

Form PTO-1449 (modified)	Attorney Docket No. SVL920020050US1	Serial No. 10/198,827
LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT(S)' INFORMATION DISCLOSURE STATEMENT	Applicant(s): GUEST, Michael et al.	
(Use several sheets if necessary)	Filing Date: July 18, 2002	Group Art Unit:

U.S. PATENT DOCUMENTS

[illegible]

Examiner

Date Considered

Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance *and* not considered. Include copy of this form with next communication to applicant(s).



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.

**TRANSMITTAL
FORM**

(to be used for all correspondence after initial filing)

Application Number	10/198,827
Filing Date	July 18, 2002
First Named Inventor	Michael Alan Guest
Art Unit	
Examiner Name	
Attorney Docket Number	SVL920020050US1

Total Number of Pages in This Submission

ENCLOSURES (Check all that apply)

☒ Fee Transmittal Form	☐ Drawing(s)	☐ After Allowance communication to Group
☐ Fee Attached	☐ Licensing-related Papers	☐ Appeal Communication to Board of Appeals and Interferences
☐ Amendment/Reply	☐ Petition	☐ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
☐ After Final	☐ Petition to Convert to a Provisional Application	☐ Proprietary Information
☐ Affidavits/declaration(s)	☐ Power of Attorney, Revocation Change of Correspondence Address	☐ Status Letter
☐ Extension of Time Request	☐ Terminal Disclaimer	☒ Other Enclosure(s) (please identify below):
☐ Express Abandonment Request	☐ Request for Refund	1) Return postcard
☒ Information Disclosure Statement	☐ CD, Number of CD(s) _____	2) Certificate of Mailing
☐ Certified Copy of Priority Document(s)	Remarks	3) Form 1449
☐ Response to Missing Parts/Incomplete Application		
☐ Response to Missing Parts under 37 CFR 1.52 or 1.53		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Samuel A. Kassatly
Signature	
Date	10/25/2004

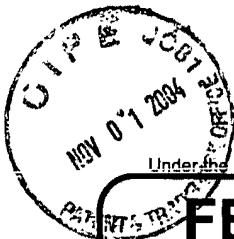
CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Typed or printed name	Samuel A. Kassatly		
Signature		Date	10/25/2004

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



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.

FEE TRANSMITTAL for FY 2004

Effective 10/01/2003. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT (\$) 0

Complete if Known

Application Number 10/198,827
Filing Date July 18, 2002
First Named Inventor Michael Alan Guest
Examiner Name
Art Unit
Attorney Docket No. SVL920020050US1

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number 09-0460
Deposit Account Name International Business Machines

The Director is authorized to: (check all that apply)

☒ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s) or any underpayment of fee(s)

☐ Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) (\$) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

Total Claims	Extra Claims	Fee from below	Fee Paid
Independent	-20** =	X	
Multiple Dependent	-3** =	X	

Large Entity		Small Entity		Fee Description
Fee Code	Fee (\$)	Fee Code	Fee (\$)	
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple dependent claim, if not paid
1204	86	2204	43	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) (\$) 0

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Small Entity

Fee Code	Fee (\$)	Fee Code	Fee (\$)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for ex parte reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1401	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	
1403	290	2403	145	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,330	2453	665	Petition to revive - unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	770	2809	385	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify)

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) (\$) 0

SUBMITTED BY

Name (Print/Type) Samuel A. Kassatly
Registration No. (Attorney/Agent) 32,247
Telephone 408-323-5111
Signature
Date 10/25/2004

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Bright Data Ltd. v. Metacluster LT UAB

IPR2023-01442, EX. 1002

153 of 318

07-22-02 10198827 071802

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
UTILITY PATENT APPLICATION TRANSMITTAL UNDER 37 CFR 1.53(b)**

Address to:	Attorney Docket No.	SVL920020050US1
Assistant Commissioner for Patents	Inventor(s)	GUEST, Michael et al.
Box Patent Application	Express Mail Label No.	EL537038042US
Washington, DC 20231	Total Pages	42 (Not including references)
	Assignee Name	International Business Machines Corporation
	Assignee Residence	Armonk, New York

Title of Application: "SYSTEM AND METHOD FOR DATA RETRIEVAL AND COLLECTION IN A STRUCTURED FORMAT"

Transmitted with the patent application are the following:

- 1 Page(s) Transmittal form (in duplicate)
- 19 Page(s) Specification, claims, abstract
- 7 Page(s) Formal Drawings
- 4 Page(s) Declaration and Power of Attorney
- 1 Page(s) Recordation of Assignment
- 2 Page(s) Assignment of the Invention to International Business Machines Corporation
- 6 Page(s) Form 1449A and Information Disclosure Statement
- 27 References cited in Form 1449A
- 1 Page(s) Return Receipt Postcard (MPEP 503)

This application is a: ☐ Continuation ☐ Divisional ☐ Continuation-in-Part of prior application of:

Fee Calculation

	Claims Filed		Extra	Rate	Fees
Basic Fee					\$740.00
Total Claims	40	-20 =	20	\$18.00	\$360.00
Independent Claims	3	-3 =	0	\$84.00	\$0.00
Multiple Dependent Claim				\$280.00	\$0.00
Assignment (Authorization to record assignment on recordation form)				\$40.00	\$0.00
				TOTAL	\$1,100

[X] The Commissioner is hereby authorized to charge the indicated fees and credit any overpayments, and to charge any additional fees required under 37 CFR 1.16 or 1.17 to **Deposit Account No. 50-0219**. A duplicate copy of this transmittal form is enclosed.

EXPRESS MAIL CERTIFICATE

I hereby certify that the above paper/fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated below and is addressed to the Assistant Commissioner for Patents, Washington, DC 20231
Date of Deposit: 07/18/2002
Person mailing paper/fee: Samuel A. Kassatly

Signature

smk

Respectfully submitted,
GUEST, Michael et al.

Samuel A. Kassatly

Samuel A. Kassatly
Reg. No. 32,247
Attorney for Applicants
Telephone: 408-323-5111
Samuel A. Kassatly, Law Office
6819 Trinidad Drive
San Jose, CA 95120-2056

SYSTEM AND METHOD FOR DATA RETRIEVAL AND COLLECTION IN A STRUCTURED FORMAT

5

FIELD OF THE INVENTION

10 The present invention generally relates to the field of data processing, and particularly to a software system and associated method for use with Web pages to transfer data from a Web page source file to a target database. More specifically, this invention pertains to a method for parsing a Web page source file, collecting the specified data on the Web page, automatically retrieving the specified data on a predetermined frequency or scheduled time, and then storing the collected data in a
15 database structure.

BACKGROUND OF THE INVENTION

20 The World Wide Web (WWW) is comprised of an expansive network of interconnected computers upon which businesses, governments, groups, and individuals throughout the world maintain inter-linked computer files known as Web pages. Users navigate these pages by means of computer software programs commonly known as Internet browsers. Web sites and Web pages found on the WWW present a wide range of data to users in varying formats. This data, such as weather
25 information or stock quotes, may be useful in a variety of ways to the Internet user beyond simply viewing that data in the form and location as it is displayed on the Internet.

30 Transferring data found on websites to a database format would allow significant data management and manipulation capabilities afforded through standard database

software. However, there are currently no effective methods for capturing and replicating that data within a database application. The user can always manually transfer the data viewed on the Internet into a database. However, when transfer is done on a regular basis, such as might be done for a regular recording of stock prices, the transfer of data from the website to the user's database becomes inefficient, labor
5 intensive, and prone to transcription errors. This is particularly true for large amounts of data on a large variety of Web pages that are retrieved periodically, such as every 10 minutes, 45 minutes, daily, etc.

10 Current systems exist for transferring Internet data to databases for archiving Web content in the form of a source file and visual representation, allowing subsequent viewing of that data at a later date. While allowing the user to capture data and archive it, these systems do not provide a mechanism for parsing individual data elements, replicating that data in a relational database format, or automatic data collection in a
15 manner specified by the user.

Another system described in U.S. Patent No. 6,078,924 to Ainsbury et. al. retrieves data from a wide range of document formats and converts that data into a common format and location. This system includes the capability to collect data automatically in a
20 manner specified by the user. It also allows the retrieval of data from a wide range of data formats, including the capability to retrieve data from HTML documents. However, while the data is collected and replicated in a central format and location such as a spreadsheet, it does not effectively account for issues of relational database replication.

25 Thus, there is need for a data collection and retrieval system and method to efficiently manipulate and analyze data displayed on websites. The system and method should be able to retrieve Internet data automatically and to assign that data to appropriate places in a database structure defined by the user. For data that changes regularly such as stock quotes or weather data, a system is desired which will

automatically retrieve data based on a user defined schedule. The need for such a system and associated method has heretofore remained unsatisfied.

SUMMARY OF THE INVENTION

5

10

The system and method for data collection and retrieval satisfy this need, and allow the capture and replication of data presented at various Web pages into a database application through text parsing of, for example, the HTML source code of the retrieved documents. Consequently, users may efficiently collect a wide range of data from a wide variety of sources, and have that data available in user-specified applications, for further data management and processing.

15

The data collection and retrieval system of the current invention allows the user to select one or more Web pages containing data of interest; to specify exactly which data within any page is to be captured; to specify how frequently data is to be collected; and to specify the conditions for collection and retrieval. The conditions for replication include data formatting and database parameters. The advantage of the present system and method is realized through efficient, automated data collection that would otherwise be impractical.

20

The present system and method generally comprise an initialization stage and an automatic execution stage. The initialization stage provides the user interface which allows the user to select:

25

- the source file that contains data the user wishes to copy;
- the target database that will receive the data; and
- the timing criteria for automatic transfer of data.

The automatic execution stage automatically transfers the data from the HTML source file to the target database as instructed by the user in the initialization stage.

30

BRIEF DESCRIPTION OF THE DRAWINGS

The various features of the present invention and the manner of attaining them will be described in greater detail with reference to the following description, claims, and drawings, wherein reference numerals are reused, where appropriate, to indicate a correspondence between the referenced items, and wherein:

FIG. 1 is a schematic illustration of an exemplary operating environment in which a data collection and retrieval system of the present invention can be used;

FIG. 2 is a schematic illustration further showing the manner in which the data collection and retrieval system of the present invention accesses data residing on various Web pages;

FIG. 3 is a block diagram that illustrates a high level architecture of the system of FIG. 1 shown used in the context of the World Wide Web;

FIG. 4 is a flow chart that depicts the operation of the initialization stage of the system of FIGS. 1 and 3,

FIG. 5 is comprised of FIGS. 5A and 5B, and represents a flow chart that depicts the operation of the automatic execution stage of the system of FIGS. 1 and 3; and

FIG. 6 is a screen shot of a Web page with data that can be collected and retrieved by the system of FIG. 1.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The following definitions and explanations provide background information pertaining to the technical field of the present invention, and are intended to facilitate the understanding of the present invention without limiting its scope:

Database: The term "database" refers, for example, to a database, a data storage device, or a memory.

GUI: Graphical User Interface, a program interface that takes advantage of the computer's graphics capabilities to make the program easier to use.

HTML (Hypertext Markup Language): A standard language for attaching presentation and linking attributes to informational content within documents. During a document authoring stage, HTML "tags" are embedded within the informational content of the document. When the Web document (or "HTML document") is subsequently transmitted by a Web server to a Web browser, the tags are interpreted by the browser and used to parse and display the document. In addition to specifying how the Web browser is to display the document, HTML tags can be used to create hyperlinks to other Web documents.

Internet: A collection of interconnected public and private computer networks that are linked together with routers by a set of stand protocols to form a global distributed network.

Schema: A database design, comprised of tables with columns, indexes, constraints, and relationships to other tables. The column specification includes a data type and related parameters such as the precision of a decimal or floating-point number.

Server: A software program or a computer that responds to requests from a Web browser by returning ("serving") Web documents.

5 URL (Uniform Resource Locator): A unique address that fully specifies the location of a content object on the Internet. The general format of a URL is protocol://server-address/path/filename.

10 Web browser: A software program that allows users to request and read hypertext documents. The browser gives some means of viewing the contents of Web documents and of navigating from one document to another.

15 Web Site: A database or other collection of inter-linked hypertext documents ("Web documents" or "Web pages") and associated data entities, which is accessible via a computer network, and which forms part of a larger, distributed informational system such as the WWW. In general, a Web site corresponds to a particular Internet domain name, and includes the content of a particular organization. Other types of Web sites may include, for example, a hypertext database of a corporate "intranet" (i.e., an internal network which uses standard Internet protocols), or a site of a hypertext system that uses document retrieval protocols other than those of the WWW.

20 World Wide Web (WWW): An Internet user - server hypertext distributed information retrieval system.

25 FIG. 1 portrays the overall environment in which a data collection and retrieval system 10 according to the present invention may be used. The system 10 includes a software or computer program product that is typically embedded within, or installed on a host server (or a client server or machine) 15. Alternatively, the system 10 can be saved on a suitable storage medium such as a diskette, a CD, a hard drive, or like devices. While the system 10 will be described in connection with the WWW, the

system 10 can be used with a stand-alone database of terms that may have been derived from the WWW or other sources.

5 The cloud-like communication network 20 is comprised of communication lines and switches connecting servers such as servers 25, 27, to gateways such as gateway 30. The servers 25, 27 and the gateway 30 provide the communication access to the Internet. Users, such as remote Internet users, are represented by a variety of computers such as computers 35, 37, 39, and can query the host server 15 for desired information through the communication network 20.

10 The server 15 is connected to the network 20 via a communications link 42 such as a telephone, cable, or satellite link. The servers 25, 27 can be connected via high-speed Internet network lines 44, 46 to other computers and gateways. The servers 25, 27 provide access to stored information such as hypertext or Web documents indicated generally at 50, 55, and 60. The hypertext documents 50, 55, 60 most likely include
15 embedded hypertext link to other locally stored pages, and hypertext links 70, 72, 74, 76 to other webs sites or documents 55, 60 that are stored by various Web servers such as the server 27.

20 Typically, the system 10 accesses Web pages that have been selected by the user. These Web pages are the source of the data to be transferred to a user defined database. As an example, in FIG. 2 the data collection and retrieval system 10 connects through the communication network 20 to an Internet portal 205 such as Yahoo!®. The user selects a subsite 210 on Yahoo!®, and continues selecting subsites as shown by
25 sub-subsite 215, until a webpage such as webpage 220 is found that contains the data in which the user is interested to transcribe (or copy) to a database or storage 370.

30 An exemplary high-level architecture of the data collection and retrieval system 10 is shown in FIG. 3. System 10 includes an initialization stage 305 and an automatic execution stage 310. The initialization stage 305 provides the user interface that allows

the user to select a source Web page, i.e., 220, the target database, i.e., 370, that will receive and store the data, and the timing criteria for the automatic transfer of data.

5 The automatic execution stage 310 automatically transfers the data from the HTML source file 320 for the Web page 220, to the target database 370 as instructed by the user in the initialization stage 305. The initialization stage 305 includes a user input GUI 315, a source file 320, the parser 325, an array file 330, a refined user GUI 340, and an array locations file 345. The automatic execution stage 310 includes a timer 350, a defined source file 355, an array file 360, a data retrieval module 365, and the target
10 database 370.

In operation, and with further reference to the operational flowcharts of FIGS. 4 and 5, the user opens system 10 in step 405, and the system 10 displays the user input GUI 315. In step 410, the user specifies the URL for an HTML page 220 that contains data
15 to be transcribed to the target database 370. In so doing, the user selects the Web page 220 as shown by FIG. 2, and saves the Web page URL in system 10. The user may select the URL of the source file (320) by manually selecting the Web page URL and copying it to a specific field. Alternatively, the user may open the URL for a source file 320 and automatically save the address to a "favorites" list in a designated list, field,
20 drop-down menu, or list box.

System 10 then downloads the HTML source file 320 for the Web page 220 in step 415. In step 420 the parser 415 processes the source file 320, tokenizes the HTML format of the Web page 220 and places it in a data array 330. Ideally, the parser 325
25 divides the data in the source into identifiable units of the Web page 220, exactly replicating the Web page 220 and displaying the replicated Web page to the user in step 425 as the refined user input GUI 340. The purpose of the parsing is to isolate potentially usable data and to describe how to subsequently locate this data in the Web page source. The system 10 presents a possible schema for storing the selected data
30 in a relational database. This may be based on existing tables in the database, or it may

be a default schema constructed to match the selected data. If needed, the system 10 creates the database schema and automatically transfers the data to the database, performing data type conversions as necessary.

5 Preferably, but not necessarily, the user has access only to data that is embedded in the HTML source file 320 of the replicated Web page. For example, the user might select a Web page 220 containing stock quotes for a variety of stocks. From Web page 220, the user can only select quotes that are displayed and part of the HTML source file 320.

10 The user then selects data and or categories of interest in step 430. In step 435, system 10 saves the sequence of data or categories selected by the user to an array locations file 345. System 10 remembers the location of the data in the array 330 and retrieves data from the same locations on each successive data transfer.

15 In step 440, the user sets the timer 350 by selecting the frequency of data retrieval; i.e., every 5 minutes, 15 minutes, at a specific time such as 12:00 p.m., etc. The selection frequencies can be stored along with other control information, in database tables. Examples of such other control information include URLs, data arrays, user-
20 selected data element locations and formats, and locations for storing the data in the database. The user then specifies in step 445 where to store the retrieved data, such as database 370. The retrieved data may be stored in a specific database, the user may specify specific fields and records in the database, and the user may also specify the format the data should assume after it has been transferred.

25 The initialization stage 305 is complete once the user has specified the source file 320, the desired data or categories for retrieval, the destination for the retrieved information, database 370, and the frequency of operation. System 10 can now perform the automatic retrieval of data as illustrated by the process flow charts of FIGS. 5A and
30 5B.

At a time specified by the user, the timer 350 automatically begins the data retrieval process in step 505 and accesses the Web page 220 in step 510. Each time system 10 accesses the Web page 220, a new source file 355 is created to reflect the changing data on the Web page 220. The parser 325 parses the source file 355 in step 515 and creates an array file 360 in which the data is accessible to the data retrieval module 365.

The data retrieval module 365 extracts the data from the array 360 in step 520 and places it in step 525 in the database 370 as previously specified by the user. This process is repeated by the timer 350 at the frequency or times previously specified by the user in step 440 of FIG. 4.

Optionally, after system 10 retrieves selected data from the source file 355 in step 520, system 10 may map the data previously stored in database 370 onto the new array 355 retrieved from the source file 320 in step 530. If in step 535 the stored array in database 370 matches array 360, system 10 continues to step 540 and transfers the data in array 320 to the appropriate locations in database 370. Operational control then transfers back to the timer 350 in step 505 until the next specified time occurs.

If, however, the two arrays do not match in step 535, system 10 automatically attempts to adjust the new array 360 to match the array stored in database 370 in step 545. If the arrays match in step 550, system 10 proceeds, as before, to step 540, stores the data in database 370, and returns control to the timer in step 505. If the new array 360 does not match the array stored in database 370, system 10 notifies the user in step 555, and stops the data retrieval process in step 560 until the user approves the adjustment or reconfigures the retrieval process.

An example of an exemplary webpage 220 from which a user wishes to transcribe data is shown in FIG. 6. The data the user wants to collect and retrieve is shown, for

instance, at locations 605 and 610. The user may specify that system 10 retrieve just the data, or the data and categories describing the data (i.e., metadata). At location 605, the user selects both the category and data as shown in Table 1 below.

5 Table 1: Categories and data location 605

Category	Data
Stock Symbol	IBM
Last Trade	9:43 am
	84.90
Change	+0.90
	+1.07
Volume	451,200

The value for price and time at location 605 does not have a complete descriptive category, so the user provides to system 10 the desired category "Last Trade Time" and "Last Trade Value" for the values listed under "Last Trade". Similarly, the user assigns new categories "Change Value" and "Change Percent" to differentiate the two values found under the category "Change" at location 605. The user also wishes to select the date and time this transaction took place. This data is found at location 610, but has no associated categories. The user selects the value "May 23 2002" and assigns the category "Date" and selects "10:03 am ET" and assigns the category "Transfer Time".

15 The data as stored in the target database 370 is shown in Table 2 below.

Table 2: Categories and data stored in the target database 370.

Category	Data
Stock Symbol	IBM
Last Trade Time	9:43 am
Last Trade Value	84.90
Change Value	+0.90
Change Percent	+1.07%
Volume	451,200
Date	May 23 2002
Transfer Time	10:03 am ET

The user then specifies the times data should be transferred from the Web page to the database: 12 p.m., 3 p.m., and 5 p.m. eastern standard time. System 10 then sets the timer 350 to initiate data collection and retrieval at those times, transferring the data from Web page 220 to database 370 at those times, in the configuration specified by the user.

It is to be understood that the specific embodiments of the invention that have been described are merely illustrative of certain application of the principle of the present invention. Numerous modifications may be made to the data collection and retrieval system invention described herein without departing from the spirit and scope of the present invention. Moreover, while the present invention is described for illustration purpose only in relation to the WWW, it should be clear that the invention is applicable as well to any collection of data or databases accessible either through an Internet or Intranet connection.

CLAIMS

What is claimed is:

1. A method of selectively retrieving data from a source file and automatically collecting retrieved data in a target database, comprising:

initializing a data retrieval stage by:

selecting the source file;

parsing selected data elements in the source file into a list of data elements;

selecting at least one of the data elements from the list to be stored in the target database;

selecting the target database and a schema in which the at least one of the data elements is to be stored in the target database; and

automatically collecting at least one of the parsed data elements by:

transferring at least one of the parsed data elements from the source file to the target database; and

storing at least one of the transferred data elements in the target database in the schema.

2. The method of claim 1, wherein initializing the data retrieval stage further comprising setting a schedule frequency for transferring the at least one of the parsed data elements from the source file to the target database.

3. The method of claim 1, further comprising processing the at least one of the transferred data elements stored in the target database.

4. The method of claim 3, wherein processing the at least one of the transferred data elements is executed automatically.

5. The method of claim 4, wherein initializing the data retrieval stage is executed automatically.

6. The method of claim 1, wherein the data structure format is different from the schema of the target database.

7. The method of claim 6, wherein the data structure format is a network-based format.

8. The method of claim 7, wherein the data structure format is a markup language.

9. The method of claim 8, wherein the data structure format is HTML.

10. The method of claim 6, wherein the target database is a relational database.

11. The method of claim 1, wherein initializing the data retrieval stage further comprises presenting selection options to a user.

12. The method of claim 1, wherein initializing the data retrieval stage further comprises presenting options for the design of the schema to a user.

13. The method of claim 1, wherein the at least one of the data elements is dynamic.

14. The method of claim 13, wherein initializing the data retrieval stage further comprises analyzing the at least one of the data elements.

15. The method of claim 14, wherein the at least one of the data elements comprises data that changes periodically.

16. The method of claim 1, wherein processing the at least one of the transferred data is executed manually.

17. The method of claim 1, further comprising automatically updating the at least one of the transferred data elements that are stored in the target database.

18. The method of claim 17, wherein automatically updating the at least one of the transferred data elements comprises reading control information from database tables; and

comparing at least one of the selected data elements to the corresponding at least one of the data elements stored in the database.

19. The method of claim 1, wherein automatically updating the at least one of the transferred data elements comprises periodically updating the at least one of the transferred data elements.

20. The method of claim 1, further comprising reinitializing the data retrieval stage.

21. A computer program product having instruction codes for selectively retrieving data from a source file and for automatically collecting retrieved data in a target database in a schema, comprising:

a first set of instruction codes for initializing a data retrieval stage by:

selecting the source file;

parsing selected data elements in the source file into a list of data elements;

selecting at least one of the data elements from the list to be stored in the target database;

selecting the target database and the schema in which the at least one of the data elements is to be stored in the target database; and

a second set of instruction codes for automatically collecting at least one of the parsed data elements by:

transferring at least one of the parsed data elements from the source file to the target database; and

storing at least one of the transferred data elements in the target database in the schema.

22. The computer program product of claim 21, wherein the data structure format is different from the schema of the target database.

23. The computer program product of claim 22, wherein the data structure format is a network-based format.

24. The computer program product of claim 23, wherein the data structure format is a markup language.

25. The computer program product of claim 24, wherein the data structure format is HTML.

26. The computer program product of claim 21, wherein the target database is a relational database.

27. The computer program product of claim 21, wherein the at least one of the data elements is dynamic.

28. The computer program product of claim 21, wherein the at least one of the data elements comprises data that changes periodically.

29. The computer program product of claim 21, further comprising a third set of instruction codes for presenting transfer options to a user.

30. The computer program product of claim 21, further comprising a fourth set of instruction codes for reinitializing the data retrieval stage.

