document having hypertext links to other documents; and

displaying a menu, when the user selects one of said items in said electronic document having hypertext links to other documents, said menu allowing the user to select character strings in said item for display of their dictionary entries, as well as allowing the user to select access to a document to which said item is linked by a hypertext link.

9. The method of claim **1**, wherein said dictionary access information is attached only to character strings defined in said electronic dictionary.

10. The method of claim **1**, further comprising displaying a message stating that the character string selected by the user is not found in the electronic dictionary, if the user selects a character string not defined in said electronic dictionary.

11. The method of claim **10**, further comprising keeping first records indicating how frequently different character strings with attached dictionary access information have been selected by the user.

12. The method of claim 11 wherein, in said displaying said electronic document with said dictionary access information attached, said character strings are displayed in different ways responsive to said first records.

13. The method of claim 11, further comprising deciding, according to said first records, whether to attach said dictionary access information to the individual character strings in said electronic dictionary.

14. The method of claim 11, further comprising keeping second records indicating how frequently different character strings have been displayed with attached dictionary access information.

15. The method of claim 14 wherein, in said displaying said electronic document with said dictionary access information attached, said character strings are displayed in different ways responsive to said first records and said second records.

16. The method of claim 14, further comprising deciding, according to said first records and said second records, whether to attach said dictionary access information to the individual character strings in said electronic dictionary.

17. The method of claim 16, wherein said first records comprise first counts indicating numbers of times respective character strings have been looked up, said second records comprise second counts indicating numbers of times said dictionary access information has been attached to respective character strings, and said deciding comprises further:

calculating a ratio of one of said first records to one of said second counts;

comparing said ratio with a first threshold; and

comparing said one of said second counts with a second threshold.

18. The method of claim 17, wherein said dictionary access information is attached when said ratio exceeds said first threshold, and said dictionary access information is also attached when said one of said second counts does not exceed said second threshold.

19. The method of claim 1, further comprising:

adding to said electronic document a control item enabling the user to select a dictionary-access mode; and

displaying said electronic document with said control item, wherein

said attaching of dictionary access information and displaying said electronic document with said dictionary access information attached are carried out when the user selects said dictionary-access mode.

20. The method of claim 19, further comprising:

storing said electronic document with said dictionary access information attached in a dictionary-access-ready document store; and

retrieving said electronic document with said dictionary access information attached from said dictionary-access-ready document store if said dictionary-access mode is selected again.

21. A document display system having a client device for displaying electronic documents to a user, receiving commands from the user, and enabling the user to select character strings in said electronic document, comprising:

an electronic dictionary coupled to said client device, having a plurality of entries stored on an electronically accessible medium;

a dictionary linker coupled to said client device, operable to attach dictionary access information to character strings in an electronic document displayed by said client device, responsive to a command received by said client device from the user, said dictionary access information causing said client device to retrieve and display an entry from said electronic dictionary when the user selects a character string to which said dictionary access information is attached; and

a link remover coupled to said dictionary linker, operable to remove hypertext links from electronic documents before said dictionary linker attaches said dictionary access information to character strings in said electronic documents.

22. The system of claim 21, further comprising a dictionary-access-ready document store, coupled to said dictionary linker, operable electronic documents together with dictionary access information attached thereto by said dictionary linker.

23. The system of claim 21, also comprising a linked document server coupled between said client device and said dictionary linker, operable to supply said electronic documents to said client device, activate said dictionary linker responsive to said command, receive said dictionary access information from said client device when the user selects a character string to which said dictionary access information is attached, obtain said entry, and supply said entry to said client device.

24. The system of claim 23, wherein said electronic dictionary has hypertext link information enabling said entries to be retrieved from hypertext documents, and said dictionary access information comprises hypertext links pointing to the entries in said electronic dictionary.

25. The system of claim 23, also comprising a dictionary entry extractor coupled to said electronic dictionary, operable to extract entries from said electronic dictionary.

26. The system of claim 25, wherein said dictionary access information comprises commands causing said linked document server to activate said dictionary entry extractor.