31. A system for selectively retrieving data from a source file and for automatically collecting retrieved data in a target database in a schema, comprising:

means for initializing a data retrieval stage by:

- selecting the source file;
- parsing selected data elements in the source file into a list of data elements;
- selecting at least one of the data elements from the list to be stored in the target database;
- selecting the target database and the schema in which the at least one of the data elements is to be stored in the target database; and

means for automatically collecting at least one of the parsed data elements by:

- transferring at least one of the parsed data elements from the source file to the target database; and
- storing at least one of the transferred data elements in the target database in the schema.

32. The system of claim 31, wherein the data structure format is different from the schema of the target database.

33. The system of claim 32, wherein the data structure format is a network-based format.

34. The system of claim 33, wherein the data structure format is a markup language.

35. The system of claim 34, wherein the data structure format is HTML.

36. The system of claim 31, wherein the target database is a relational database.

37. The system of claim 31, wherein the at least one of the data elements is dynamic.

38. The system of claim 31, wherein the at least one of the data elements comprise data that changes periodically.

39. The system of claim 31, further comprising means for presenting transfer options to a user.

40. The system of claim 31, further comprising means for reinitializing the data retrieval stage.

**SYSTEM AND METHOD FOR DATA RETRIEVAL
AND COLLECTION IN A STRUCTURED FORMAT**

5

ABSTRACT OF THE INVENTION

A data collection and retrieval system and associated method allow the capture and replication of data presented at various Web pages into a database application through text parsing of the HTML source code of that document. The system allows the user to select one or more Web pages containing data of interest; to specify exactly which data within any page is to be captured; to specify how frequently data is to be collected; and to specify the conditions for collection and retrieval. The advantage of the system is realized through efficient, automated data collection that would otherwise be impractical. The system includes an initialization stage and an automatic execution stage. The initialization stage provides the user interface which allows the user to select the source file that contains data the user wishes to copy, target database that will receive the data, and timing criteria for automatic transfer of data. The automatic execution stage automatically transfers the data from the source file to the target database as instructed by the user in the initialization stage.

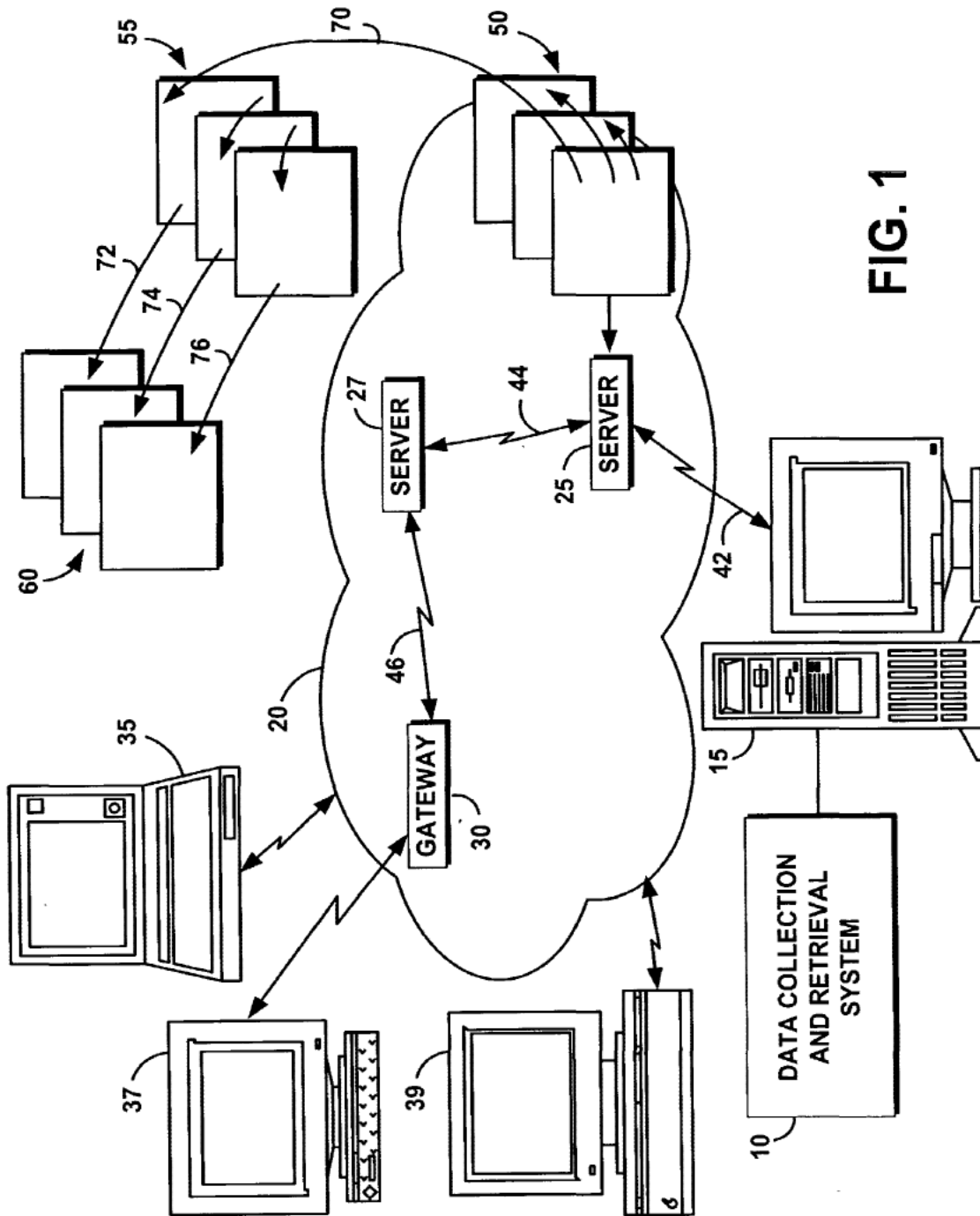


FIG. 1

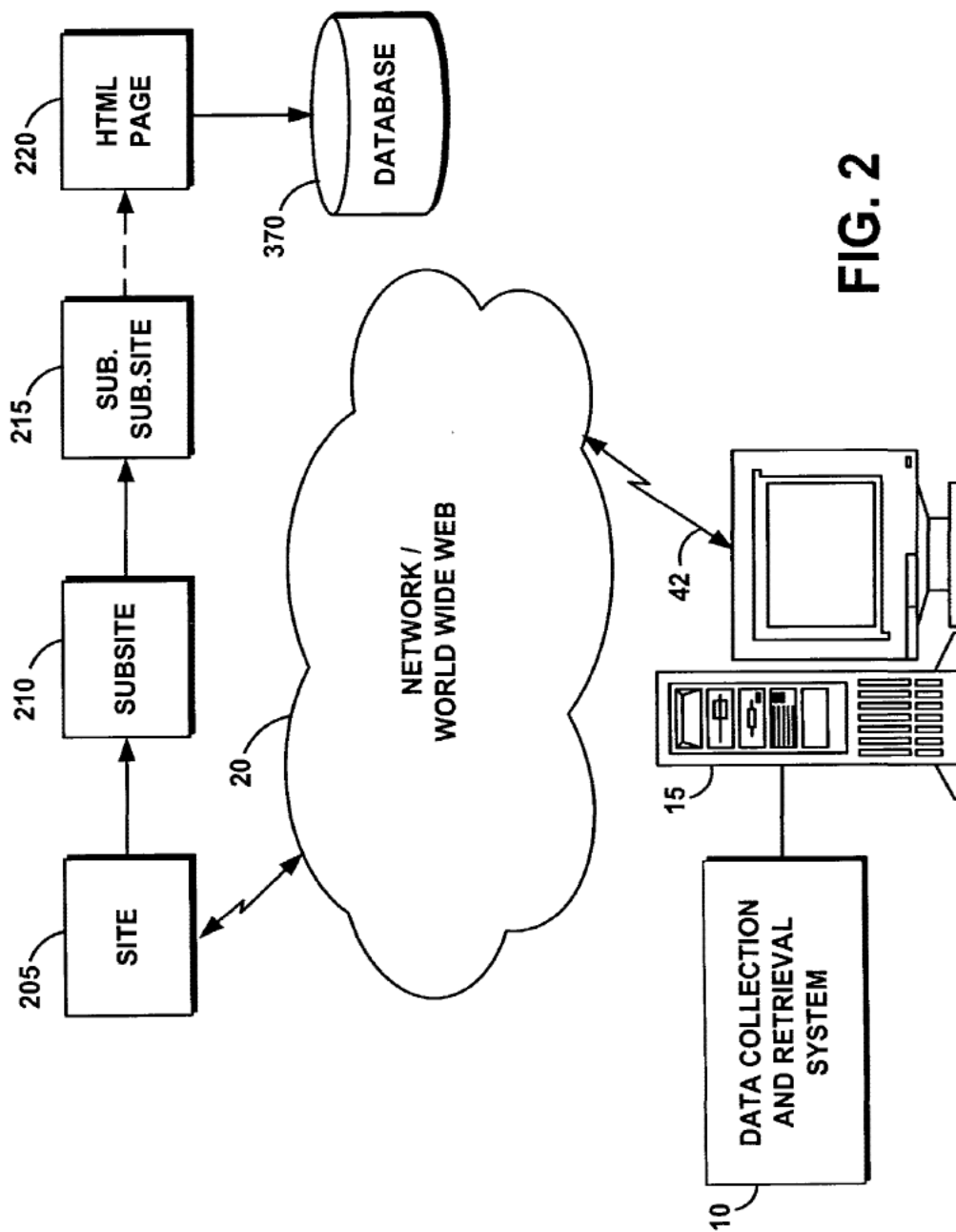
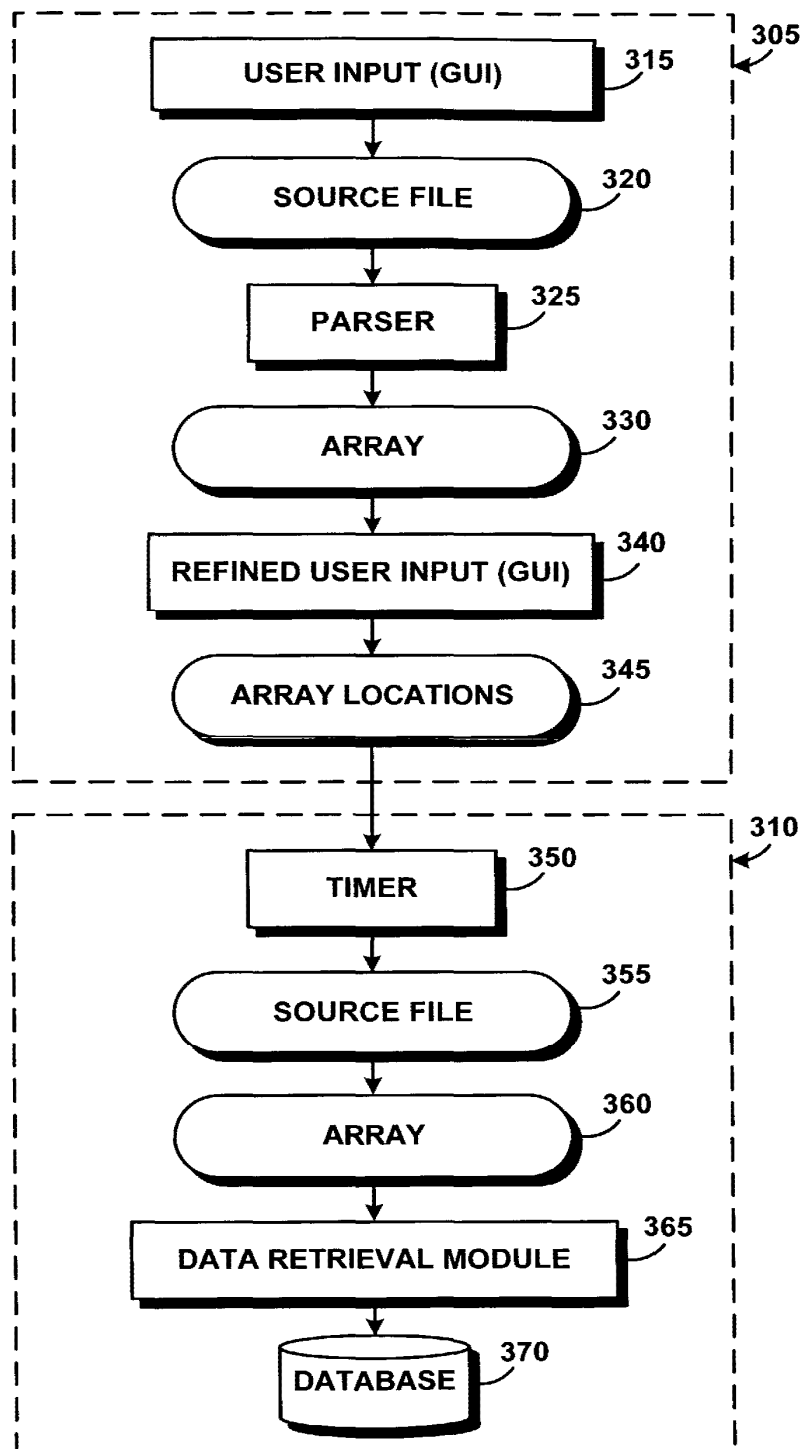
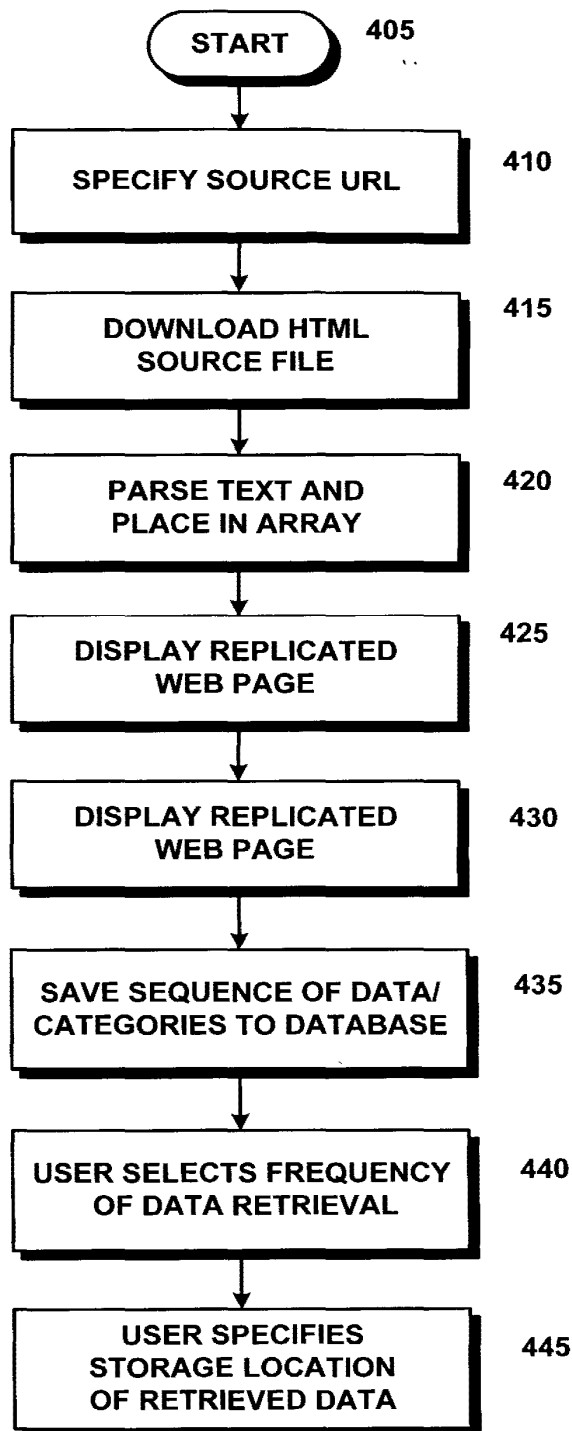
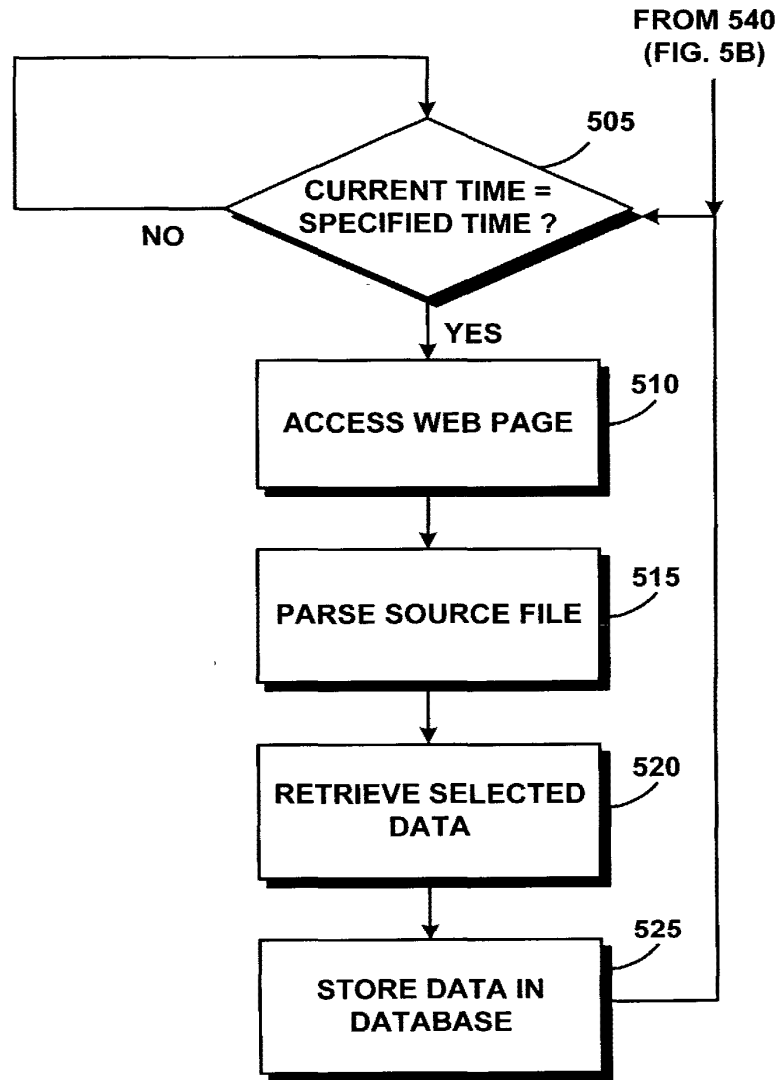


FIG. 2

**FIG. 3**

**FIG. 4**

**FIG. 5A**

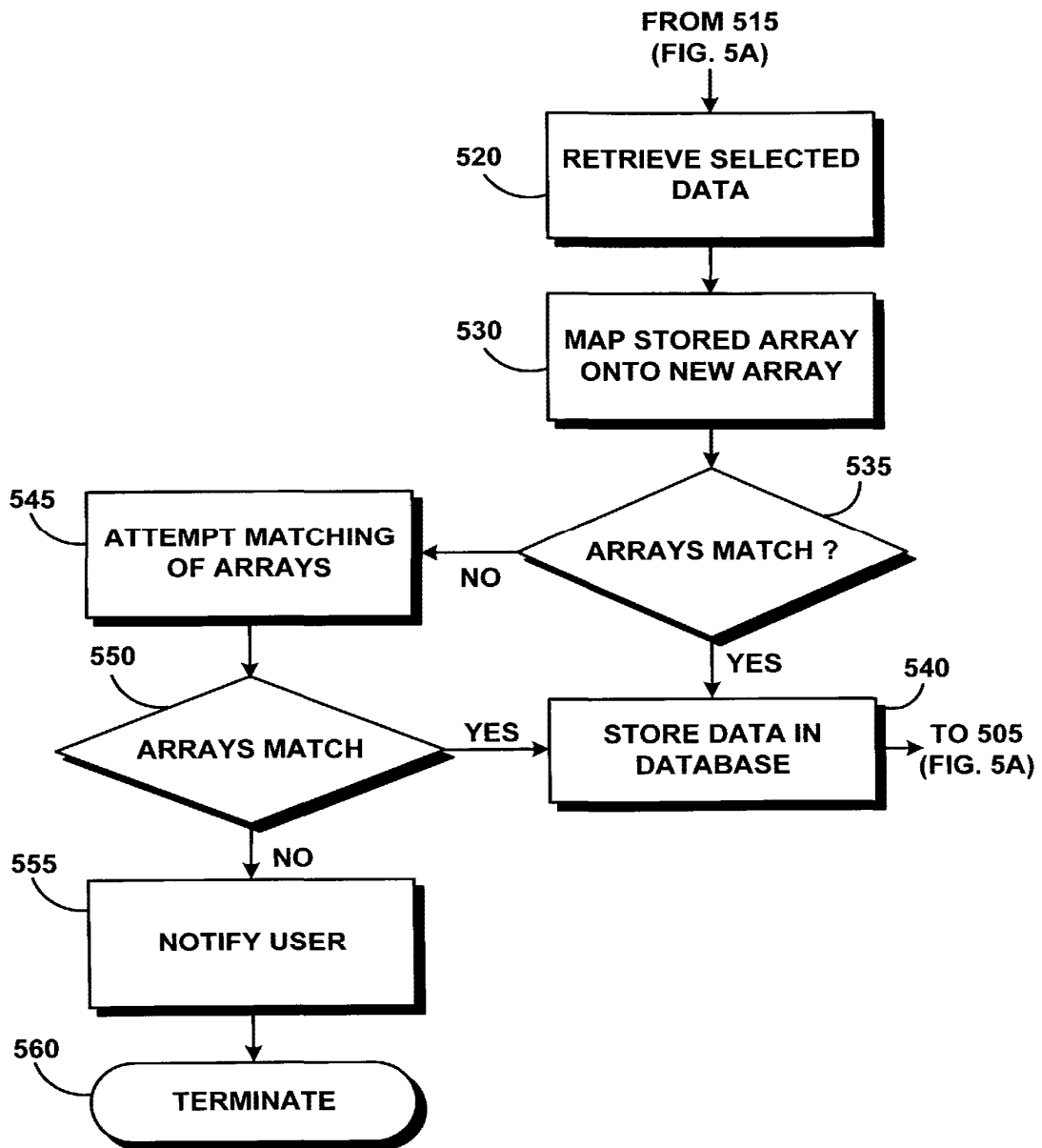


FIG. 5B

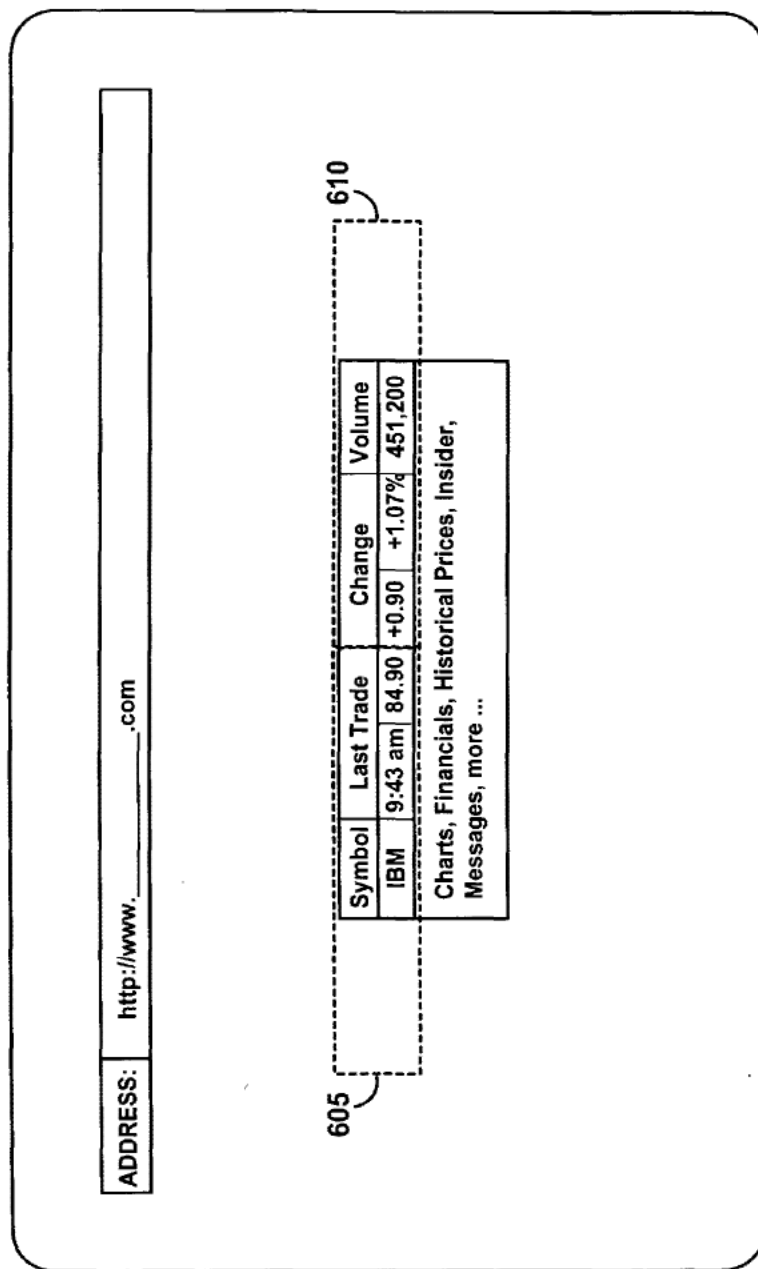


FIG. 6

10198827 . 071802

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

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 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

**“SYSTEM AND METHOD FOR DATA RETRIEVAL
AND COLLECTION IN A STRUCTURED FORMAT”**

the specification of which is attached hereto unless the following box is checked:

was filed on _____

as United States Application Number or PCT International Application Number

and was amended on _____ (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 37 CFR §1.56.

I hereby claim foreign priority benefits under 35 USC §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, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s):

Priority Not Claimed

X

(Number)

(Country)

(Day/Month/Year Filed)

I hereby claim the benefit under 35 USC §119(e) of any United States provisional application(s) listed below:

Provisional Application(s):

(Application Number)

(Filing Date)

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

(Application Number)

(Filing Date)

(Status - patented, pending, abandoned)

Power of Attorney:

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Romualdas Strimaitis, Reg. No. 35,697

Prentiss W. Johnson, Reg. No. 33,123

Ingrid M. Foerster, Reg. No. 36,511

Gregory M. Plow, Reg. No. 43,005

Farrokh E. Pourmirzaie, Reg. No. 45,297

Christine H. Smith, Reg. No. 43,133

Richard M. Goldman, Reg. No. 25,585

Christopher A. Hughes, Reg. No. 26,914

John E. Hoel, Reg. No. 26,279

Stanley B. Green, Reg. No. 24,351

Joseph C. Redmond, Jr., Reg. No. 18,753

Samuel A. Kassatly, Reg. No. 32,247

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

Address all telephone calls to:

Samuel A. Kassatly

(408) 323-5111

Address all correspondence to:

Samuel A. Kassatly

6819 Trinidad Drive

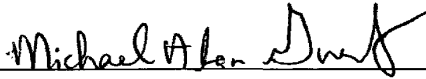
San Jose, California 95120

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.

Full name of sole or first inventor:

GUEST, Michael Alan

Inventor's signature



Date

7-9-02

Residence:

3641 Copperfield Dr. #391

San Jose, CA 95136

Citizenship: USA

Mailing Address: Same

Full name of second inventor:

SHARP, Frederick Thomas

Inventor's signature

Date:

Residence:

440 Gilbert Avenue

Menlo Park, CA 94025-2926

Citizenship: USA

Mailing Address: Same

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

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 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

**“SYSTEM AND METHOD FOR DATA RETRIEVAL
AND COLLECTION IN A STRUCTURED FORMAT”**

the specification of which is attached hereto unless the following box is checked:

was filed on _____

as United States Application Number or PCT International Application Number

and was amended on _____ (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 37 CFR §1.56.

I hereby claim foreign priority benefits under 35 USC §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, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s):

Priority Not Claimed

X

(Number)

(Country)

(Day/Month/Year Filed)

I hereby claim the benefit under 35 USC §119(e) of any United States provisional application(s) listed below:

Provisional Application(s):

(Application Number)

(Filing Date)

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

(Application Number)

(Filing Date)

(Status - patented, pending, abandoned)

Power of Attorney:

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Romualdas Strimaitis, Reg. No. 35,697

Prentiss W. Johnson, Reg. No. 33,123

Ingrid M. Foerster, Reg. No. 36,511

Gregory M. Plow, Reg. No. 43,005

Farrokh E. Pourmirzaie, Reg. No. 45,297

Christine H. Smith, Reg. No. 43,133

Richard M. Goldman, Reg. No. 25,585

Christopher A. Hughes, Reg. No. 26,914

John E. Hoel, Reg. No. 26,279

Stanley B. Green, Reg. No. 24,351

Joseph C. Redmond, Jr., Reg. No. 18,753

Samuel A. Kassatly, Reg. No. 32,247

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

Address all telephone calls to:

Samuel A. Kassatly

(408) 323-5111

Address all correspondence to:

Samuel A. Kassatly

6819 Trinidad Drive

San Jose, California 95120

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.

Full name of sole or first inventor:

GUEST, Michael Alan

Inventor's signature:

Date

Residence:

3641 Copperfield Dr. #391
San Jose, CA 95136

Citizenship USA

Mailing Address: Same

Full name of second inventor:

SHARP, Frederick Thomas

Inventor's signature:

Frederick T. Sharp

Date

8 July 2002

Residence:

440 Gilbert Avenue
Menlo Park, CA 94025-2926

Citizenship: USA

Mailing Address: Same

PATENT APPLICATION SERIAL NO. _____

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

07/24/2002 HDEMESS1 00000006 500219 10198827

01 FC:101	740.00 CH
02 FC:103	360.00 CH

PTO-1556
(5/87)

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 2001

Application or Docket Number

10198827

CLAIMS AS FILED - PART I

(Column 1)

(Column 2)

SMALL ENTITY
TYPE ☐OR OTHER THAN
SMALL ENTITY

TOTAL CLAIMS	40	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	40 minus 20 = *	20
INDEPENDENT CLAIMS	3 minus 3 = *	0
MULTIPLE DEPENDENT CLAIM PRESENT ☐		

RATE	FEE
BASIC FEE	370.00
X\$ 9=	
X42=	
+140=	
TOTAL	

RATE	FEE
BASIC FEE	740.00
X\$18=	360.00
X84=	
+280=	
TOTAL	1100.00

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

CLAIMS AS AMENDED - PART II

(Column 1)

(Column 2)

(Column 3)

SMALL ENTITY

OR OTHER THAN
SMALL ENTITY

AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	*	Minus	**
	Independent	*	Minus	***
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

RATE	ADDITIONAL FEE
X\$ 9=	
X42=	
+140=	
TOTAL	

RATE	ADDITIONAL FEE
X\$18=	
X84=	
+280=	
TOTAL	

AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	*	Minus	**
	Independent	*	Minus	***
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

RATE	ADDITIONAL FEE
X\$ 9=	
X42=	
+140=	
TOTAL	

RATE	ADDITIONAL FEE
X\$18=	
X84=	
+280=	
TOTAL	

AMENDMENT C	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	*	Minus	**
	Independent	*	Minus	***
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

RATE	ADDITIONAL FEE
X\$ 9=	
X42=	
+140=	
TOTAL	

RATE	ADDITIONAL FEE
X\$18=	
X84=	
+280=	
TOTAL	

* 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.

#2

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Title: "System and Method for Data Retrieval and Collection in a Structured Format"

Applicants: GUEST, Michael Alan et al.

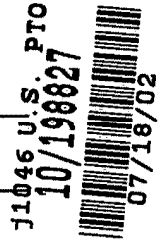
Attorney Docket No.: SVL920020050US1

Serial No.: 10/

Filed: July 18, 2002

Examiner:

Art Unit:



**Box Patent application
Assistant Commissioner
for Patents
Washington, D.C. 20231**

**INFORMATION DISCLOSURE STATEMENT
WITHIN THREE MONTHS OF FILING OR BEFORE
MAILING OF FIRST OFFICE ACTION (37 CFR 1.97(b))**

Sir:

Applicants submit herewith Form PTO-1449A listing documents believed relevant to the subject application. It is respectfully requested that these documents be made of record and an initialed copy of Form PTO-1449A be returned to the undersigned.

This Disclosure Submission Statement should not to be construed as a representation that a search has been made, that additional material to the examination of this application does not exist, or that the citations constitute prior art.

It is believed that this Disclosure Submission Statement complies with the requirements of 37 CFR 1.56, 1.97, and 1.98, and the Manual of Patent Examining Procedures, Sections 609 and 707.05. If for any reason the Examiner

SVL920020050US1

-1-

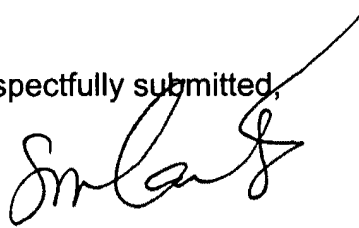
considers otherwise, it is respectfully requested that the undersigned be called so that any deficiencies can be remedied.

A copy of each document is enclosed unless indicated otherwise. Some of the documents may have markings on them. No significance is meant to be attached to the markings.

Date: July 18, 2002

Samuel A. Kassatly Law Office
6819 Trinidad Drive
San Jose, California 95120
Tel. (408) 323-5111
Fax: (408) 323-5112

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Sam Kassatly', written over a horizontal line.

Samuel A. Kassatly
Attorney for Applicants
Reg. No. 32,247

Form PTO-1449 (modified)	Attorney Docket No. SVL920020050US1	Serial No. 11046 U.S. PTO 10/198827 07/18/02
LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT(S)' INFORMATION DISCLOSURE STATEMENT	Applicant(s): GUEST, Michael et al.	
<i>(Use several sheets if necessary)</i>	Filing Date: July 18, 2002	Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initials	Document No.	Date	Name	Class / Subclass	Filing Date
	5,848,413	12.08.1998	Wolf	707/10	01.13.1995
	5,893,127	04.06.1999	Tyan et al.	707/513	11.18.1996
	5,974,430	10.26.1999	Mutschler, III et al.	707/505	09.30.1997
	6,078,924	06.20.2000	Ainsbury et al.	707/101	01.30.1998
	6,128,624	10.03.2000	Papierniak et al.	707/104	11.12.1997
	6,141,666	10.30.2000	Tobin	707/513	01.21.1997
	6,327,589	12.04.2001	Blewett et al.	707/5	06.24.1998
	6,356,899	03.12.2002	Chakrabarti et al.	707/5	03.03.1999
	2001/0042089A1	11.15.2001	Tobin	709/1	02.12.2001
	2001/0047356A1	11.29.2001	Ren et al.	707/9	04.12.2001
	2001/0018707A1	08.30.2001	Seta	709/218	02.27.2001
	2001/0007991A1	07.12.2001	Tobin	709/203	02.13.2001
	2002/0010682A1	01.24.2002	Johnson	705/59	07.19.2001

Examiner**Date Considered**

Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant(s).

Form PTO-1449 (modified)	Attorney Docket No. SVL920020050US1	Serial No.
LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT(S)' INFORMATION DISCLOSURE STATEMENT	Applicant(s): GUEST, Michael et al.	
(Use several sheets if necessary)	Filing Date: July 18, 2002	Group Art Unit:

FOREIGN PATENT DOCUMENTS

Examiner Initials	Document No.	Publication Date	Country	Int'l Class Subclass	Translation	
					Yes	No
	WO 01/75673A2	10.11.2001	WIPO/PCT	G06F 17/30	No	
	WO 01/24046A2	04.05.2001	WIPO/PCT	G06F 17/30	No	
	2,250,663	14.19.2000	Canada	G06F 17/30	No	

Examiner**Date Considered**

Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance *and* not considered. Include copy of this form with next communication to applicant(s).

Form PTO-1449 (modified)	Attorney Docket No. SVL920020050US1	Serial No.
LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT(S)' INFORMATION DISCLOSURE STATEMENT	Applicant(s): GUEST, Michael et al.	
<i>(Use several sheets if necessary)</i>	Filing Date: July 18, 2002	Group Art Unit:

PUBLICATIONS / OTHER ART

Examiner Initials	Citation
	F. Esposito et al., "A Machine Learning Approach to Web Mining," AI*IA 99: Advances in Artificial Intelligence, 6 th Congress of Italian Association for Artificial Intelligence, September 14-17, 1999, pages 190-201..
	T. Boyd et al., "The Active Guidebook - Information Retrieval by Keyword and Location," IFIP TC6 / WG6.1 Third International Working Conference on Distributed Applications and Interoperable Systems, 2001, pages 55-66.
	H. Kim et al., "Building an XML and Web-Based Document Retrieval System," 20 th National Online Meeting, Proceedings 1999, May 18-20, 1999, pages 251 - 262.
	Seung-Jin Lim et al., "An Automated Approach for Retrieving Hierarchical Data from HTML Tables," CIKM'99, 1999, pages 466 - 474.
	Sangho Ha et al., "Mapping XML Documents to the Object-Relational Form," ISIE 2001, pages 1757-1761.
	Michael Cutler et al., "Using the Structure of HTML Documents to Improve Retrieval," Proceedings of the USENIX Symposium on Internet Technologies and Systems, 1997, .

Examiner**Date Considered**

Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance *and* not considered. Include copy of this form with next communication to applicant(s).

Form PTO-1449 (modified)	Attorney Docket No. SVL920020050US1	Serial No.
LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT(S) INFORMATION DISCLOSURE STATEMENT	Applicant(s): GUEST, Michael et al.	
<i>(Use several sheets if necessary)</i>	Filing Date: July 18, 2002	Group Art Unit:

PUBLICATIONS / OTHER ART

Examiner Initials	Citation
	Tao Guan et al., "KPS --- A Web Information Mining Algorithm," downloaded from WWW URL http://www8.org/w8-papers/4a-search-mining/kps/kps.html , on March 15, 2002.
	D. Taniar et al., "Structured Web Pages Management for Efficient Data Retrieval," IEEE 2000, pages 97-104.
	F. Ferri et al., "Toward a Retrieval of HTML Documents Using a Semantic Approach," IEEE 2000, pages 1571-1574.
	H. Kinutani et al., "Identifying Result Subdocuments of XML Search Conditions," IEEE 2001, pages 254-261.
	J. Cardiff et al., "Querying Multiple Databases Dynamically on the World Wide Web," IEEE 2000, pages 238-245.

Examiner**Date Considered**

Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance *and* not considered. Include copy of this form with next communication to applicant(s).

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
11 October 2001 (11.10.2001)

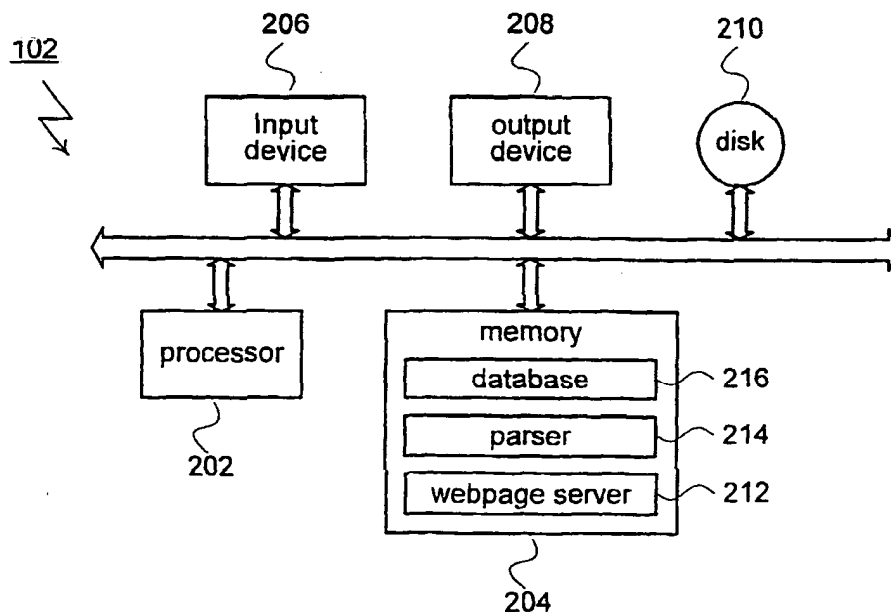
PCT

(10) International Publication Number
WO 01/75673 A2

- (51) International Patent Classification⁷: **G06F 17/30** (74) Agents: JAKOPIN, David, A. et al.; Pillsbury Winthrop LLP, 1600 Tysons Boulevard, McLean, VA 2201 (US).
- (21) International Application Number: PCT/US01/10244
- (22) International Filing Date: 30 March 2001 (30.03.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
09/540,244 31 March 2000 (31.03.2000) US
- (71) Applicant: MYPOINTS.COM, INC. [US/US]; 12th floor, 100 California Street, San Francisco, CA 94111 (US).
- (72) Inventors: BISTRICEANU, Virgil; 10 West 31st Street, Chicago, IL 60616 (US). MINOGUE, Matthew; 1418 West Division Street, Chicago, IL 60622 (US). KNOWLEDEN, Justin; One Bloomingdale Place #511, Bloomingdale, IL 60108 (US).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:
— without international search report and to be republished upon receipt of that report

[Continued on next page]

(54) Title: METHOD FOR DEVELOPING AND MAINTAINING CONTENT FOR WEBSITES



(57) Abstract: An object-oriented method for developing and maintaining content for websites is provided. For a representative embodiment, each web page within a website is defined using a hierarchy of objects. Each object is a snippet of HTML code and corresponds to a particular entity such as a header, navigation bar, content or footer. Objects may contain dynamic tags. Tags may be used to include per-user data. Dynamic tags may also be used to include other objects. The parser constructs web pages by recursively parsing the object from which they are constructed. The parser replaces dynamic tags with appropriate values during the recursive parsing process.

BEST AVAILABLE COPY



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

METHOD FOR DEVELOPING AND MAINTAINING CONTENT FOR WEBSITES

Field Of The Invention

5 The present invention is generally related to software engineering for computer networks. More specifically, the present invention relates to methods for simplifying the design, implementation and maintenance of website software and content.

Background Of The Invention

10 Techniques for making software simple to design and maintain are important to any software development process. This is especially true where software is being developed for use as part of a website. Projects of this type tend to be highly artistic, creative and content driven. Website development can also be somewhat informal and is often performed by a mixture of technical and non-technical contributors.

15 In most cases, changing website content requires a multi-step process. First the site source code is checked out from a software repository. The source code is then modified as to reflect the content change. When required, the source code is then re-compiled. Once these steps are complete, the source code can be checked back into the repository.

20 This multi-step process has certain drawbacks when used for website development. Arguable the most significant of these is the need for skilled software engineers to oversee the process. This is true even where the change is limited largely or completely to content, without any change to website function. The multi-step process is also relatively inflexible and not ideally suited for piecemeal changes.

25 For these and other reasons, a need exists for new methods for managing website content. It is important that these methods allow website content to be developed and changed without assistance from software engineers. This need is particularly relevant where website content is subject to frequent or on-going change. It is an object of the present invention to provide a method that addresses these needs.

Summary Of The Invention

30 An embodiment of the present invention provides a method for developing and maintaining content for websites. For a representative embodiment, each web page within a website is defined as a top-level object. Each top-level object may be recursively defined in terms of lower-level objects. The top-level and lower-level objects for the entire website are preferably stored in a database.

Each object is a snippet of HTML code and corresponds to a particular entity such as a header, navigation bar, content or footer. Objects may contain dynamic tags. Each dynamic tag is a placeholder for a particular value. In some cases, dynamic tags are placeholders for per-user data (such as a user's name or address). Dynamic tags can also be placeholders for other objects.

5 The software for the website includes a parser. The parser is invoked each time a web page is requested. Once invoked, the parser recursively parses the top-level object for the requested web page as well as all included lower level objects. As the parser processes each object, it replaces each instance of a dynamic tag with the tag's corresponding value.

10 Stated differently, an embodiment of the present invention includes a method for providing web pages, the method comprising the steps of: defining each web page as a corresponding object hierarchy, receiving a request for a specified web page, recursively parsing the corresponding object hierarchy to construct the specified web page; and returning the constructed web page.

15 Advantages of the invention will be set forth, in part, in the description that follows and, in part, will be understood by those skilled in the art from the description herein. The advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims and equivalents.

Brief Description Of The Drawings

20 The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Figure 1 is a block diagram of a computer network shown as an exemplary environment for an embodiment of the present invention.

Figure 2 is a block diagram of a computer or terminal as used in the network of Figure 1.

Figure 4 is a block diagram showing the components of a sample web page.

25 Figure 3 is a block diagram showing an object hierarchy that corresponds to the sample web page of Figure 3.

Detailed Description Of The Preferred Embodiments

30 Reference will now be made in detail to preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever convenient, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

In Figure 1, a computer network 100 is shown as a representative environment for an embodiment of the present invention. Computer network 100 is intended to be representative of the

complete spectrum of computer network types including Internet and internet-like networks. Computer network 100 includes a number of computers, of which computers 102a through 102e are representative. Computers 102 are intended to be representative of the entire spectrum of devices that may be interconnected in computer networks. Computers 102 may also be television set top boxes, such as the WebTV system. Computers 102 may also be handheld devices such as personal data assistants or cellular telephones. Numerous other consumer devices are also possible.

Figure 2 shows a representative implementation for computers 102. Structurally, each computer 102 includes a processor, or processors 202, and a memory 204. Processor 202 can be selected from a wide range of commercially available or custom types. An input device 206 and an output device 208 are connected to processor 202 and memory 204. Input device 206 and output device 208 represent all types of I/O devices such as disk drives, keyboards, modems, network adapters, printers and displays. Each computer 102 may also include a disk drive 210 of any suitable disk drive type (equivalently, disk drive 210 may be any non-volatile mass storage system such as "flash" memory).

An embodiment of the present invention provides an object-oriented method for developing and maintaining content for websites. For the purposes of illustration, it is assumed that one of computer systems 102 is a web server and has been configured to use the method of the present invention. For that computer system 102, memory 204 includes a web server program 212, a parser program 214 and a database 216. Web server program 212 accepts requests for web pages from users of network 100. Web server program 212 responds to these requests by invoking parser program 214 to construct appropriate web pages using database 216.

Figure 2 shows web server program 212, parser program 214 and database 216 as separate entities. This particular configuration is intended to be representative in nature and other configurations are equally practical. Parser program 214 may be, for example, part of web server program 212. Parser program 214 may also be part of the software associated with database 216 or part of a middle tier (application server) positioned between web server program 212 and database 216.

Figure 2 also shows web server program 212, parser program 214 and database 216 to be resident in memory 204. It should be understood that web server program 212, parser program 214 and database 216 may be fully or partially resident in memory 204. Under certain circumstances, for example web server program 212, parser program 214 and database 216 may be fully or partially located on disk drive 210. Web server program 212, parser program 214 and database 216 may also be remotely located on two or more computers 102.

Construction of web pages by parser program 214 is better understood in terms of the simplified, representative web page 300 of Figure 3. Web page 300 includes a navigation bar 302, an

image 304, a footer 306 and a welcome banner 308. Navigation bar 302 is subdivided into buttons 310a through 310e for home, next, previous, last and first operations. Footer 304 is subdivided into button 312a showing a current page number and button 312b showing a copyright information operation.

5 Within database 216, each web page (i.e., each entity that has it's own URL) is stored as a top-level object. Each top-level object is recursively defined in terms of zero or more lower-level objects. In the case of web page 300, the top-level object includes lower-level objects for navigation bar 302, image 304, footer 306 and welcome banner 308. Each lower-level object may also be recursively defined in terms of still lower-level objects. This is true for navigation bar 302 which is
10 defined in terms of buttons 310a through 310e. Footer 306 is also defined in terms of lower-level objects (buttons 312a and 312b). The deconstruction of objects into lower-level objects is subject to infinite variation.

A hierarchy of objects corresponding to web page 300 is shown in Figure 4.

Each object (i.e., each top-level object and each lower-level object) is composed of a snippet
15 of HTML code. The HTML code for each object may include any number of dynamic tags. Each dynamic tag is a placeholder for a particular value and gets replaced by that value during page construction. Depending on the particular embodiment, dynamic tags can be used to represent a range of different values. For one such embodiment, database 216 includes personal information describing users. This type of information would be available, for instance, in cases where users subscribe to
20 particular websites or services. For this type of embodiment, dynamic tags can be used to represent particular pieces of personal information. The following example shows how objects can be used to build page 300:

```

25       <HTML>
          <HEAD>
          <TITLE>Sample usage of dynamic tags</TITLE>
          </HEAD>
          <BODY>
          <_parseObject header.global _>
          Welcome, <_ getFirstAndLastName _>
30       <_parseObject mainPage.global _>
          <_parseObject footer.global _>
          </BODY>
      </HTML>

```

35 Within this object, the dynamic tag <_getFirstAndLastName _> represents the user's name (in this hypothetical case "John Q. Public") and is replaced by parser program 214 each time web page 300 is constructed. Thus, welcome banner 308 would be appear differently if the user was "Jane Doe" or "Joe Smith".

The specific type of information stored for each user, and the range of dynamic tags used to display personal information, is entirely dependent on the particular embodiment of web server program 212, parser program 214 and database 216. Typical examples include names, addresses, phone numbers, account balances and other personal data. Dynamic tags can also implement personal link counters (i.e., how many times has this user visited this page). Dynamic tags can also be used to award points associated with web page usage. A points system is described in detail in U.S. Application No. 09/602,168 filed June 22, 2000, entitled "Reward-based Model for ISP Service" That disclosure is incorporated in this document by reference.

The use of personal information implies that web server program 212, parser program 214 and database 216 have the ability to recognize individual users. This can be accomplished using several methods. For one method, identity information is extracted from cookies supplied by the users' browser programs. For another method, users explicitly identify themselves by logging in.