27. The system of claim 25, also comprising a dictionary entry store coupled to said dictionary entry extractor, for storing the entries extracted by said dictionary entry extractor from said electronic dictionary.

28. The system of claim 27 wherein, when the user selects a character string to which dictionary access information is attached, if a corresponding entry is already stored in said dictionary entry store, said dictionary entry extractor is operable to obtain said corresponding entry from said dictionary entry store, and if a corresponding entry is not already stored in said dictionary entry store, said dictionary entry extractor is operable to extract a corresponding entry from said electronic dictionary, supply said entry to said client device, and store said corresponding entry in said dictionary entry store.

29. The system of claim 27 wherein, when said dictionary linker is operable to attach dictionary access information to a character string, said dictionary entry extractor is operable to extract a corresponding entry from said electronic dictionary and store said entry in said dictionary entry store, unless said entry is already stored in said dictionary entry store, and wherein said dictionary access information comprises a hypertext link to the corresponding entry store din said dictionary entry store.

30. The system of claim 27 wherein, when said dictionary linker attaches dictionary access information to a character string, said dictionary linker is operable to check said dictionary entry store, attach a hypertext link, pointing to a corresponding entry in said dictionary entry store, as said dictionary access information if said corresponding entry is already stored in said dictionary entry store, and attach a

command for activating said dictionary entry extractor, as said dictionary access information, if said corresponding entry is not already stored in said dictionary entry store.

31. The system of claim **21**, further comprising a menu generator coupled to said dictionary linker, wherein:

when said dictionary linker is attaching said dictionary access information, if said dictionary linker encounters an item to which a hypertext link is already attached, said dictionary linker is operable to activate said menu generator, and said menu generator is operable to generate menu information for displaying a menu allowing the user to select retrieval and display of an electronic document indicated by said hypertext link and also allowing the user to select dictionary access for character strings in said item;

said dictionary linker is operable to attach said menu information to said item; and

said client device is operable to display said menu when the user selects said item.

32. The system of claim **21**, wherein said dictionary linker is operable to attach said dictionary access information only to character strings defined in said electronic dictionary.

33. The system of claim **21**, wherein said client device is operable to display a message indicating that no dictionary entry is present, if the user selects a character string not defined in said electronic dictionary.

34. The system of claim **21**, further comprising a dictionary access tabulator coupled to said dictionary linker, for keeping records about past selections, by the user, of character strings to which said dictionary access information was attached.

35. The system of claim **34**, wherein said dictionary linker uses the records kept by said dictionary access tabulator in deciding whether to attach said dictionary access information to said character strings.

36. The system of claim **34**, wherein said dictionary linker, in attaching said dictionary access information, uses the records kept by said dictionary access tabulator by attaching attribute information causing said client device to display said character strings in different ways indicating to the user how frequently said character strings have been previously selected.

37. The system of claim **34**, wherein the records kept by said dictionary access tabulator comprises first counts indicating how frequently said character strings are selected by the user.

38. The system of claim **37**, further comprising a tag tabulator coupled to said dictionary linker, operable to keep second counts indicating how frequently said dictionary access information has been attached to said character strings.

39. The system of claim **38**, wherein said dictionary linker uses said first counts and said second counts in deciding whether to attach said dictionary access information to said character strings.

40. The system of claim **39**, further comprising:

a ratio calculator coupled to said dictionary linker, operable to calculate ratios of said first counts to said second counts; wherein

said dictionary linker uses said ratios and said second counts in deciding whether to attach said dictionary access information to said character strings.

41. The system of claim **40**, wherein said dictionary linker is operable to compare said ratios with a first threshold, compare said second counts with a second threshold, and attach said dictionary access information to character strings the rations of which exceed said first threshold, and to character strings the second counts of which do not exceed said second threshold.

42. The system of claim **21** wherein, when said dictionary linker attaches said dictionary access information to character strings in said electronic document, said client device is operable to display said electronic document with the character strings to which said dictionary access information is attached visibly marked to indicate that said dictionary access information is present, without displaying said dictionary access information itself.

* * * * *