Dynamic tags can also be used to include objects within other objects. Thus, the top level object for web page 300 could include dynamic tags for the lower level objects representing navigation bar 302, image 304 and footer 306. The lower-level object for navigation bar 302 could include dynamic tags for the objects representing buttons 310a through 310e for home, next, previous, last and first operations. Footer 304 could include dynamic tags for the objects representing buttons 312a and 312b. In the sample code fragment shown above, the dynamic tag `<_ parseObject footer.global _>` tells the parser to include (recursively) another object. The argument following `parseObject` tells the parser which object to include (in this case `footer.global`).

Each top-level and lower-level object is preferably stored in database 216. When parser program 214 receives a request for a web page (i.e., receives a request for a specific URL) it retrieves the corresponding top-level object from database 216. Starting with that top-level object, parser program 214 begins a recursive parsing process to render the requested web page. During this process, parser program 214 replaces each instance of a dynamic tag with its corresponding value. For dynamic tags representing personal values, parser program 214 retrieves the corresponding information (e.g., first name, last name, address, account balance, etc.) from database 216. For other dynamic tags, parser program 214 may consults appropriate resources (which may include database 216) to perform the appropriate resolution.

Dynamic tags that correspond to other lower-level objects cause parser program 214 to retrieve the corresponding lower-level objects from database 216. Parser program 214 then continues the recursive parsing process with the retrieved objects. In this way, the preferred parsing method resembles a depth first traversal of the top-level and lower-level objects that are included in the requested web page. It should be appreciated, however, that the depth first search order is preferred but not required. Other search orders are also possible.

Objects in database 216 may be configured and maintained using a range of methods. For one embodiment, objects in database 216 are maintained using a database maintenance interface. The maintenance interface controls the way in which users access the top-level and lower-level objects in database 216. All objects are normally provided to users on a read-only basis. To modify a particular object, a user locks the object using the maintenance interface. The maintenance interface then provides the user with read/write access to the object. This prevents simultaneous conflicting modifications to the same object by different users.

The maintenance interface is preferably configured to maintain revision control information for each object. The revision control information tracks the change history of each object—showing how each object has been modified and which user made the modification.

The maintenance interface is preferably configured support building and preview of both top-level and lower-level objects. This allows developers to see the results of any modifications they have made.

The method of the present invention effectively minimizes the need for technically skilled software developers. In practice, software development is limited to the creation of modification of dynamic tags (i.e. to creating or modifying the software that retrieves from database 216 user-specific information or performs actions in database 216 related to user's actions on the web site). Content developers, on the other hand, are free to build content that includes dynamic tags without ever having to be concerned with the details of the tags' implementations. In this way, the roles of software developer and content developer are largely separated.

Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope of the invention being indicated by the following claims and equivalents.

WHAT IS CLAIMED IS:

1. A method for providing a web page, the method comprising the steps of:
receiving a request for the web page;
5 retrieving a top-level object associated with the web page;
parsing the top-level object to construct the web page;
returning the constructed web page.
2. A method as recited in claim 1 that further comprises the steps of:
retrieving a lower-level object included in the top-level object; and
10 parsing the lower-level object to construct the web page.
3. A method as recited in claim 2 that further comprises the steps of:
detecting a dynamic tag in the top-level or lower-level objects;
retrieving per-user information corresponding to dynamic tag; and
inserting the per-user information in place of the dynamic tag in the top-level or
15 lower-level object.
4. A method as recited in claim 3 that further comprises the step of identifying the user using a
cookie supplied by the user's browser.
5. A method as recited in claim 2 that further comprises the steps of:
detecting a dynamic tag in the top-level or lower-level objects;
20 using the dynamic tag as a link to the lower-level object; and
inserting the lower-level object in place of the dynamic tag in the top-level object.
6. A method as recited in claim 2 that further comprises the steps of:
detecting a dynamic tag in the top-level or lower-level objects;
using the dynamic tag as a link to a second lower-level object; and

8

inserting the second lower-level object in place of the dynamic tag in the top-level or lower-level object.

7. A method for providing web pages, the method comprising the steps of:
- 5 defining each web page as a corresponding object hierarchy;
- receiving a request for a specified web page;
- recursively parsing the corresponding object hierarchy to construct the specified web page; and
- returning the constructed web page.
8. A method as recited in claim 7 wherein each object is a snippet of HTML code.
- 10 9. A method as recited in claim 7 wherein each object may include one or more dynamic tags, each dynamic tag corresponding to a particular piece of per-user information, and wherein the step of recursively parsing the object hierarchy further comprises the step of replacing each dynamic tag with its corresponding per-user information.
- 15 10. A method as recited in claim 7 wherein each object may include one or more dynamic tags, each dynamic tag acting as a link to a lower-level object, and wherein the step of recursively parsing the object hierarchy further comprises the step of replacing each dynamic tag with the corresponding lower-level object.
- 20 11. A data storage medium having machine-readable code stored thereon, the machine-readable code comprising instructions executable by an array of logic elements, the instructions defining a method for providing a web page, the method comprising:
- receiving a request for the web page;
- retrieving a top-level object associated with the web page;
- parsing the top-level object to construct the web page;
- returning the constructed web page.

12. A data storage medium as recited in claim 11 wherein the method further comprises the steps of:
- retrieving a lower-level object included in the top-level object; and
- parsing the lower-level object to construct the web page.
- 5 13. A data storage medium as recited in claim 12 wherein the method further comprises the steps of:
- detecting a dynamic tag in the top-level or lower-level objects;
- retrieving per-user information corresponding to dynamic tag; and
- inserting the per-user information in place of the dynamic tag in the top-level or
- 10 lower-level object.
14. A data storage medium as recited in claim 13 that further comprises the step of identifying the user using a cookie supplied by the user's browser.
15. A data storage medium as recited in claim 12 wherein the method further comprises the steps of:
- 15 detecting a dynamic tag in the top-level or lower-level objects;
- using the dynamic tag as a link to the lower-level object; and
- inserting the lower-level object in place of the dynamic tag in the top-level object.
16. A data storage medium as recited in claim 12 wherein the method further comprises the steps of:
- 20 detecting a dynamic tag in the top-level or lower-level objects;
- using the dynamic tag as a link to a second lower-level object; and
- inserting the second lower-level object in place of the dynamic tag in the top-level or
- lower-level object.
17. A system for providing a web page, comprising:

10

a memory device; and

a processor in communication with said memory device, said processor configured to perform a method that comprises the steps of:

- 5
- receiving a request for the web page;
 - retrieving a top-level object associated with the web page;
 - parsing the top-level object to construct the web page;
 - returning the constructed web page.

18. A system as recited in claim 17 that further comprises the step of identifying the user using a cookie supplied by the user's browser.

- 10
19. A system as recited in claim 17 wherein the method further comprises the steps of:
- detecting a dynamic tag in the top-level or lower-level objects;
 - using the dynamic tag as a link to the lower-level object; and
 - inserting the lower-level object in place of the dynamic tag in the top-level object.

- 20.
20. A system as recited in claim 17 wherein the method further comprises the steps of:
- 15
- detecting a dynamic tag in the top-level or lower-level objects;
 - using the dynamic tag as a link to a second lower-level object; and
 - inserting the second lower-level object in place of the dynamic tag in the top-level or lower-level object.

21. A system for providing a web page that comprises:
- 20
- means for receiving a request for the web page;
 - means for retrieving a top-level object associated with the web page;
 - means for parsing the top-level object to construct the web page;
 - means for returning the constructed web page.

22. A system as recited in claim 21 that further comprises:
- 25
- means for retrieving a lower-level object included in the top-level object; and

means for parsing the lower-level object to construct the web page.

23. A system as recited in claim 22 that further comprises:

means for detecting a dynamic tag in the top-level or lower-level objects;

means for retrieving per-user information corresponding to dynamic tag; and

5 means for inserting the per-user information in place of the dynamic tag in the top-level or lower-level object.

24. A system as recited in claim 23 that further comprises means for identifying the user using a cookie supplied by the user's browser.

25. A system as recited in claim 22 that further comprises:

10 means for detecting a dynamic tag in the top-level or lower-level objects;

means for using the dynamic tag as a link to the lower-level object; and

means for inserting the lower-level object in place of the dynamic tag in the top-level object.

26. A system as recited in claim 22 that further comprises:

15 means for detecting a dynamic tag in the top-level or lower-level objects;

means for using the dynamic tag as a link to a second lower-level object; and

means for inserting the second lower-level object in place of the dynamic tag in the top-level or lower-level object.

Fig. 1

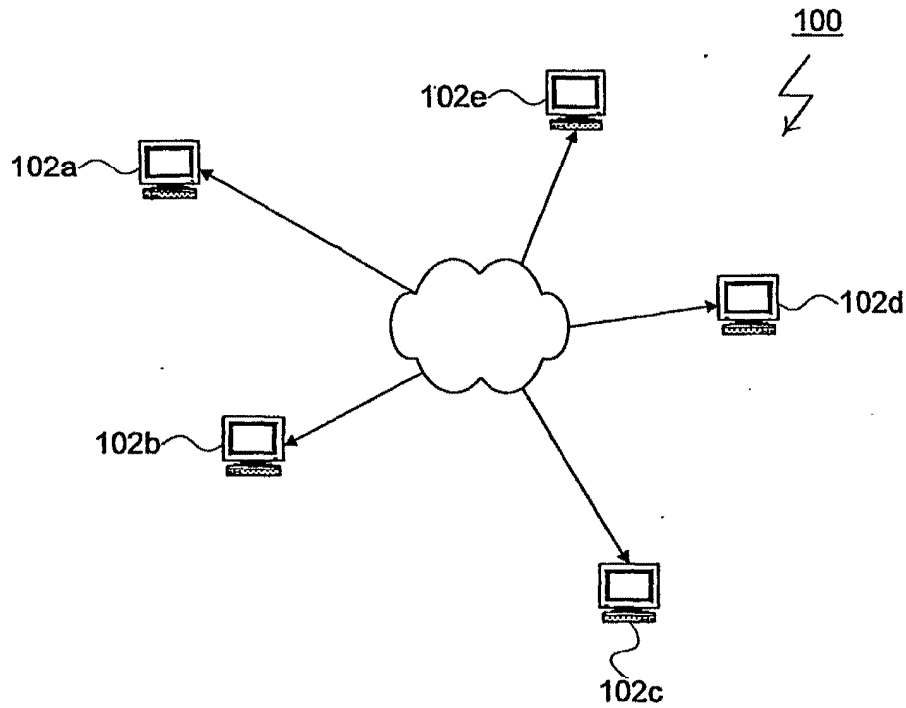
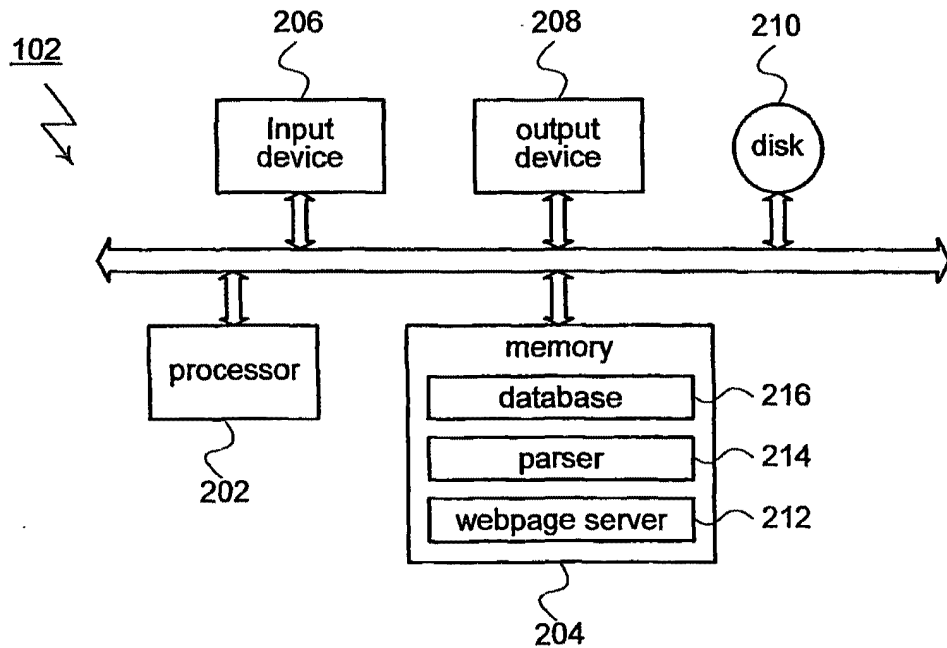


Fig. 2



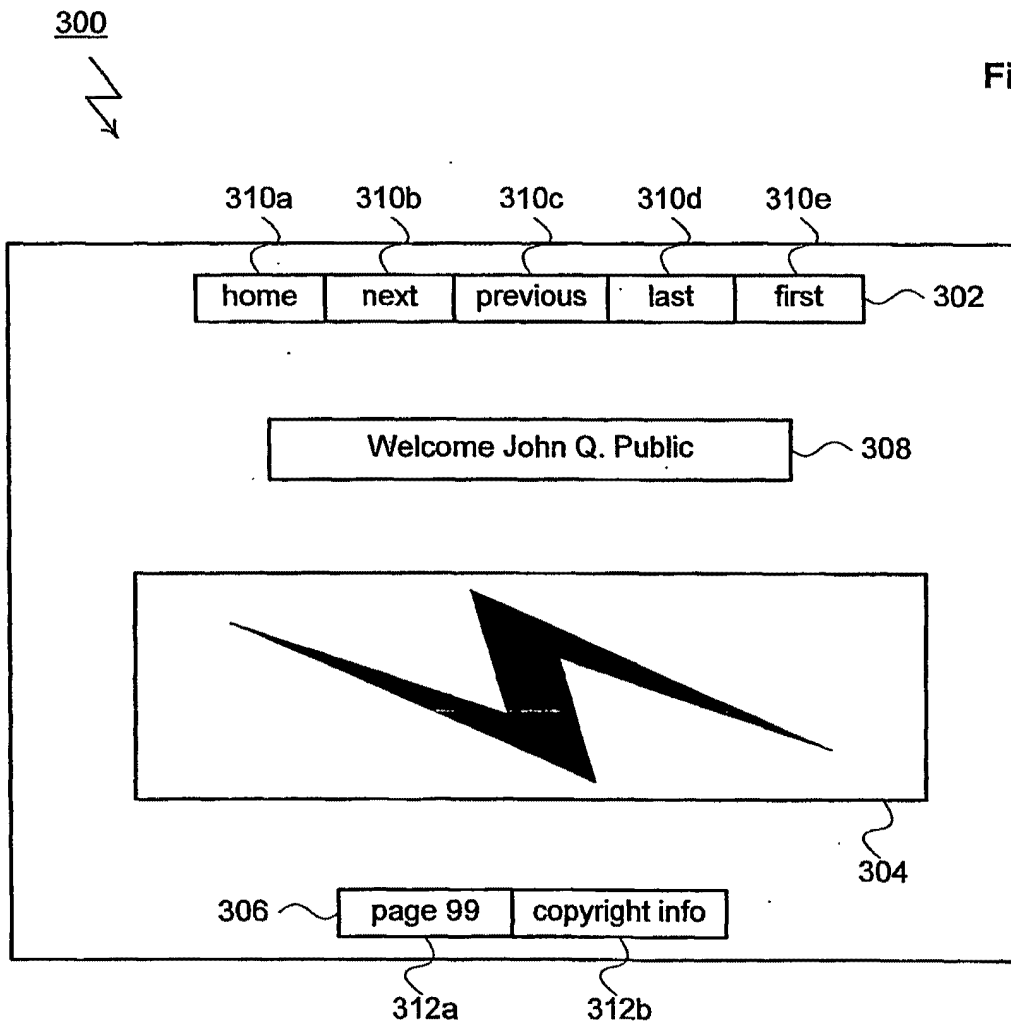
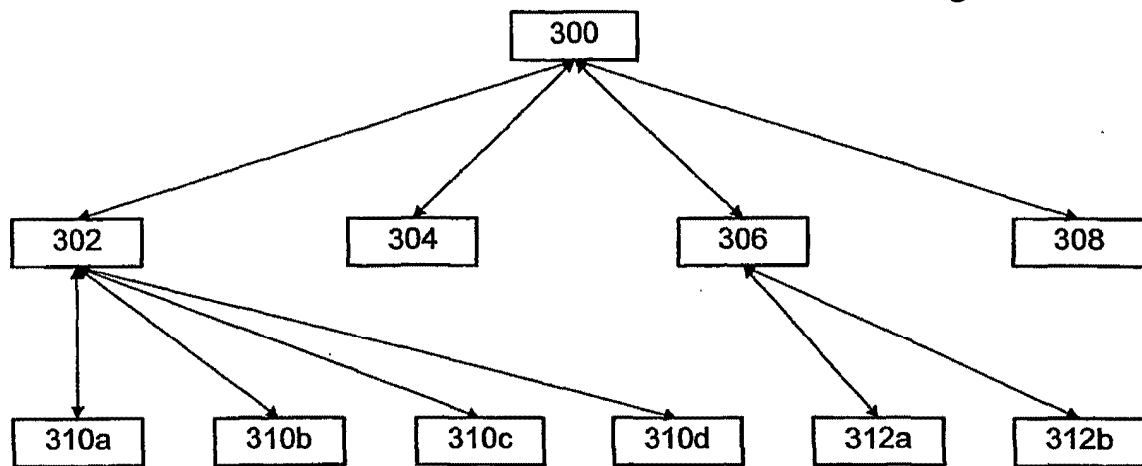


Fig. 3

Fig. 4



(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
5 April 2001 (05.04.2001)

PCT

(10) International Publication Number
WO 01/24046 A2

(51) International Patent Classification⁷: G06F 17/30

(21) International Application Number: PCT/CA00/00861

(22) International Filing Date: 21 July 2000 (21.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/407,336 29 September 1999 (29.09.1999) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 09/407,336 (CON)
Filed on 29 September 1999 (29.09.1999)

(71) Applicant (for all designated States except US): XML-GLOBAL TECHNOLOGIES, INC. [US/US]; 701 Dexter Avenue, Suite 420, Seattle, WA 98109 (US).

(72) Inventors; and

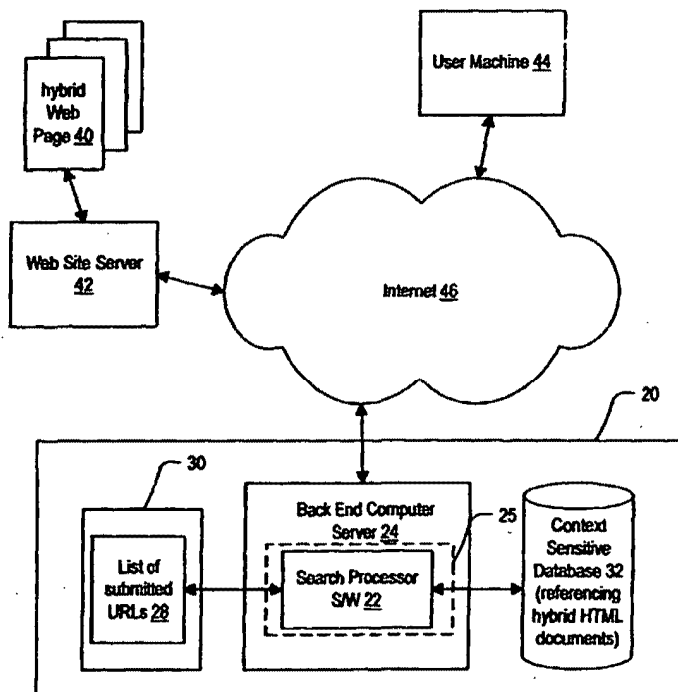
(75) Inventors/Applicants (for US only): NICKULL, Duane, Allan [CA/CA]; 839 Grover Avenue, Coquitlam, British Columbia V3J 3E1 (CA). MACKENZIE, Chad, Matthew [CA/CA]; 1238 Richards Street, Suite 1806, Vancouver, British Columbia V6B 3G2 (CA). HOGLUND, Jamie, Michael, Thomas [US/CA]; 1516 Davie Street, Vancouver, British Columbia V6B 2V6 (CA).

(74) Agents: KINGWELL, Brian, G. et al.; Smart & Biggar, Box 11560, Vancouver Centre, 650 West Georgia Street, Suite 2200, Vancouver, British Columbia V6B 4N8 (CA).

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

[Continued on next page]

(54) Title: AUTHORING, ALTERING, INDEXING, STORING AND RETRIEVING ELECTRONIC DOCUMENTS EMBEDDED WITH CONTEXTUAL MARKUP



(57) Abstract: The invention provides a method, computer-readable instructions and a system for generating and indexing context sensitive HTML documents. A context sensitive HTML document is generated by inserting an opening contextual markup tag before an item of character data (or other computer-readable data) within an HTML document and by inserting a closing contextual markup tag after the item of character data. A predefined prefix for identifying contextual information is included in both the opening and closing contextual markup tags. At least one contextual term identifying a context within which the item of character data is used is also included in both the opening and closing contextual markup tags. Each contextual markup tag is marked with HTML delimiters. Context sensitive HTML documents generated in this way form hybrid HTML documents which remain compatible with HTML and which can be processed and incorporated into a context sensitive database that may be searched by users. A hybrid HTML document is processed by scanning it in search of character data marked with the contextual markup tags. The character data and associated contextual terms are retrieved from the hybrid HTML

documents and added to a context sensitive database. When the character data is added to the database, the associated contextual terms and the character data are linked, and the character data is linked to an address identifying a location of the hybrid HTML document from which originated the character data and associated contextual terms.

BEST AVAILABLE COPY

Bright Data Ltd. v. Metacluster LT UAB

IPR2023-01442, EX. 1002

209 of 318



(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— Without international search report and to be republished upon receipt of that report.

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

- 1 -

**AUTHORING, ALTERING, INDEXING,
STORING AND RETRIEVING ELECTRONIC DOCUMENTS
EMBEDDED WITH CONTEXTUAL MARKUP**

5

FIELD

The present invention relates generally to electronic documents, and more particularly to a method, apparatus and system for authoring, altering, indexing, storing, and
10 retrieving electronic documents embedded with contextual markup tags.

BACKGROUND

The Internet has rapidly become one of the leading communications mediums of our
15 age. One of the most popular applications used in the Internet is the World Wide Web (also referred to as the "Web" or "WWW"). Tens of thousands of Web sites around the world house millions of Web pages and related electronic documents.

Yet as the volume and diversity of information available over the Internet continues to
20 grow, the ability to locate relevant information is becoming a greater challenge. With the growth of information accessible over the Internet, search engines have become an important tool for enabling users to search for and retrieve information over the Internet that is relevant to their needs. Popular search engines for searching the Internet are available from Yahoo™ (<http://www.yahoo.com>), Infoseek™
25 (<http://www.infoseek.com>), Lycos™ (<http://lycos.cs.cmu.edu>), AltaVista™ (<http://www.altavista.com>), Excite™ (<http://www.excite.com>), Microsoft™ Network (<http://www.msn.com>) and others. These search engines, however, do not provide context sensitive search services for context sensitive data.

30 The lack of context sensitive search engines is due, in part, to the widespread use of the HyperText Markup Language (HTML) to author and publish electronic documents. HTML is derived from the well known Standard Generalized Markup

- 2 -

Language (SGML) and has been widely adopted to generate Web pages. HTML provides a set of predefined markup codes, commonly referred to as "tags", that can be inserted into or included in a text-based file (or document) that is viewable through a Web browser. The tags help define the display semantics of the text and instruct the Web browser on how to display the text.

The ease with which an author or publisher may generate, modify and publish electronic documents using HTML has contributed to the tremendous success of the Internet. However, conventional HTML does not provide a standard for including context sensitive data in electronic documents. As a result, it can be difficult, if not impossible, to build and maintain a context sensitive database based on the content of conventional HTML documents.

The contents of the databases which are used by search engines to provide users with search results are typically maintained with computer programs commonly known as "crawlers" and "spiders". A crawler is a program that can be used to autonomously explore the Internet or other networks in search of new or updated, publicly accessible resources such as Web sites, files available in FTP archives and Gopher documents. When a resource is found, it can be accessed by a spider which adds the location, identity and data from the resource to a search engine database.

When a spider accesses an identified Web page, it will typically index the Web page and add searchable content from the Web page to the search engine database. The absence of context sensitive data in conventional HTML documents limits the ability of a spider to meaningfully capture and index the context within which textual elements are written in the HTML documents. As a result, a common approach to indexing is to merely build and manage a non-contextual database. When a spider indexes a Web page, indexing will typically involve processing the HTML encoded document that embodies the Web page and separating text elements into words that are added to the database. In addition, the Web page may be indexed by adding to the database only words found in specific HTML tags, known as <META> tags, embedded in the Web page. While these <META> tags can be used to assign

- 3 -

keywords to a document as a whole, there is currently no mechanism available for assigning contextual meaning to particular data within an HTML document or for organizing such data contextually. The limitations of conventional HTML make it very difficult for a spider to produce an extensive context sensitive database using HTML documents. For instance, while the term "Washington" may be used in a document to refer to a president, a state, a city, a capital, an actor, or another individual's name, such fine distinctions cannot be accomplished using conventional HTML.

10 With the absence of rich context sensitive data in HTML documents, users are discovering that searching for relevant information is becoming a more difficult task, even with the advanced search processes supported by conventional search engine technology. A search may return thousands or even hundreds of thousands of search results. Further aggravating the problem, search results can be cluttered with
15 redundant or irrelevant information, making the retrieval of relevant information difficult.

In order to assist users in performing more directed searches, some spiders have been specialized so as to build indexes which are categorized by general topic areas. In
20 such known solutions, a specialized spider is programmed to index predefined Web sites relating to specific topic areas, such as on-line medical journals. While categorizing Web sites according to general topic areas can generally assist search engines perform more targeted searches, this approach does not produce a context sensitive database for users to search. As a result, the user is still faced with the
25 difficulty that performing a search on the topic-related index can produce many results that contain the search terms but not in the context desired by the user.

The Extensible Markup Language (XML) provides another solution to the need to include context sensitive data within electronic documents accessible over the Internet and other networks. XML was introduced in part to serve as the basis for applications
30 that permit Web authors and publishers to create XML Web pages containing structured context sensitive data. While XML is becoming more commonly used in

- 4 -

the Internet community, HTML continues to be the markup language of choice for many Web authors and publishers, and the majority of Web sites continue to be populated with HTML documents.

5 In order to leverage the use of the large amounts of information captured in HTML documents, it would be desirable to have a mechanism for adapting existing and new HTML documents so that authors may place a context on their data. It would be further desirable to be able to implement such a mechanism in a way that would not interfere with the processing of HTML codes embedded within the documents.

10

SUMMARY OF THE INVENTION

The above and related desires are addressed in the present invention by providing a novel and nonobvious method, system and computer-readable instructions for
15 authoring, altering, indexing, storing and retrieving context sensitive HTML documents (also referred to herein as hybrid HTML documents). The present invention can also be equally applied to electronic documents generated with XHTML or a context insensitive markup language that is a subset of SGML.

20 In accordance with one aspect of the invention, a computer-readable medium contains an electronic document generated with a context insensitive markup language, the electronic document having contextual markup tags and items of computer-readable data marked by the contextual markup tags. The contextual markup tags can each include a predefined prefix for identifying contextual information, at least one
25 contextual term identifying a context within which at least one of the items of computer-readable data is used within the electronic document, and HTML delimiters. The contextual markup tags are preferably arranged into pairs of contextual markup tags encapsulating the at least one of the items of computer-readable data, with each pair including an opening contextual markup tag and a closing contextual markup tag.
30 Marking items of computer-readable data within the electronic document with opening and closing contextual markup tags provides an easy mechanism for

- 5 -

generating a rich range of context sensitive data within an otherwise context insensitive document.

5 In accordance with another aspect of the invention, a context sensitive HTML document is generated by inserting an opening contextual markup tag before an item of computer-readable data within an HTML document and by inserting a closing contextual markup tag after the item of computer-readable data. At least one contextual term identifying a context within which the item of computer-readable data is used is included in both the opening and closing contextual markup tags. Each
10 contextual markup tag is marked with HTML delimiters. Preferably, a predefined prefix for identifying contextual information is also inserted in both the opening and closing contextual markup tags. Generating context sensitive HTML documents in this way with the use of the predefined prefix within the contextual markup tags provides hybrid HTML documents that remain compatible with HTML and avoids the
15 risk of a conflict between the contextual markup tags and HTML markup tags. In this latter method, the predefined prefix of the opening contextual markup tag is preferably inserted before at least one contextual term thereof for ease of viewing and processing. A terminator may be inserted between the HTML delimiters of the closing contextual markup tag to distinguish the closing contextual markup tag from
20 the opening contextual markup tag. The marked item of computer-readable data may also be advantageously marked with additional pairs of opening and closing contextual markup tags.

25 In another aspect of the invention, a computer-readable memory is used to store a spider and an index processor. The spider is programmed to scan electronic documents for context sensitive data and to retrieve the context sensitive data from the electronic documents, including electronic documents which conform to a context insensitive markup language. The index processor is programmed to add the context sensitive data retrieved by the spider to a context sensitive database. The spider
30 preferably has computer-readable instructions to identify items of computer-readable data marked by at least one pair of contextual markup tags within the electronic documents. These latter instructions may be performed in conjunction with a separate

- 6 -

set of code making up a parser. The spider can retrieve contextual terms from the contextual markup tags and the items of computer-readable data marked by the contextual markup tags. The index processor preferably includes instructions to index within the context sensitive database the items of computer-readable data at least according to the contextual terms retrieved from the contextual markup tags associated with the items of computer-readable data. In one embodiment, a crawler is included which is programmed to perform a preliminary scan of the electronic documents to determine which of the electronic documents include at least one item of context sensitive data.

Retrieving and indexing context sensitive data from an electronic document, generated with a context insensitive markup language but marked with contextual markup tags, provides a mechanism for supporting a richly organized context sensitive database which may then be readily and accurately searched in a contextual basis. Such context-based searches performed on the context sensitive database produce search results that are much less cluttered with irrelevant or out of context information, allowing users to more quickly retrieve relevant search results.

In yet another aspect of the invention, a method is provided for managing a context sensitive database in a computer system. In this aspect, an electronic document is scanned in search of items of computer-readable data marked with contextual markup tags that have contextual terms associated with the items of character data. The items of computer-readable data and associated contextual terms are retrieved from the electronic document and added to the context sensitive database. Preferably, in this latter stage the associated contextual terms and the items of computer-readable data are linked, and the items of computer-readable data are linked to an address identifying a location of the electronic document from which originated the items of computer-readable data and associated contextual terms. Preferably, during the scanning of the electronic document, contextual markup tags are examined for a predefined prefix distinguishing the contextual markup tags from other markup tags within the electronic document. When the electronic document is scanned, the search

- 7 -

can be limited to contextual markup tags beginning and ending with at least one HTML delimiter.

5 According to another aspect of the invention, a computer system is provided having a context sensitive database and a spider sub-system. The spider sub-system is programmed to scan an electronic document for items of computer-readable data marked by contextual markup tags that each include a contextual term associated with at least one of the items of computer-readable data. The spider sub-system is also
10 programmed to add the items of computer-readable data and associated contextual terms to the context sensitive database. When scanning the electronic document, the spider sub-system may also be programmed to scan the contextual markup tags for a predefined prefix distinguishing the contextual markup tags from other markup tags. Preferably, the spider sub-system includes instructions to link the associated contextual terms and the items of computer-readable data and to link the items of
15 computer-readable data to an item of location information identifying a location of the electronic document.

In accordance with another aspect of the invention, a computer system is provided having a context sensitive database and a computer server coupled to the context
20 sensitive database, wherein the computer system includes a search engine. The search engine is programmed to receive a search request from a requesting device. Based on the search request, the search engine is programmed to search the context sensitive database for references to electronic documents containing items of computer-readable data and contextual terms associated with the items of computer-readable
25 data. The references and corresponding items of computer-readable data and associated contextual terms are retrieved from the context sensitive database by the search engine and transmitted to the requesting device.

30 In accordance with another aspect of the invention, a computer-readable medium is provided containing a data structure having a first segment, a second segment and a third segment. The first segment identifies a resource that contains an electronic document having items of computer-readable data marked with contextual markup

- 8 -

tags including contextual terms associated with the items of computer-readable data and a predefined prefix distinguishing the contextual markup tags from other markup tags. The second segment identifies at least one of the items of the character data located within the electronic document. The third segment identifies at least one of the contextual terms associated with the at least one of the items of computer-readable data.

In accordance with yet another aspect of the invention, a computer-readable medium is provided containing a context sensitive database having items of computer-readable data and associated contextual terms scanned from electronic documents generated with a context insensitive markup language but embedded with contextual markup tags containing the contextual terms. This latter computer-readable medium also contains location information identifying addressable locations for the electronic documents.

According to yet another aspect, a computer system is provided for managing a context sensitive database wherein the computer system includes means for scanning an electronic document, generated with a context insensitive markup language, in search of items of computer-readable data marked with contextual markup tags, means for retrieving contextual terms associated with the items of computer-readable data from the contextual markup tags, means for retrieving the items of computer-readable data from the electronic document, and means for adding the items of computer-readable data and associated contextual terms to the context sensitive database.

In accordance with another aspect of the the invention, a method is provided of performing a context-based computer search on a context sensitive database. In this latter method, a search request containing a search term is received from a requesting party. The search term is compared with contextual terms stored in the context sensitive database. At least one contextual term, found to be associated with or to match the search term, is retrieved from the context sensitive database. The retrieved contextual term(s) is transmitted to the requesting entity along with instructions for a

- 9 -

context-based search request to be submitted using the contextual term(s) transmitted to the requesting party.

Other aspects and features of the present invention will become apparent to those ordinarily skilled in the art upon review of the following description of specific embodiments of the invention in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

In the accompanying drawings which illustrate embodiments of the invention,

FIG. 1 is a schematic diagram of a back end system for scanning HTML Web pages and retrieving context sensitive data from the scanned Web pages, according to a first embodiment of the invention;

FIG. 2 is a block diagram illustrating the conventional structure of an HTML document;

FIG. 3 is a block diagram of the general structure by which context sensitive data is embedded in HTML documents to form hybrid HTML documents in accordance with the first embodiment;

FIG. 4 is another diagram illustrating the general structure for context sensitive data in the first embodiment;

FIG. 5 is a block diagram illustrating a hybrid HTML document having character data encapsulated with contextual markup tags in accordance with the first embodiment;

FIG. 6 is another schematic diagram of the back end system of the first embodiment;

- 10 -

FIG. 7 is a schematic diagram of a search engine system according to the first embodiment having a front end system and back end system;

FIG. 8 is a block diagram illustrating the structure of a context sensitive database for storing information extracted from hybrid HTML documents;

FIG. 9 is a flow diagram illustrating the generation hybrid HTML documents; and

FIG. 10 is a flow diagram illustrating a method of processing hybrid HTML documents in accordance with the first embodiment.

It will be appreciated that for simplicity and clarity of illustration, elements illustrated in the accompanying drawings have not necessarily been drawn to scale. For example, the dimensions of some of the elements are exaggerated relative to other elements for clarity. Further, where considered appropriate, reference numerals and labels have been repeated among the drawings to indicate corresponding or analogous elements.

DETAILED DESCRIPTION

Reference will now be made in detail to implementations and embodiments of the invention, examples of which are illustrated in the accompanying drawings. In the implementations and embodiments which follow, the present invention is applied to conventional HTML documents to generate hybrid HTML documents embedded with context sensitive data. However, upon reading this specification, it will be appreciated by persons skilled in the art that the methodology of the present invention can be equally applied to electronic documents generated with XHTML or a context insensitive markup language that is a subset of SGML.

- 11 -

FIG. 1 is a schematic diagram of a back end system 20 for scanning hybrid HTML Web pages 40 and retrieving context sensitive data from the hybrid Web pages 40 according to a first embodiment of the invention. The back end system 20 includes a search processor 22 that resides in memory 25 and executes as software on a back end computer server 24. The search processor 22 receives and stores location information submitted by Web authors, Webmasters, publishers, organizations, crawlers and the like via user machines 44, Web site servers 42 and other networked sources. The location information identifies the location of resources directly or indirectly accessible by the back end computer server 24 and which have hybrid HTML documents (e.g. hybrid Web pages 40) containing context sensitive data to be processed by the search processor 22, as further described below. In this specification, the term "resource" refers to any computer-implemented object or data that can be accessed via the Internet or another computer network (intranet, LAN, wireless etc.) and which contains (or refers to electronic data files which contain), in whole or in part, text-based information. Examples of resources include Web sites, Web pages, file directories, URIs, URNs, URLs, IP addresses, POP, S/MIME, electronic data files and other electronic documents accessible over a network.

In the first embodiment, the location information is represented by Uniform Resource Locators (commonly known as URLs) which specify the locations of hybrid Web pages 40 made up of HTML documents embedded with context sensitive data (referred to in this specification as "hybrid HTML documents"). URLs received by the search processor 22 are preferably stored by the back end computer server 24 as a list (or queue) of URLs 28 on a local storage device 30. As discussed in further detail below, the search processor 22 includes software components which use the list of URLs 28 to access, retrieve and process the hybrid HTML documents so that the context sensitive data contained within the hybrid HTML documents can be extracted and indexed within a context sensitive database 32 accessible by the back end computer server 24. Contextual markup tags 56 and 58 (see FIG. 3) encapsulate items of computer-readable data, including character data 52 and other markup tags (for example, graphical or multimedia objects), to form the context sensitive data within

- 12 -

the hybrid HTML documents. The term "character data" refers to textual elements of an electronic document which are not part of any HTML markup tags.

Once the context sensitive data is processed and added to the context sensitive database 32, the context sensitive data may be used by a search engine 82 running on a front end computer server 80 (see FIG. 8) to generate search results identifying the location(s) of hybrid Web pages 40 (or other hybrid HTML documents which are not part of the Web) containing the context sensitive data.

As illustrated in FIG. 2, conventional HTML documents 46 contain character data 48 (and other items of computer-readable data 47) that lacks contextual meaning. As a result, while such HTML documents may be searched for relevant information, the lack of contextual meaning for the character data 48 of HTML documents makes indexing such character data and performing automated searches over large volumes of such standard HTML documents difficult. Often, searching such conventional HTML documents results in search results that are not particularly relevant to the human or machine entity requesting the search.

FIG. 3 is a block diagram illustrating the general structure used in the first embodiment to identify context sensitive data within hybrid HTML documents. In its most basic form, an item of context sensitive data within a hybrid HTML document is made up of character data 52 (or markup tags or both) encapsulated with opening and closing contextual markup tags 56 and 58 each comprising delimiters 60, contextual term 64, and preferably, predefined prefix 62. In the first embodiment, the closing contextual markup tag 58 is distinguished from the opening contextual markup tag 56 by a terminator 66 located in the closing contextual markup tag 58. As illustrated in FIG. 4 and 5, the terminator 66 is preferably a backslash ("/") located just after an initial HTML delimiter 60 of the closing contextual markup tag 58. Using a backslash for the terminator 66 and locating it just after the initial HTML delimiter 60 provides a mechanism for marking the closing contextual markup tag 58 as an end tag in a manner that is widely recognized by parsers and browsers. In an alternative arrangement, the opening and closing contextual markup tags 56 and 58 may be

- 13 -

differentiated with a distinguishable indicator located within the opening contextual markup tag 56.

FIG. 5 illustrates a sample hybrid HTML document 50 having character data 52 encapsulated with opening and closing contextual markup tags 56 and 58 in accordance with the first embodiment. For ease of reference, the contextual markup tags 56 and 58 used in document 50 are presented in upper case to visually distinguish them from the conventional HTML markup in the hybrid HTML document 50. As illustrated, the contextual markup tags 56 and 58 are implemented using proper tag nesting, with an opening contextual markup tag 56 preceding a corresponding closing contextual markup tag 58 in the hybrid document 50. The contextual markup tags 56 and 58 also begin and end with HTML style delimiters 60. The HTML style delimiters 60 implemented in the contextual markup tags are preferably the commonly used less-than (" $<$ ") and greater-than (" $>$ ") characters. By defining the boundaries of the contextual markup tags 56 and 58 with HTML delimiters 60, the contextual markup tags 56 and 58 are concealed from view when the hybrid HTML document 50 is rendered by a Web browser such as Microsoft's Internet Explorer™, while the specific character data 52 encapsulated by the contextual markup tags 56 and 58 is presented by the Web browser as such character data 52 ordinarily would be presented in the absence of the contextual markup tags 56 and 58. Thus, by integrating a characteristic of HTML delimiters into the predefined structure of the contextual markup tags 56 and 58, the contextual markup tags 56 and 58 can be used to add meaning to character data in an HTML document without adversely affecting the visual presentation of the HTML document when it is processed through a Web browser.

In addition to the HTML delimiters 60, each contextual markup tag (56, 58) includes at least two contextual components located between the delimiters 60: a predefined prefix 62 and a contextual term 64. As indicated above, each contextual markup tag (56, 58) preferably includes predefined prefix 62. The predefined prefix 62 is used to identify a tag as a contextual markup tag. The predefined prefix 62 is a predefined set of one or more characters which do not conflict with any predefined HTML tags or

- 14 -

any other standards recognized for use within HTML tags. Although the contextual markup tags can be used to add context to content within HTML documents without using the predefined prefix 62, the use of the predefined prefix 62 is preferred as it avoids the risk of a contextual markup tag conflicting with known or future HTML tags.

In the first embodiment, for ease of illustration, the predefined prefix 62 is a sequence of characters, "XHML", which serves as a flag identifying the information between the HTML delimiters 60 as contextual markup information. By implementing the contextual markup tags (56, 58) with HTML delimiters 60 and the predefined prefix 62 advantageously provides a mechanism for adding meaning to character data while avoiding the risk that the contextual markup tags may otherwise interfere with or corrupt the ordinary processing of the hybrid HTML document 50 as a conventional HTML document by a Web browser.

The contextual term 64 within a contextual markup tag provides the actual context specific meaning to the character data 52 encapsulated by an opening and closing pair of contextual markup tags 56 and 58. The contextual term 64 may be any set of one or more characters (e.g. alphanumeric characters, special characters, etc.). Thus, the contextual term 64 may include one or more words, or a sequence of characters which have no meaning in any written language but which provide a representation that the author of the contextual markup tags wishes to use to index the encapsulated character data. For example, the contextual term 64 may include a sequence of characters such as "xqtr" which, while this sequence has no meaning in written human language, may be used to associate character data with a computer-readable classification recognized by a computer program.

In an extension of the base structure of the opening and closing contextual markup tags (56, 58), a predefined separator 63 (the colon in FIG. 4 and 5) may be included between the predefined prefix 62 and the contextual term 64. The predefined separator 63 may be used to improve the human readability of the hybrid terminology when the source code for the hybrid HTML document 50 is viewed through a Web

- 15 -

browser, other viewer or in hardcopy format, particularly when the predefined prefix is a sequence of several characters.

5 The combination of HTML delimiters 60, the predefined prefix 62 and the contextual term 64 form the basis of the opening and closing contextual markup tags (56, 58) which are used to mark character data in HTML documents so as to form the hybrid HTML documents. Once an item of character data is encapsulated with a pair of such opening and closing contextual markup tags (56, 58), the resulting combination forms context sensitive data within the hybrid HTML documents that may be processed by
10 software components of the search processor 22 on the back end computer server 24.

As illustrated in FIG. 5, a particular set of character data (or other items of computer-readable data) may be encapsulated with more than one pair of contextual markup tags 56 and 58, providing the capacity to encapsulate character data in an HTML
15 document with layers of context so as to provide for different levels of meaning being assigned to the character data by the different layers of contextual markup tags. As indicated above, additional computer codes such as HTML tags may also be encapsulated by the contextual markup tags (56, 58), either alone or in combination with the character data 52. Thus, for instance, graphical objects or multimedia objects
20 may also be marked contextually with the contextual markup tags.

FIG. 6 shows a schematic diagram further illustrating the search processor software 22 of the first embodiment. The software components of the search processor 22 include a scheduler 68 and a spider sub-system 70. The scheduler 68 is programmed
25 to launch the spider sub-system 70 which retrieves, scans and indexes hybrid Web pages 40 identified by the URLs in the list of URLs 28. The scheduler 68 may use one of many known scheduling techniques to schedule the launch of the spider sub-system 70, including, by way of example, date and time scheduling, constant scheduling or event scheduling. In the first embodiment, the scheduler 68 schedules
30 the spider sub-system 70 to at predetermined intervals (e.g. at midnight each night).

- 16 -

The list of URLs 28 accessible to the spider sub-system 70 may be generated by one or more techniques. URLs may be submitted to the list 28 by web authors and the like to the search processor 22 via interface 71. URLs may be batch added to the list (or queue) 28. URLs may also be added to the list 28 by a crawler 67 residing on the back end computer server 24 or another computer. The crawler 67 is programmed to retrieve the Web pages 40 referred to in the list of URLs 28 and to scan such Web pages 40 for hyperlinks to other Web pages. The crawler 67 adds the URLs for such other Web pages to the list 28 and traverses the hyperlinks in search of further URLs to add to the list 28. The crawler 67 can validate URLs in the list 28 to ensure that such URLs still exist and may also verify that such URLs are accessible.

In a preferred arrangement, the crawler 67 is programmed to perform a preliminary scan of the Web pages 40 identified by the URLs stored in the list 28 to determine if these Web pages 40 in fact include context sensitive data. Where a preliminary scan is performed, the crawler 67 produces a refined list of URLs based on list 28. The refined list represents those Web pages 40 within which the crawler 67 has found context sensitive data that is marked using the aforementioned contextual markup tags (56, 58) coded in an acceptable configuration (see FIG. 3 to 5). In this case, the spider sub-system 70 accesses the refined list referring to Web pages 40 verified as having hybrid HTML documents as opposed to the original list 28 of unverified URLs.

As illustrated in FIG. 6, the spider sub-system 70 includes a spider 72, a parser 74, and an index processor 76. The spider 72 is programmed to retrieve the Web pages 40 referred to by the URLs in the list of URLs 28, to provide such retrieved Web pages 40 to the parser 74 and to scan such Web pages 40 for contextual markup tags of the type shown in FIG. 3 to 5. In the first embodiment, contextual markup tags (56 and 58) are identified by the spider sub-system 70 by detecting the predefined prefix 62 within such tags.

The parser 74 is programmed to parse the source code of Web pages retrieved by the spider 72. In order to parse a Web page (or other electronic document), the parser 74

- 17 -

accesses parsing rules specifying the acceptable structure and use of contextual markup tags 56 and 58 (see FIG. 3) in hybrid HTML documents. The parsing rules may also specify markup codes 77 that are to be excluded from processing if filtering is desired. The parsing rules can be stored using one of several approaches, including, as illustrated in FIG. 6, with the use of a parsing rules lookup table 75.

As the Web pages 40 are parsed, the contextual markup tags (56 and 58 in FIG. 3) and their associated character data (or other computer-readable data marked by tags 56 and 58) are retrieved. The contextual terms (64 in FIG. 3) are extracted from the contextual markup tags by the parser 74 (or the spider 72) and are used by the index processor 76 to index within the context sensitive database 32 the extracted character data associated with such contextual terms. Contextual terms which are not currently included in the context sensitive database 32 to classify character data and associated location information are added by the index processor 76 to the context sensitive database 32.

In one variation, predefined context symbols can be used by the spider sub-system 70 (and in particular the parser 74 in the first embodiment) to expand, collapse or modify at least one of the contextual terms retrieved from a hybrid HTML document. The predefined context symbols can be stored in a lookup table 79 which associates each of the predefined context symbols with one or more predetermined contextual terms. If a contextual term retrieved from a hybrid HTML document matches any of the predetermined contextual terms in the lookup table 79, the predefined context symbol corresponding with the matching predetermined contextual term is retrieved and used by the spider sub-system 70 to replace the retrieved contextual term. This technique may be used to replace certain contextual terms retrieved from a document with abbreviated contextual terms so as to reduce the amount of storage requirements for contextual terms within the context sensitive database 32. The predefined context symbols may also be used to expand or simply replace cryptic contextual terms retrieved from a document with more meaningful contextual terms using the lookup table.

- 18 -

Once the context sensitive data is extracted from the hybrid HTML documents (e.g. the hybrid Web pages 40) and the contextual terms and their associated character data and location information are added to the context sensitive database 32, the contextual terms and associated character data within the context sensitive database 32 may be used by a search engine on a front end computer server (see FIG. 8) to locate hybrid Web pages containing the context sensitive data and to generate search results identifying such hybrid Web pages. Advantageously, the context sensitive database 32 may also be used to store and search for context sensitive information retrieved from XML-derived (or based) documents. In this latter case the search processor 22 can support the processing and indexing of context sensitive data from XML-derived documents as well as from electronic documents embedded with pairs of the contextual markup tags 56 and 58.

FIG. 7 is a block diagram illustrating a structure 90 for the context sensitive database 32 (see FIG. 8) used to store contextual terms (e.g. 64 in FIG. 3), associated character data (e.g. 52 in FIG. 3) and associated location information for the source hybrid HTML documents (e.g. 40 in FIG. 8). The context sensitive database 32 is preferably structured to include a link table 92 which maps out the relative locations of a set of tables within the context sensitive database 32. As illustrated in FIG. 7, the link table 92 includes several fields including a contextual table ID field 93, a character data table ID field 95 and a resource location table ID field 97, that are used to reference a contextual terms table 94, a character data table 96 and a resource location table 98, respectively.

The contextual terms table 94 provides a mechanism to manage and link contextual terms (64) stored within the context sensitive database 32 along with character data (52) extracted from hybrid HTML documents (e.g. 40) and resource location information for the associated hybrid HTML documents. The contextual terms table 94 may also be used to associate contextual terms (64) with other information derived from hybrid HTML documents in which such contextual terms (64) are used to add meaning to character data (52). In the first embodiment, the contextual terms table 94 includes a plurality of records 100 for storing and associating valid contextual terms

- 19 -

(64) with character data (52) stored in the character data table 96 and with resource location information stored in the resource location table 98. Each record 100 includes: a term field 102 for storing a contextual term (64) which has been added to the context sensitive database 32 by the spider 70; and one or more fields 104
5 referencing locations in the character data table 96 where there is character data (52) associated with the contextual term (64) of the respective record 100. Preferably, each record 100 also includes one or more fields 106 referencing locations in the resource location table 98 where addressing information is located for resources associated with the contextual term of the respective record 100.

10 The character data table 96 provides a mechanism to store or reference character data (e.g. 52) and other items of computer-readable data which the spider has determined were encapsulated by contextual markup tags (56 and 58 in FIG. 3) within hybrid HTML documents 40. The character data table 96 includes a plurality of records 110
15 for storing and associating character data retrieved from hybrid HTML documents with location information identifying the hybrid HTML documents containing such character data. In the first embodiment, each record 110 includes a field 112 for storing character data (or references thereto). Each record 110 also includes one or more fields 114 linking the corresponding character data stored in field 112 with
20 location information within the resource location table 98 identifying hybrid HTML documents that contain such character data. The character data table 96 preferably also includes one or more fields 116 linking the character data with associated contextual terms stored in the contextual terms table 94. These latter fields 116 provide a mechanism for the search engine 82 (FIG. 8) to easily retrieve, using a
25 preliminary search term identifying character data within the context sensitive database 32, a list of contextual terms associated with the search term within the context sensitive database 32. The list of contextual terms may then be presented by the search engine 82 to the user to assist the user in refining his or her search based on the context assigned to one or more items of character data within the search term.

30 The resource location table 98 provides a mechanism to identify the specific location information for hybrid HTML documents 40 processed by the spider 70 (FIG. 6). The

- 20 -

resource location table 98 includes records 120 each having a field 122 that identifies the location information of a corresponding hybrid HTML document. The resource location table records 120 may also include one or more additional fields 124 to reference character data within the character data table 94 that are found in corresponding hybrid HTML documents. In a further variation, the resource location table records 120 may include one or more fields 126 to reference the contextual terms stored within the contextual terms table 96 used in the hybrid document. These latter fields 126 provide an easy mechanism for the search engine 82 to retrieve and transmit to a user all or some of the contextual terms used to give meaning within a hybrid HTML document. This can assist the user in his or her searching by providing a summary of the contextual meaning embedded within the hybrid HTML document.

FIG. 8 is a schematic diagram of a search engine system 85 according to the first embodiment having a front end system and back end system. On the front end, the search engine 82 resides as software on a front end computer server 80 and provides search engine services to user machines 84 directly or indirectly connected to the search engine 82. In the embodiment illustrated, some of the user machines 84 access the services of the search engine through a dial-up connection with an ISP and a network connection established over the Internet. Other connections between the user machines 84 and the computer server 80 may also be established. For example, the user machines may access the search engine services of the search engine 82 through an intranetwork, a direct dial-up connection, a cable or xDSL modem connection, a wireless connection, or a dedicated network connection or the like.

The search engine 82 is programmed to provide the user machines 84 with at least one type of search form 81 for completion by an end-user. Completed search forms 81 or the search criteria entered into such search forms 81 serve as search requests 83 which are transmitted to the front end computer server 80. Search requests 83 received by the front end computer server 80 are validated by the search engine 82 to ensure that they conform with at least one predefined search request structure recognized by the search engine 82.

- 21 -

Preferably, to assist the search engine 82 in performing a context sensitive search of the context sensitive database 32, the search forms 81 include at least one contextual term stored within the context sensitive database 32. This list provides the user with an indication of some or all of the contextual terms available within the context sensitive database 32 to assist the user in formulating a context-based search request 83. A search request 83 can be transmitted by the user machines 84 to the front end computer server 80 as a data signal 83a embodied in a carrier wave. This search request data signal 83a can represent one or more segments 83b of the search request 83. Each segment 83b may be used to identify search terms provided by the user and may include a segment 83c identifying one or more contextual terms defining the context within which the search is to be performed on some or all of the other search terms. In receiving the at least one contextual term with the context-based search request 83, the search engine 82 can then search the context sensitive database 32 for references to hybrid HTML documents (for example, Web pages 40) having character data associated with such contextual term(s) and return search results 87 to the user machine that submitted the context-based search request 83.

The search results 87 include information identifying one or more matches, if any, within the context sensitive database 32. A "match" represents an entry in the context sensitive database 32 identifying a hybrid HTML document having character data fitting within the parameters of the search criteria including being associated with one or more contextual terms which formed part of the requestor's context-based search request 83. A match may also include reference to a specific set of contextual terms and character data that form part of the match. A match can be transmitted from the computer server 80 to the requesting party's user machine 84 as a search result data signal 89 embodied in a carrier wave. The search result data signal 89 can have a segment 89a identifying the hybrid HTML document associated with the match.

In a variation, if the search request(s) 83 received by the search engine 82 lacks any contextual terms used within the context sensitive database 32, the search engine 82 may be preferably programmed to compare the search term(s) submitted with the search request(s) 83 with valid contextual terms used by the context sensitive database

- 22 -

32. If the search engine 82 determines that any valid contextual terms within the context sensitive database 32 are associated with (or match) any of the search terms, the search engine 82 then provides the user machine that submitted the search request 83 with at least one list of valid contextual terms in the context sensitive database 32 that are found to be associated with one or more of the search terms, along with instructions providing the user with the option to submit a context-based search request using one or more of the valid contextual terms provided. The user may then refine the search request 83 and specify the context sensitive nature of the search request 83 by adding one or more of the contextual terms from the list of valid contextual terms provided by the search engine 82. The refined search request (now context-based), once transmitted to and received by the front end computer server 80, may be used by the search engine 82 to conduct a context sensitive search for hybrid HTML documents referenced by the context sensitive database 32.

FIG. 9 to 10 show flow diagrams illustrating the operation of a system for authoring, altering, indexing, storing and retrieving electronic documents embedded with contextual markup tags according to the first embodiment of the invention. For ease of reference in the following discussions, reference is made to FIG. 3 to 10 collectively. Reference numerals and labels have been repeated among the drawings to indicate corresponding elements and blocks.

In the first embodiment, an HTML document is generated or edited by a Web author or publisher using a commercially available HTML compatible editor. However, any text editor or word processor capable of storing a document as a text file may be used to generate or edit HTML formatted files for publication as Web pages. As the HTML document is generated or edited, pairs of contextual markup tags (56 and 58) are added at block 200 by the Web author to encapsulate or "tag" character data within the HTML document, resulting in hybrid HTML document 40 (FIG. 9). As discussed above, the contextual markup tags (56 and 58) are used to assign meaning to the encapsulated character data so that compatible software applications such as the spider 70 can understand the context within which the encapsulated character data is used in the hybrid HTML document.

- 23 -

Once the HTML document is modified with the contextual markup tags to form the hybrid HTML document 40, the author (or publisher) preferably submits the location information (e.g. the URL) for the hybrid HTML document 40 to the interface 71 which the author communicates with via the user machine 84 at block 202. The interface 71 adds the URL to the list of URLs 28 identifying the location of hybrid HTML documents to be indexed by the search processor software 22.

At block 210 a preliminary scan is performed of the Web pages corresponding to the URLs in the list of URLs 28 managed by the search processor 22. Preferably, this preliminary scan is performed by the crawler 67 which scans through the Web pages to determine whether or not they contain at least one pair of contextual markup tags which conform to the specified syntax (see FIG. 3 and 4). The URLs of Web pages containing contextual markup tags are added to a refined list of URLs by the crawler 67.

After a refined list of URLs is generated at block 210, the spider sub-system 70 may be launched by the scheduler 68 to retrieve and index the hybrid Web pages 40 referenced in the refined list. By way of illustration, in the first embodiment the spider sub-system 70 selects at block 212 a URL from the list of refined URLs and invokes the spider 72 to retrieve the corresponding hybrid Web page 40 at block 214. The retrieved hybrid Web page 40 is scanned by the spider 72 at block 216 for context sensitive data identified by pairs of contextual markup tags (56 and 58). The context sensitive data is identified with the assistance of the parser 74 and the parsing rules lookup table 75. Context sensitive data that is identified in block 216 is retrieved by the spider 72 for further processing by the index processor 76. Preferably, for each retrieved Web page 40, the spider 72 retrieves and temporarily stores (caches) the contextual terms 64, the character data 52 corresponding to the contextual terms 64 (and other items of computer-readable data encapsulated by the contextual markup tags) and the location information for the retrieved Web page 40. When storing this retrieved collection of information, the spider 72 maintains the association between the contextual terms 64 and corresponding character data 52.

- 24 -

Once a retrieved hybrid Web page 40 is scanned and the context sensitive data retrieved and temporarily stored (cached), the context sensitive data may be added to the context sensitive database 32 at block 218. In the first embodiment, the spider 72 invokes the index processor 76 to organize the insertion of the temporarily stored context sensitive data into the context sensitive database 32. In block 218, the index processor 76 adds the location information for the retrieved hybrid Web page 40 to a new instance of record 120 in the resource location table 98 if the Web page has not already been indexed into the database 32.

The index processor 76 also updates the context sensitive database 32 in block 218 to identify the relationships between the character data and the contextual terms retrieved from the hybrid Web page 40. Updating the context sensitive database 32 in this way depends upon the state of the database 32 in relation to the retrieved character data and contextual terms. If the character data is new to the database 32 and the contextual term that encapsulated the character data is also new to the database 32, then the index processor 76 adds the new character data to a new instance of record 110 in character table 96. In this latter case, the index processor 76 also adds the new contextual term to the contextual terms table 94. The new contextual term is added to a new instance of record 100 which is linked to the corresponding character data in character data table 96.

If, in block 218, the character data is new to the database 32 but the contextual term that encapsulated the character data already exists within the database 32, then, as before, the index processor 76 adds the new character data to a new instance of record 110 in character table 96. However, the index processor 76 also retrieves the reference to the existing record 100 in the contextual terms table 94 that already stores the contextual term. The fields 104 and 106 for such existing record 100 are expanded or modified to add links to the new instance of record 110 and to the record in the resource location table 98 for the associated Web page 40. Similarly, if the contextual term is new but the particular character data marked by the contextual term already exists in the context sensitive database 32, the index processor 76 proceeds to

- 25 -

create a new instance of record 100 for the contextual term and retrieve and modify the existing record 110 for the character data to link the record 110 to the new record 100 and the record 120 for the location of the associated Web page 40.

5 If both the character data and the associated contextual term already exist in the context sensitive database 32, then the index processor 76 retrieves the existing record 100 for the contextual term and the existing record 110 for the character data and links those records to the record 120 for the associated Web page 40.

10 In block 218, a new instance of record 120 is used by the index processor 76 to store the location information for a Web page whose location does not already exist within the context sensitive database 32. If the location information to be stored in the context sensitive database 32 already exists in a record 120, then such record 120 is used by the index processor 76 to link with records (100 and 110) for new contextual
15 terms and character data retrieved from the Web page 40 associated with the location information.

After processing of the retrieved hybrid Web page 40 is completed in block 218, the spider sub-system 70 checks at block 220 to see if any further hybrid Web pages
20 remain to be processed from the refined list of URLs (generated in block 210). If any further hybrid Web pages remain, processing returns to block 212 where the spider sub-system 70 selects another URL from the refined list of URLs for the spider 72 to retrieve and process. Otherwise, the spider sub-system 70 concludes the current round of processing and awaits further instructions from the scheduler 68 to begin
25 processing once more from block 210.

Although this invention has been described with reference to illustrative and preferred embodiments of carrying out the invention, this description is not to be construed in a limiting sense. Various modifications of form, arrangement of parts, steps, details and
30 order of operations of the embodiments illustrated, as well as other embodiments of the invention, will be apparent to persons skilled in the art upon reference to this

- 26 -

description. It is therefore contemplated that the appended claims will cover such modifications and embodiments as fall within the true scope of the invention.

- 27 -

WHAT IS CLAIMED IS:

1. A computer-readable medium containing an electronic document generated with a context insensitive markup language, the electronic document having contextual markup tags and items of computer-readable data marked by the contextual markup tags.
5
2. The computer-readable medium of Claim 1, wherein each of the contextual markup tags includes a predefined prefix for identifying contextual information, and at least one contextual term identifying a context within which at least one of the items of computer-readable data is used within the electronic document.
10
3. The computer-readable medium of Claim 1, wherein the items of computer-readable data includes at least one reference to a computer-based multimedia entity or a computer-based graphical entity.
15
4. The computer-readable medium of Claim 1, wherein the items of computer-readable code include character data.
20
5. The computer-readable medium of Claim 1, wherein the electronic document is an HTML document.
6. The computer-readable medium of Claim 5, wherein the contextual markup tags include HTML delimiters, a predefined prefix for identifying contextual information, and at least one contextual term identifying a context within which at least one of the items of computer-readable data is used within the HTML document.
25
7. The computer-readable medium of Claim 6, wherein the contextual markup tags comprise pairs of contextual markup tags encapsulating the at least one of
30

- 28 -

the items of computer-readable data, including an opening contextual markup tag and a closing contextual markup tag.

5 8. The computer-readable medium of Claim 7, wherein the HTML delimiters include delimiters selected from the group consisting of: "<" and ">".

10 9. The computer-readable medium of Claim 8, wherein the closing contextual markup tag includes a terminator between the HTML delimiters of the closing contextual markup tag to distinguish the closing contextual markup tag from the opening contextual markup tag.

10. The computer-readable medium of Claim 9, wherein the terminator is a backslash located immediately after the "<" delimiter.

15 11. A method of generating a context sensitive HTML document, the method comprising:

20 inserting an opening contextual markup tag before an item of computer-readable data within an HTML document, including inserting as part of the opening contextual markup tag HTML delimiters, and at least one contextual term identifying a context within which the item of computer-readable data is used; and

25 inserting a closing contextual markup tag after the item of computer-readable data, including inserting as part of the closing contextual markup tag the HTML delimiters and the at least one contextual term.

30 12. The method as claimed in Claim 11, including inserting into each of the opening and closing contextual markup tags a predefined prefix for

- 29 -

distinguishing the opening and closing contextual markup tags from HTML tags.

5 13. The method as claimed in Claim 11, including inserting a terminator between the HTML delimiters of the closing contextual markup tag to distinguish the closing contextual markup tag from the opening contextual markup tag.

10 14. The method as claimed in Claim 11, including inserting the predefined prefix of the opening contextual markup tag before the at least one contextual term thereof.

15 15. The method as claimed in Claim 11, including marking the item of computer-readable data with additional pairs of opening and closing contextual markup tags.

16. A computer-readable memory comprising:

20 (a) a spider for scanning electronic documents for context sensitive data and for retrieving the context sensitive data from the electronic documents, wherein at least one of the electronic documents conforms to a context insensitive markup language; and

(b) an index processor, responsive to the spider, for adding the context sensitive data to a context sensitive database.

25 17. The computer-readable memory of Claim 16, wherein the spider further comprises computer-readable instructions to identify items of computer-readable data marked by at least one pair of contextual markup tags within the electronic documents.

30 18. The computer-readable memory of Claim 17, wherein the spider further comprises computer-readable instructions to retrieve contextual terms from the

- 30 -

contextual markup tags and to retrieve the items of computer-readable data marked by the contextual markup tags.

- 5 19. The computer-readable memory of Claim 18, wherein the index processor further comprises computer-readable instructions to index the items of computer-readable data within the context sensitive database at least according to the contextual terms retrieved from the contextual markup tags associated with the items of computer-readable data.
- 10 20. The computer-readable memory of Claim 19, further comprising a crawler having computer-readable instructions to perform a preliminary scan of the electronic documents to determine which of the electronic documents include at least one item of context sensitive data.
- 15 21. The computer-readable memory of Claim 19, wherein the at least one of the electronic documents complies with HTML.
22. The computer-readable memory of Claim 19, wherein a plurality of the electronic documents comply with XHTML.
- 20 23. The computer-readable memory of Claim 19, wherein the spider further comprises instructions to replace one or more predetermined contextual terms retrieved from the electronic documents with abbreviated contextual terms.
- 25 24. The computer-readable memory of Claim 19, wherein the spider further comprises instructions to expand or replace one or more predetermined contextual terms retrieved from the electronic documents with more meaningful contextual terms using a lookup table.
- 30 25. The computer-readable memory of Claim 16, wherein the spider further comprises instructions to replace one or more predetermined contextual terms retrieved from the electronic documents with abbreviated contextual terms.

- 31 -

26. The computer-readable memory of Claim 16, wherein the spider further comprises instructions to expand or replace one or more predetermined contextual terms retrieved from the electronic documents with more meaningful contextual terms using a lookup table.
27. A method of managing a context sensitive database in a computer system, the method comprising:
- scanning an electronic document in search of items of computer-readable data marked with contextual markup tags that have contextual terms associated with the items of computer-readable data;
 - retrieving the items of computer-readable data and associated contextual terms from the electronic document; and
 - adding the items of computer-readable data and associated contextual terms to the context sensitive database.
28. The method of Claim 26, wherein scanning the electronic document for a predefined prefix distinguishing the contextual markup tags from other markup tags within the electronic document.
29. The method of Claim 28, wherein adding further comprises linking the associated contextual terms and the items of computer-readable data, and linking the items of computer-readable data to an address identifying a location of the electronic document.
30. The method of Claim 29, wherein scanning further comprises searching for contextual markup tags beginning and ending with at least one HTML delimiter.

- 32 -

- 5
31. The method of Claim 30, wherein scanning further comprises identifying pairs of opening and closing contextual markup tags including identifying a terminator in the closing contextual markup tag.
32. The method of Claim 31, further comprising determining if the electronic document was generated with a context insensitive markup language.
- 10 33. The method of Claim 27, further comprising replacing one or more predetermined contextual terms retrieved from the electronic document with abbreviated contextual terms.
- 15 34. The method of Claim 27, further comprising replacing one or more predetermined contextual terms retrieved from the electronic document with other contextual terms using a lookup table.
35. A computer-readable memory having stored instructions for use in the execution of the method of Claim 27.
- 20 36. A computer-readable memory having stored instructions for use in the execution of the method of Claim 28.
37. A computer-readable memory having stored instructions for use in the execution of the method of Claim 29.
- 25 38. A computer-readable memory having stored instructions for use in the execution of the method of Claim 30.
- 30 39. A computer system comprising a computer and a memory having computer-readable codes for instructing the computer to perform the method of Claim 28.

- 33 -

40. A computer system comprising a computer and a memory having computer-readable codes for instructing the computer to perform the method of Claim 29.
- 5 41. A computer system comprising a computer and a memory having computer-readable codes for instructing the computer to perform the method of Claim 30.
- 10 42. A computer-readable memory having stored instructions for use in the execution of the method of Claim 31.
43. A computer system comprising:
- 15 (a) a context sensitive database; and
- (b) a spider sub-system having computer-readable instructions to: (i) scan an electronic document for items of computer-readable data marked by contextual markup tags that each include a contextual term associated with at least one of the items of computer-readable data, and (ii) add
- 20 the items of computer-readable data and the associated contextual terms to the context sensitive database.
44. The computer system of Claim 43, wherein the spider sub-system further comprises computer-readable instructions to scan the electronic document for
- 25 a predefined prefix within each contextual markup tag, the predefined prefix distinguishing the contextual markup tags from other markup tags.
45. The computer system of Claim 44, wherein the spider sub-system further comprises computer-readable instructions to link the associated contextual
- 30 terms and the items of computer-readable data and to link the items of computer-readable data to an item of location information identifying a location of the electronic document.

- 34 -

46. A computer system comprising:

(a) a context sensitive database; and

(b) a computer server coupled to the context sensitive database, the computer server comprising a search engine operable to instruct the computer server to:

(i) receive a search request from a requesting device;

(ii) based on the search request, search the context sensitive database for references to electronic documents generated with a context insensitive markup language and containing items of computer-readable data embedded with contextual markup tags; and

(iii) retrieve from the context sensitive database at least one of the references; and

(iv) transmit the at least one reference to the requesting device.

47. The computer system of Claim 46, wherein the search engine is further operable to instruct the computer server to retrieve from the context sensitive database, and transmit to the requesting device, a contextual term associated with at least one of the items of computer-readable data embedded with contextual markup tags.

48. A computer-readable medium containing a data structure comprising:

(a) a first segment identifying a resource that contains an electronic document having items of computer-readable data marked with

- 35 -

contextual markup tags, wherein the contextual markup tags each include (i) a contextual term associated with at least one of the items of computer-readable data and (ii) a predefined prefix distinguishing the contextual markup tags from other markup tags;

5

(b) a second segment identifying at least one of the items of computer-readable data located within the electronic document; and

10

(c) a third segment identifying the contextual term associated with the at least one of the items of computer-readable data marked with the contextual tags.

49. A computer-readable medium containing a context sensitive database comprising:

15

(a) items of computer-readable data and contextual terms associated with the items of computer-readable data, the items of computer-readable data representing information scanned from electronic documents generated with a context insensitive markup language but embedded with contextual markup tags containing the contextual terms; and

20

(b) location information identifying addressable locations for the electronic documents.

25

50. The computer-readable medium of Claim 49, wherein the electronic documents include at least one HTML document.

30

51. The computer-readable medium of Claim 49, the context sensitive database further comprising:

- 36 -

- (a) a first table containing a plurality of records identifying the location information for each of the electronic documents;
- 5 (b) a second table containing a plurality of records for storing the items of computer-readable data and for associating the items of computer-readable data with the location information of one or more of the electronic documents containing such items of computer-readable data; and
- 10 (c) a third table containing a plurality of records for storing and relating the contextual terms with (i) the items of computer-readable data stored in the second table and (ii) the location information stored in the first table.
- 15 52. A computer system for managing a context sensitive database, the computer system comprising:
- (a) means for scanning an electronic document, generated with a context insensitive markup language, in search of items of computer-readable data marked with contextual markup tags;
- 20 (b) means for retrieving contextual terms associated with the items of computer-readable data from the contextual markup tags;
- 25 (c) means for retrieving the items of computer-readable data from the electronic document; and
- (d) means for adding the items of computer-readable data and associated contextual terms to the context sensitive database.
- 30 53. The computer system of Claim 52, comprising means for linking the associated contextual terms and the items of computer-readable data within the

- 37 -

context sensitive database and for linking the items of computer-readable data stored in the context sensitive database to an address identifying a location of the electronic document containing the items of computer-readable data and associated contextual terms.

5

54. A method of performing a context-based computer search on a context sensitive database, the method comprising:

receiving from a requesting entity a search request containing a search term;

10

comparing the search term with contextual terms stored in the context sensitive database;

retrieving from the context sensitive database at least one contextual term found to be associated with or match the search term; and

15

transmitting the at least one contextual term to the requesting entity along with instructions for a context-based search request to be submitted using the at least one contextual term transmitted.

20

1/8

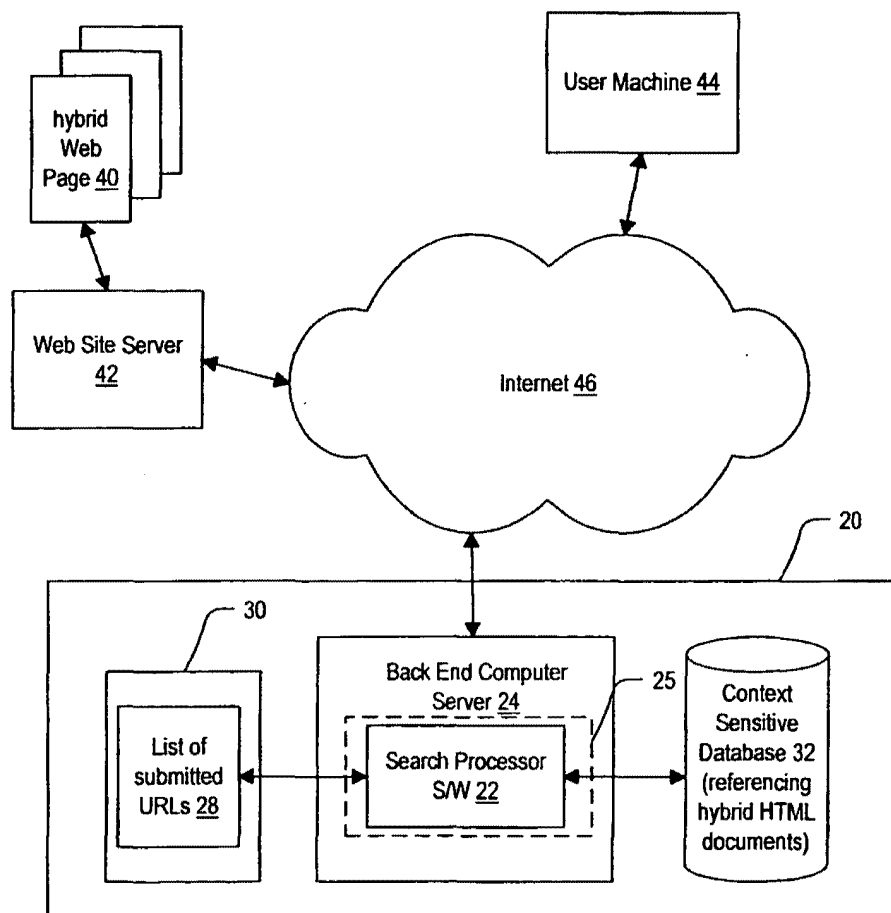


FIG. 1

2/8

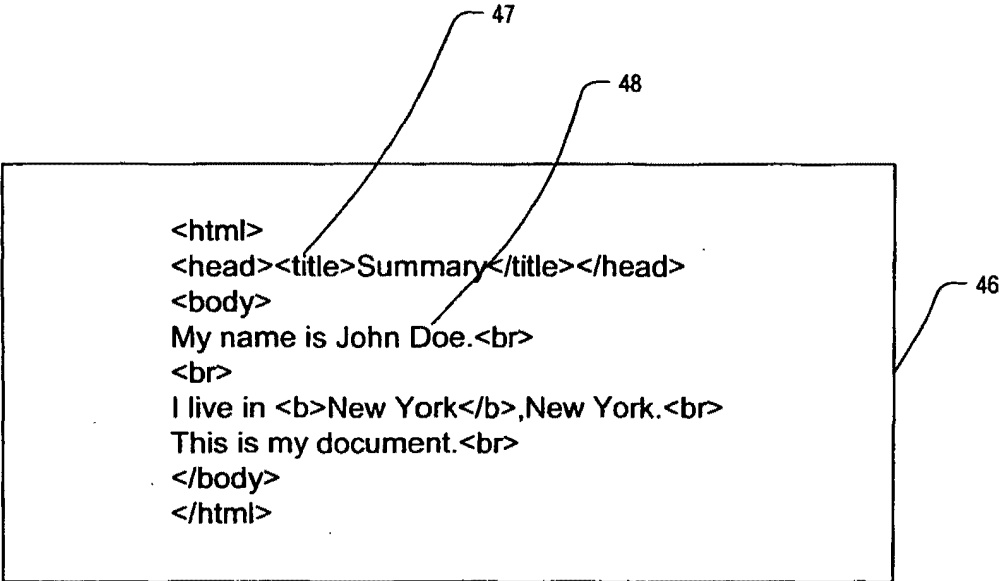


FIG. 2
(Prior Art)

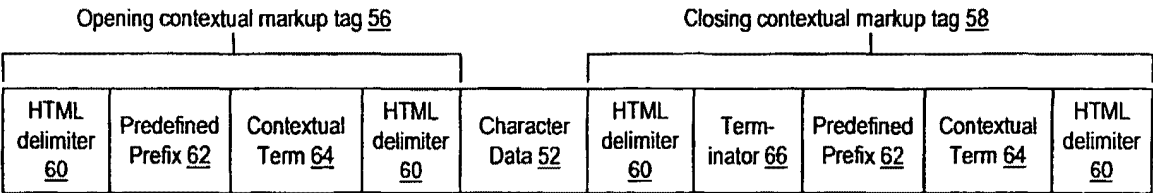


FIG. 3

3/8

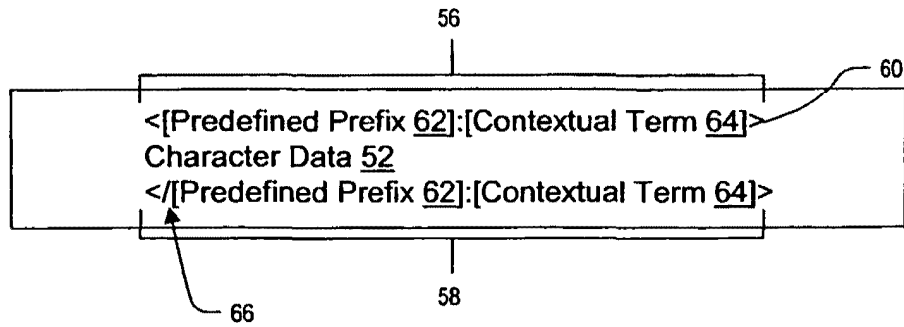


FIG. 4

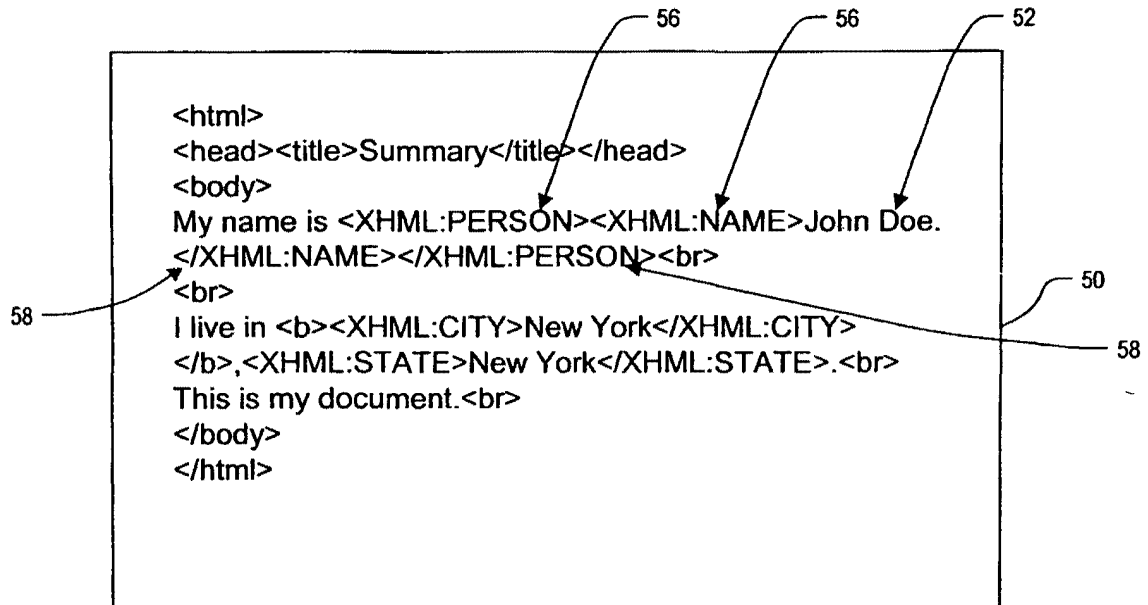


FIG. 5

4/8

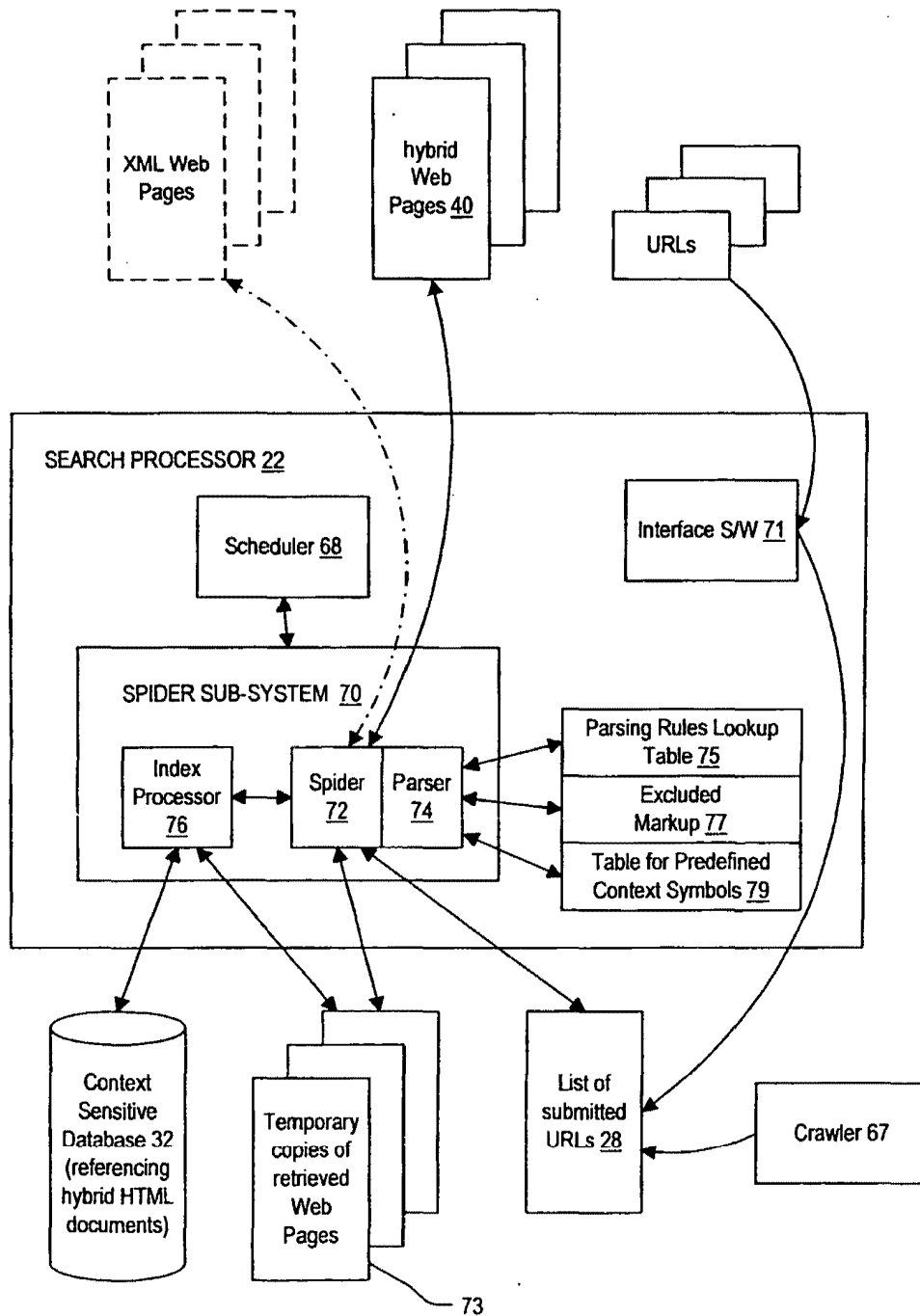


FIG. 6

5/8

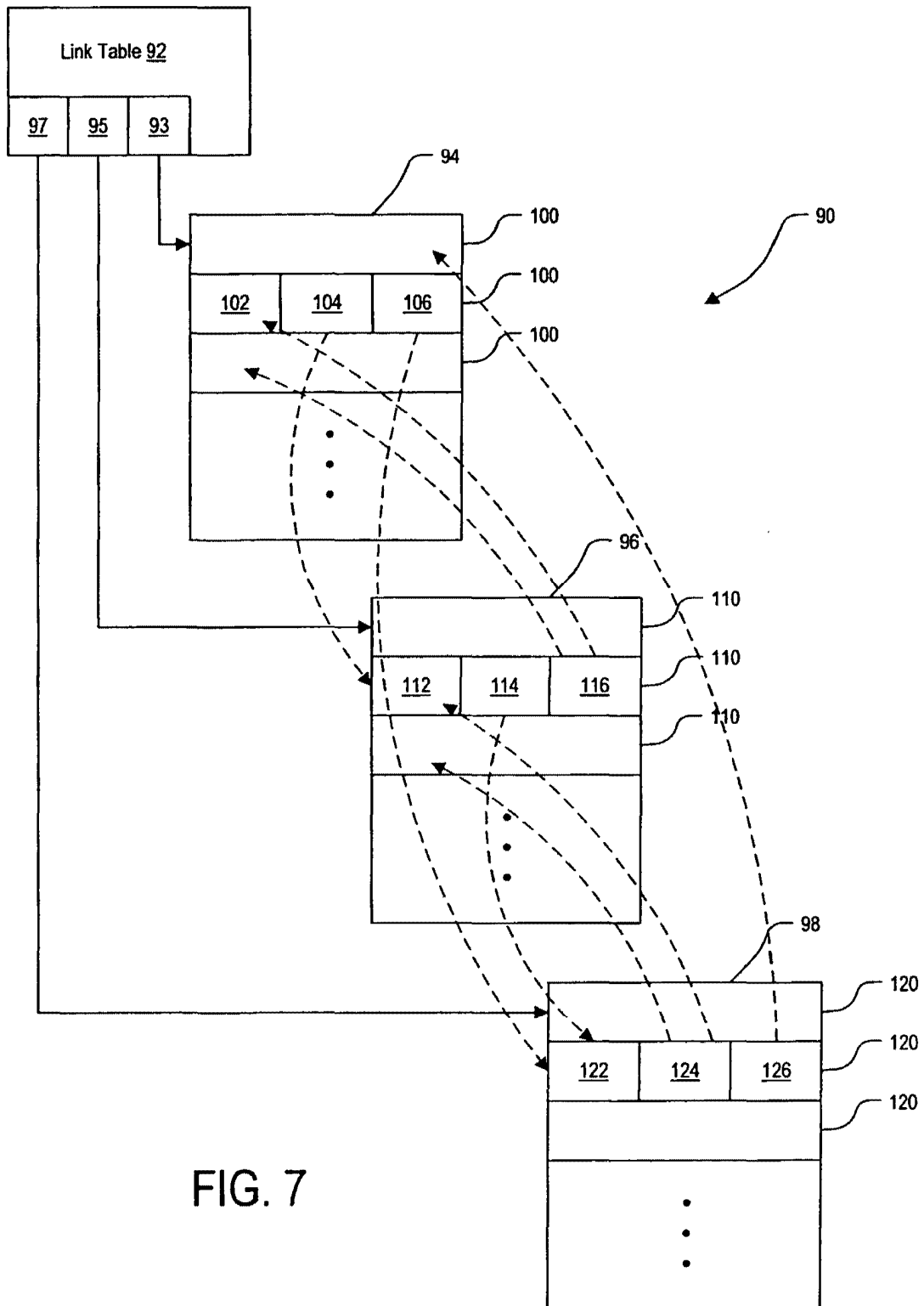


FIG. 7

6/8

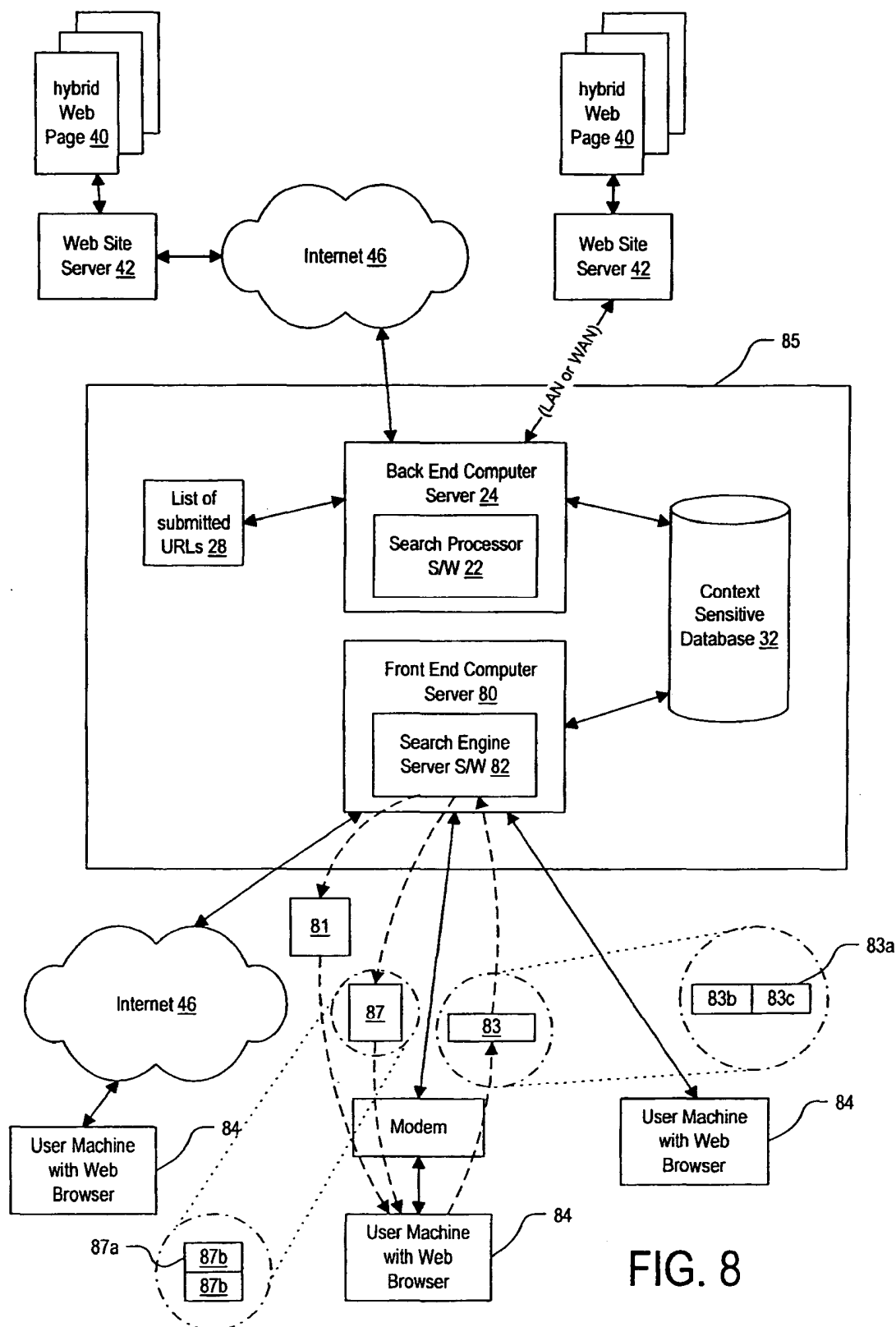


FIG. 8

7/8

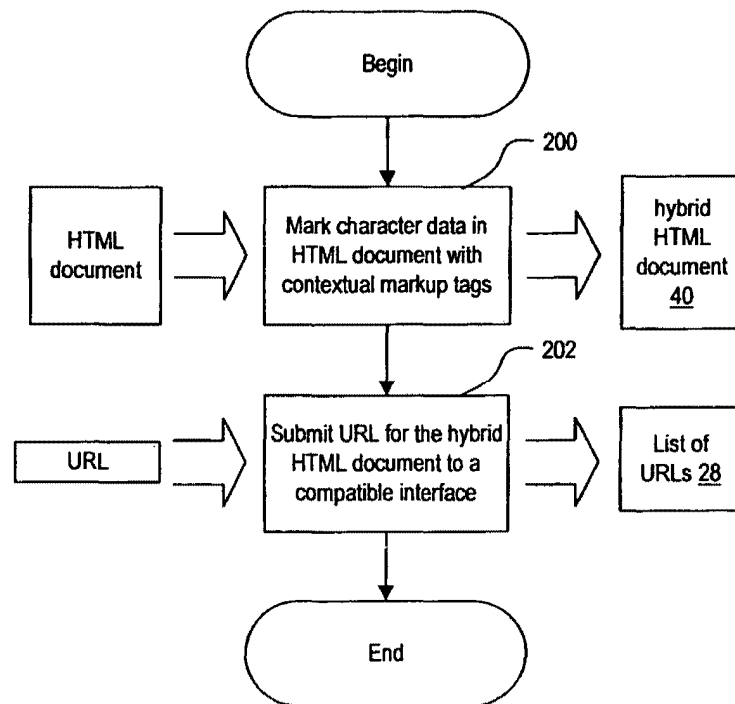
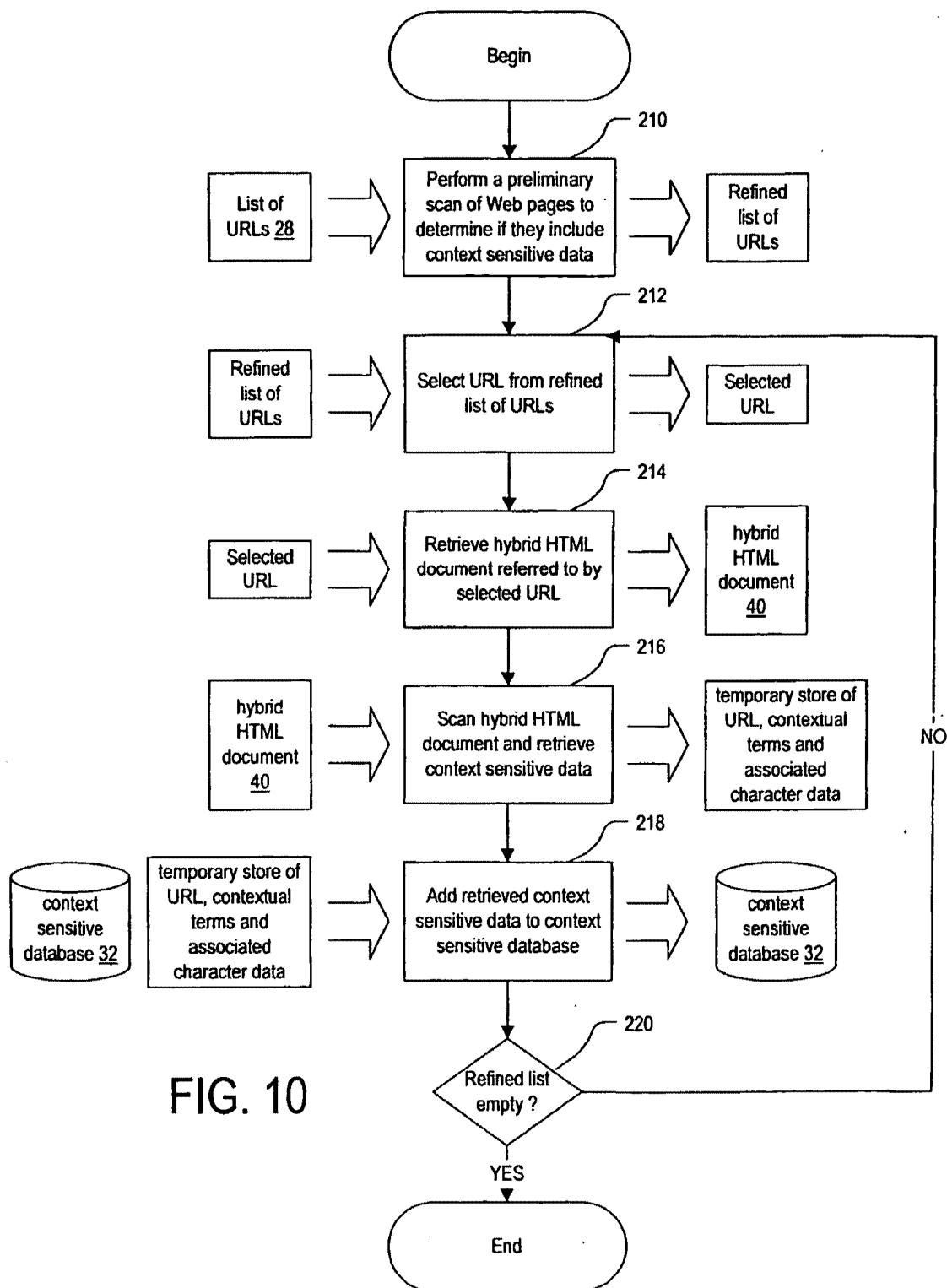


FIG. 9

8/8



CA9-98-038

WEB-BASED FILE REVIEW SYSTEM

ABSTRACT

A system for reviewing files which permits comments to be inserted in files to be viewed with a hypertext browser. When the hypertext mark-up language employed is HTML, text files are converted to an HTML representation. An HTML file is represented by a linked list of objects. Comment insertion markers and comment display objects are inserted at predefined points in the HTML linked list representation. The linked list is stored as a binary file and has a comment file associated with it. Access to the HTML file by reviewers and authors causes the regeneration of the HTML document by a Common Gateway Interface which recreates the linked list representation of the document from the binary file and which then generates HTML code from the linked list. Comments may be entered by reviewers working in parallel on the HTML document. Comments are displayed at the next regeneration of the HTML document by the system.

BEST AVAILABLE COPY

CA9-98-038

WEB-BASED FILE REVIEW SYSTEM

FIELD OF THE INVENTION

5 The present invention is directed to an improvement in computing systems and in particular to systems for the review of files by multiple reviewers.

BACKGROUND OF THE INVENTION

10 Computer systems often provide review tools which allow writers and developers to review drafts of information. The information can be in the form of simple text files representing documents or computer program code, or may be structured documents such as those defined by such languages as HTML. The latter type of documents are in common use due to the existence of web-based publishing, both on the Internet World Wide Web, and on Intranets which are found within a company or organization.

15 Prior art systems, such as the IBM host-based tool, Revufile™, have been developed which provide online reviewing environments for flat linear documents. Such tools, however, are not appropriate for reviewing webs of information, and no appropriate web-based review tool has been developed. Previously developed systems are constrained in accepting comments to files made by different reviewers in parallel and in permitting dynamic display of the comments made to files.

20 It is therefore desirable to have a review tool which will provide an interaction with its users through a web browser, whether the tool is running through the Internet and the World Wide Web, or through a company Intranet or Local Area Network. Such a review tool or system will permit comments to be added to documents and displayed dynamically.

CA9-98-038

SUMMARY OF THE INVENTION

According to one aspect of the present invention, there is provided an improved system and method for storing, managing and displaying comments in a file review system.

According to another aspect of the present invention there is provided a file review system for
5 storing and managing a set of comments associated with a source file, comprising:

- (a) means for accepting data from the source file and storing a representation of the source file as a markup file,
- (b) means for creating a comment file containing data representing the set of comments associated with the source file,
- 10 (c) means for accepting new comments for inclusion in the set of comments associated with the source file and for updating the comment file to correspond to the complete set of comments,
- (d) means for generating an hypertext document from the markup file and from the comment file, the hypertext document corresponding to the source file and including
15 portions corresponding to one or more of the set of comments associated with the source file, and
- (e) means for communicating the hypertext document to a user for display.

According to another aspect of the present invention there is provided a the above file review system in which the means for accepting data from the source file and storing the source file as a markup
20 file further comprises means for representing the source file as a linked list of objects corresponding to a hypertext representation of the source file, wherein the markup file is a binary file representation of the linked list, the objects in the linked list being appropriately selected from the set of objects comprising simple tag objects, tag objects, end tag objects, text objects, comment insertion objects,

CA9-98-038

comment display objects, complex whitespace objects, spaces objects, newlines objects, hypertext comment objects, ordered list item tag objects, and link tag objects.

5 According to another aspect of the present invention there is provided the above file review system in which the means for representing the source file as a linked list of objects further comprises means for inserting comment insertion objects into the linked list of objects and in which the means for generating a hypertext document further comprises means to include hypertext data in the hypertext document to display one or more symbols representing each comment insertion object.

10 According to another aspect of the present invention there is provided the above file review system in which the one or more symbols representing each comment insertion object comprise a hypertext link to a hypertext form for user input of a comment.

15 According to another aspect of the present invention there is provided a file review system of the type described above in which the means for representing the source file as a linked list of objects further comprises means for inserting comment display objects into the linked list of objects, each comment display object being associated with a comment insertion object and in which the means for generating a hypertext document further comprises means to include hypertext data in the hypertext document to display comments in accordance with the corresponding comment display objects.

20 According to another aspect of the present invention there is provided the above file review system in which each one of the comment display objects comprises

(a) open tag data representing the open hypertext tags, at the location of the comment display object, in the linked list of objects

CA9-98-038

(b) and in which the means for generating a hypertext document further comprises

means to insert closing hypertext data in the hypertext document, prior to a selected one of the comments being displayed in the hypertext document, and means to insert reopening hypertext data in the hypertext document, following the selected one of the comments being displayed in the hypertext document, the closing hypertext data and the reopening hypertext data both being defined by the open tag data for the comment display object associated with the selected one of the comments being displayed.

According to another aspect of the present invention there is provided a file review system of the type described above in which the means for generating an hypertext document further comprises means to generate a hypertext document which corresponds to a predefined subsection of the markup file and the associated subsection of the comment file.

According to another aspect of the present invention there is provided a file review system of the type described above, in which each object in the linked list of objects has a type, and comprises contents derived from a portion of the source file, each comment insertion object and each comment display object comprises a contents line identifier, representing the line number defining the location of the object in relation to the source file, the comment file consists of comment line identifier data associated with the set of comments representing the line numbers in the source file with which each comment is related, and in which the means for generating a hypertext document further comprises

- (a) means to traverse the linked list of objects comprising, and means to select a display object from the linked list of objects,
- (b) means to determine the type of the display object,
- (c) means to generate hypertext data to correspond to the type of the display object and the contents of the display object,
- (d) means to identify the contents line identifier for the display object and for selecting

CA9-98-038

comment display data from the comment file based on the contents line identifier and the comment line identifiers of the comment file, and

(e) means to generate Hypertext data from the comment file representing the selected one of the set of comments corresponding to the comment line identifier associated with the contents
5 line identifier.

According to another aspect of the present invention there is provided a file review system as described above in which each comment in the set of comments has associated with it a severity and in which the comment file consists of data representing the severity of each comment, and in which the means for accepting new comments for inclusion in the set of comments associated with the
10 source file and for updating the comment file to correspond to the complete set of comments, comprises means for accepting data representing the severity of the new comments, and means for updating the comment file to add data representing the severity of the new comments.

According to another aspect of the present invention there is provided a file review system of the
15 type described above, in which the means for generating a hypertext document comprises means for selectively excluding portions of the comment file corresponding to one or more comments in the set of comments, whereby a selectively defined subset of the set of comments is represented in the hypertext document.

20 According to another aspect of the present invention there is provided a file review system of the type described above in which comments have attributes which are defined from the following options: comment type, comment severity, comment date, comment status, comment author, and comment content.

CA9-98-038

According to another aspect of the present invention there is provided a file review system of the type described above in which system users are uniquely identified in the system and in which the means for accepting new comments further comprises means for selectively restricting the inclusion of new comments in the set of comments based on the unique identity of system users.

5

According to another aspect of the present invention there is provided a file review system of the type described above further comprising means to navigate the comment file to provide user access to comments in user-defined sequenced order.

10

According to another aspect of the present invention there is provided a web-based file review system for storing and managing comments from a plurality of reviewers, the comments being associated with one or more webs of source files, comprising

15

(a) a parser to parse a selected one of the set of source files into a linked list of objects corresponding to a hypertext representation of the selected source file, the linked list further comprising comment insertion objects and comment display objects, the parser writing the linked list of objects to a binary markup file representing the linked list of objects and corresponding to the selected one of the set of source files, each comment display object being capable of being associated with one or more comments,

20

(b) a set of comment files, each comment file being associated with a one of the set of source files and comprising data representing comments associated with the one of the set of source files,

(c) common gateway interface program code means for accepting new comments for inclusion in the set of comments associated with a reviewer-defined source file and

CA9-98-038

for updating the associated comment file,

- 5 (d) common gateway interface program code means for generating a hypertext document from a markup file corresponding to reviewer-selected source file and from the associated comment file, the hypertext document corresponding to the reviewer-selected source file and
- (e) the hypertext document including portions corresponding to one or more of the set of comments associated with the reviewer-selected source file, the hypertext data for each portion relating to a comment to be displayed being defined by the associated comment display object,
- 10 (f) the hypertext document selectively including hypertext links representing comment insertion objects, the hypertext links providing reviewers with forms for reviewer entry of comments,
- (g) the hypertext document selectively including hypertext data for calling the common gateway interface program for generating a hypertext document and the hypertext
- 15 document selectively including hypertext data for calling the common gateway interface program for accepting new comments,
- (h) means for communicating the hypertext document to a browser for display.

According to another aspect of the present invention there is provided an article of manufacture comprising:

- 20 a computer usable medium having computer readable program code means embodied therein for causing the storage and management of comments in a web-based file review system, the comments being from a plurality of reviewers, and being associated with one or more webs of source files, the computer readable program code means in the article of manufacture comprising

CA9-98-038

(a) computer readable program code means for causing a computer to parse a selected one of the set of source files into a linked list of objects corresponding to a hypertext representation of the selected source file, the linked list further comprising comment insertion objects and comment display objects, the parser writing the linked list of objects to a binary markup file representing the linked list of objects and corresponding to the selected one of the set of source files, each comment display object being capable of being associated with one or more comments,

(b) computer readable program code means for causing the computer to create and manage a set of comment files, each comment file being associated with a one of the set of source files and comprising data representing comments associated with the one of the set of source files,

(c) computer readable program code means for causing the computer to accept new comments for inclusion in the set of comments associated with a reviewer-defined source file and to update the associated comment file,

(d) computer readable program code means for causing the computer to generate a hypertext document from a markup file corresponding to reviewer-selected source file and from the associated comment file, the hypertext document corresponding to the reviewer-selected source file and

the hypertext document including portions corresponding to one or more of the set of comments associated with the reviewer-selected source file, the hypertext data for each portion relating to a comment to be displayed being defined by the associated comment display object,

the hypertext document selectively including hypertext links representing comment insertion objects, the hypertext links providing reviewers with forms for reviewer entry of comments,

CA9-98-038

the hypertext document selectively including hypertext data for calling the common gateway interface program for generating a hypertext document and the hypertext document selectively including hypertext data for calling the common gateway interface program for accepting new comments,

- 5 (e) computer readable program code means for causing the computer to communicate the hypertext document to a browser for display.

According to another aspect of the present invention there is provided a computer program product for use with a hypertext server, the computer program product comprising:

- 10 a computer usable medium having computer readable program code means embodied in the medium for causing the storage and management of comments in a web-based file review system, the comments being from a plurality of reviewers, and being associated with one or more webs of source files, the computer program product having:

- 15 (a) computer readable program code means for causing a computer to parse a selected one of the set of source files into a linked list of objects corresponding to a hypertext representation of the selected source file, the linked list further comprising comment insertion objects and comment display objects, the parser writing the linked list of objects to a binary markup file representing the linked list of objects and corresponding to the selected one of the set of source files, each comment display object being capable of being associated with one or more comments,

- 20 (b) computer readable program code means for causing the computer to create and manage a set of comment files, each comment file being associated with a one of the set of source files and comprising data representing comments associated with the one of the set of source files,

- (c) computer readable program code means for causing the computer to accept new

CA9-98-038

comments for inclusion in the set of comments associated with a reviewer-defined source file and to update the associated comment file,

- 5 (d) computer readable program code means for causing the computer to generate a hypertext document from a markup file corresponding to reviewer-selected source file and from the associated comment file, the hypertext document corresponding to the reviewer-selected source file and

10 the hypertext document including portions corresponding to one or more of the set of comments associated with the reviewer-selected source file, the hypertext data for each portion relating to a comment to be displayed being defined by the associated comment display object,

the hypertext document selectively including hypertext links representing comment insertion objects, the hypertext links providing reviewers with forms for reviewer entry of comments,

15 the hypertext document selectively including hypertext data for calling the common gateway interface program for generating a hypertext document and the hypertext document selectively including hypertext data for calling the common gateway interface program for accepting new comments,

- (e) computer readable program code means for causing the computer to communicate the hypertext document to a browser for display.

20 According to another aspect of the present invention there is provided a method of storing and managing a set of comments associated with a source file, in a file review system, the method comprising the steps of

- (a) accepting data from the source file and storing a representation of the source file as a markup file,

CA9-98-038

- (b) creating a comment file containing data representing the set of comments associated with the source file,
- (c) responding to user input to accept new comments for inclusion in the set of comments associated with the source file and updating the comment file to correspond to the complete set of comments,
- (d) responding to user input to dynamically generate a hypertext document from the markup file and from the comment file, the hypertext document corresponding to the source file and including portions corresponding to one or more of the set of comments associated with the source file,
- (e) communicating the hypertext document to a user for display.

According to another aspect of the present invention there is provided a program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for storing and managing comments in a web-based file review system, the method steps comprising the steps of the above method.

Advantages of the present invention include a review tool in which the only software required on the reviewer's computer is a browser that supports the level of hypertext generated by the CGI of the review tool (the system of the preferred embodiment uses HTML 3.2). The reviewer's browser can be running on any platform; the CGI of the system knows nothing about the reviewer's environment.

The CGI, and any upgrades or fixes to it, need to be installed only on the author's workstation. Reviewers do not need to install or upgrade anything; the only indication they may have that an upgrade has occurred is that new function is available. A department of authors can further reduce the impact of upgrading by setting up a dedicated server for all webs created in the department.

CA9-98-038

Because the CGI runs on any Intel-based workstation, technical writers, managers, developers and others can install the system of the preferred embodiment on their workstations, and allow others to review their webs.

- 5 The user can shut down and restart a server machine with minimal impact on users, because each CGI call is a separate invocation of the CGI program. No comments are lost, and links within the review documents that a user was viewing before the shutdown will work again as soon as the server is brought back up.

10 BRIEF DESCRIPTION OF THE DRAWINGS

The preferred embodiment of the invention is shown in the drawings, wherein:

Figure 1 is a block diagram showing the architecture of the review tool of the preferred embodiment;

- 15 Figure 2 is a flowchart showing a high-level representation of the operation of the review tool of the preferred embodiment.

In the drawings, the preferred embodiment of the invention is illustrated by way of example. It is to be expressly understood that the description and drawings are only for the purpose of illustration and as an aid to understanding, and are not intended as a definition of the limits of the invention.

20

CA9-98-038

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to Figure 1, there is illustrated in a block diagram view, the architecture of the preferred embodiment of the invention. In Figure 1, a server 10 is shown connected to browser 12 and browser 14. Server 10 is shown containing CGI 16, markup file 18 and comment file 20. The server 10, and browsers 12, 14 may be located on the same computer or on different computers, in which case the connection between browsers 12, 14 and server 10 may be achieved over an Internet connection, or a Local Area Network connection. In the preferred embodiment, the browsers are commercially available HTML browsers such as Netscape Navigator™. CGI 16 is a computer program located in the server and referenced by the appropriate cgi-bin designation for common gateway interfaces in HTML. CGI 16 accesses both markup file 18 and comment file 20.

The system of the preferred embodiment is implemented in HTML and in C++. HTML is a system that permits the creation of "hypertext", a way of presenting information online with connections between one piece of information, such that non-linear navigation of a single file or several files is possible. The users of the system must run browsers which are capable of reading HTML documents. The functionality of much of the system is provided using forms and links in predefined HTML files which are accessible by users of the system. Other aspects of the system, such as parsing input files, and retrieving data from markup file 18 and comment file 20, are implemented in C++ code. As will be appreciated by those skilled in the art, other markup languages that permit the generation of hypertext (with compatible browsers), and other programming languages may be used to implement the system of the invention.

The system of the preferred embodiment provides access to a collection of files for review and comment by one or more reviewers. A single collection of files is termed "a review" in the system of the preferred embodiment. A given review, or collection of files, may be considered to be itself a web, but as is set out below, the system will function for collections of files even if the files were

CA9-98-038

not created as webs in the sense of being designed originally to be mutually linked HTML files.

The system recognizes users as falling into one of the following classes, for any given review: Author, Reviewer, Reader, and Administrator.

5

In Figure 2, there is illustrated in a flow chart view, a high-level representation of the algorithm of the preferred embodiment.

10 Box 30 in Figure 2 represents the creation of a review by an author. Each author is able to create one or more reviews. The author creates a review by placing one or more review source files in a directory to which the server is provided access. The author is then able to use HTML forms provided by the system to specify that the source files are to be considered a review by the system and to specify the types of the various source files (for example, flat text, HTML, gif, jpeg, xbm files). The author is also able to specify the order in which the files will be displayed for users of
15 the review. Other steps may also be carried out by the author at this point and these will be described in further detail below.

20 It is possible for an author to create a review from a set of HTML files, or from a set of flat text files, or both. For word-processing applications or SGML applications, the system of the preferred embodiment requires the creation of HTML or flat text versions of the documents before creating a review. There is no conversion in the preferred embodiment from formats such as IBMIDDOC (trade-mark), Lotus WordPro (trade-mark), or Microsoft Word (trade-mark). However, these applications usually have built-in transforms or export utilities to generate either flat text or HTML

CA9-98-038

text files.

In Figure 2, box 32 represents the step of a reviewer accessing a particular selected review file from an author-defined review. If the review file has not yet been accessed by a reviewer, the system of the preferred embodiment carries out a series of steps as shown in boxes 36, 38, 40, 42, 44 in Figure 2. These steps are described in more detail below. The end result of these steps is to create a markup file 18 and a comment file 20, as shown in Figure 1, which are able to be accessed by CGI 16 to create an HTML file representing the review file, as is shown in box 46 in Figure 2.

The reviewer is able, using the mechanism set out below, to add comments to the review file (shown in box 48 of Figure 2). The reviewer may move to other files in the review (shown in decision box 50 of Figure 2). Although the flowchart of Figure 2 shows the completion of the reviewer's use of a particular review file as a sequential step following the addition of comments to a review file, it will be appreciated that the reviewer may move to another review file without adding comments to a first review file. The reviewer may either select another review file from the list of files contained in the review, or the reviewer may follow a link from a first review file to another review file. Similarly, the reviewer may exit the system itself by terminating the browser program or by using the browser to select a file outside the review.

As is apparent from the architecture of the system, as shown in the example of Figure 1, it is possible for several browsers to access the same review and review file, at the same time. The architecture of the system provides that comments may be accepted by different reviews at the same time. Similarly, authors may take action on the comments while reviewers are browsing and adding comments to the review.

CA9-98-038

On the first access of a file by a reviewer, the system performs the steps summarized in boxes 36, 38, 40, 42, 44 in Figure 2.

The first step carried out by the system of the preferred embodiment is the parsing of the input file (also referred to as the source file or source document). The parsing of the input file is carried out by the following process:

1. The contents of the file is read into a linked list of objects. The file is broken down into the following objects:

- HTML tags (e.g. <body>)
- HTML end tags (e.g. </p>)
- Whitespace between tags (e.g. a newline character between </p> and <p>)
- Text (Note that whitespace within a text object is part of the text object, whereas whitespace between two tags is a whitespace object.)
- HTML comments (e.g. <!--Test-->)

In the preferred embodiment, each HTML tag or end tag is assigned a unique numeric value in the linked list that identifies the tag. For example, any tag named <a>, regardless of attributes (,) is assigned the numeric value 1, while any tag <html> has the numeric value 23. By assigning a numeric value to each tag, the linked list can rapidly be searched by type of HTML tag. For example, a search for the first <p> tag in the file will be implemented by the process of iterating through all objects in the linked list, checking each tag object's value for the value associated with a <p> tag and returning the first match.

CA9-98-038

2. Check the file for HTML markup. In the preferred embodiment, for the file to be considered valid HTML, it must meet these conditions:

- At least 2/3 of the HTML tags found in the document must be recognized tags that are part of the HTML 3.2 standard,
- At least one of <html>, <head>, <title>, or <body> must be present, and
- There must be at least one HTML tag for every 1,000 bytes of text.

It is necessary for a file to match all three conditions because the file being processed may be a flat text file containing excerpts of HTML for illustration, or may be a program source file that processes HTML documents. Simply checking for <html> somewhere in the document would not be adequate. The above criteria will result in misidentification in some cases. Where the system is to be implemented in circumstances where it is likely that misidentifications will occur, the criteria for identifying a file as being HTML can be made more rigorous.

3. If the file is not found to be valid HTML, it is considered to be a flat text file. The linked list is modified to reflect an HTML document consisting of flat text by inserting HTML tags and a title before and after the text:

a. Objects for the following tags and the file's name are inserted in the linked list prior to the flat text object. Each tag is shown in square brackets:

[<html>] [<head>] [<title>] [filename] [</title>] [</head>] [<body>] [<pre>]

b. Objects for the following tags are added after the flat text object:

[</pre>] [</body>] [</html>]

The resulting linked list, if saved to disk and displayed by the program in a browser, would

CA9-98-038

look like a flat text file, displayed in a fixed-pitch font with line breaks exactly where they occur in the source file.

The result of the parsing of the input file is a linked list which reflects an HTML document. The next step shown in Figure 2 is the normalization of the file data as shown in box 38. The normalization of the data in the linked list is carried out by the following steps:

1. The linked list is scanned for text objects within the document body that are not enclosed within any HTML tags other than the <body> tag. Paragraph start and end tags are added to the start and end of such text objects. An example of an HTML file which requires normalization is the following:

```
<html><head><title>Bad HTML</title></head>
<body>
This should have been in a paragraph
<p>This already is</p>
</body></html>
```

The sentence "This should have been in a paragraph" is not within <p> or any other tags within the body, so a <p> is added after <body> and a </p> is added after "paragraph".

2. Certain tags that may interfere with the proper functioning of the CGI application, such as some <meta> tags that force a document refresh after a specified period, or prevent a document refresh altogether, are removed. If left in the document, these meta tags might cause the file to display improperly in a CGI environment.

3. Certain HTML-defined symbols, such as the string . which represents a period, are converted to their text equivalents. This is done for any symbol that might be construed as an end-of-sentence indicator, such as a period, exclamation point, question mark, or colon. The change is made so that adding comment insertion markers after each sentence can be more easily accommodated later.

CA9-98-038

4.Unnecessary HTML tags that are commonly inserted by certain HTML-generating tools are removed. These tags often clutter up an HTML file and make parsing more difficult. An example of such HTML code is a list marked up as follows:

```

5      <dl>
        <p>
        <dt>Term</td>
        <p>
        <dd>Definition text</dd>
        </dl>

```

10 This list is converted during normalization, by removing the unnecessary <p> tags, to result in the following code:

```

15      <dl>
        <dt>Term</td>
        <dd>Definition text</dd>
        </dl>

```

5.Matching end tags are added for any start tags that are not properly closed. For example, in HTML it is allowable to indicate the end of one paragraph or list item simply by starting the next one:

```

20      <p>One two
        <p>Buckle my shoe
        <ul>
        <li>Three four
        <li>Knock at the door
25      </ul>

```

Since incomplete HTML makes automated parsing more difficult, end tags are added as follows:

```

        <p>One two</p>
        <p>Buckle my shoe</p>

```

CA9-98-038

```

<ul>
<li>Three four</li>
<li>Knock at the door</li>
</ul>

```

- 5 This process adds end tags for paragraphs (<p>), list items (<dt>, <dd>,), examples (<pre>), high-level tags (<html>, <head>, <title>, <body>), and heading tags (<h1>...<h6>).

As the above indicates, the normalization process is carried out on the file representation found in the linked list data structure first created in the parsing stage (of block 36 in Figure 2). The use of a linked list permits the efficient insertion and deletion of objects during the normalization step. The resulting data structure represents an HTML document which will be displayed as a text equivalent to the source file. The normalization step adds objects and deletes objects to allow comment insertion and display to be carried out more efficiently and effectively.

10 Once the file has been read into a linked list and normalized, the linked list is manipulated in a series of steps that add comment insertion markers and comment display objects at key points (this is represented in Figure 2 by box 40). The comment insertion markers and comment display objects are each objects which are added to the linked list representation of the HTML document which represents the input file, or source document.

15 A comment insertion marker is used by the system of the preferred embodiment to permit the display of a hypertext link that allows a reviewer to add a comment at a particular point. A comment display object is used by the system of the preferred embodiment to determine where, and how, to display comments that were added at a specific point.

20 The following describes how the step of box 40 in Figure 2 is carried out:

1. Any text object that is stored between a <pre> tag and its matching </pre> tag is split into several objects:

- 25 • A comment insertion marker at the start of each line

CA9-98-038

- A text object for each line
- A whitespace object for the carriage return at the end of each line
- A comment display object at the end of each line

- 5 2.A comment insertion marker is added at the start of each paragraph-level HTML tag, such as <p>, , <dt>, and headings (<h1>...<h6>), and a comment display object is added after the matching end tag for each of the above.
- 10 3.Certain regions of the linked list may now contain comment insertion markers and comment display objects that do not belong. For example, there may be markers within a table because the table contains lists or paragraphs. Or there may be a set of nested lists in which markers were inserted in a fashion that will make displaying comments difficult. These markers and display objects are removed.
- 4.Markers and display objects are added to each row in a table.
- 15 5.Certain combinations of markers and display objects are rectified so that there is a one-to-one correspondence between them. For example, a pattern such as [M][C][M][M][C] may develop (where M is a marker and C is a comment display object). This is changed by removing one of the two consecutive markers, with a resulting pattern of [M][C][M][C]. Note that there are intervening text and tag objects between these marker and display objects that are not shown in this example.
- 20 6.The entire linked list is scanned sequentially, and a record of what tags are currently open is stored in a variable. Each time a comment display object is encountered, the currently open tags are saved into that comment display object, so that, if a comment later gets displayed at that location, the system of the preferred embodiment can determine what HTML tags to close before displaying the comment and what tags to open after the comment.

CA9-98-038

7. For list items in ordered lists, a start attribute must also be added so that, if a list is closed by a comment display object because a comment is being shown there, the next item in the list starts at the right number. An example of how a list is converted is shown for the following list:

5
 [M]One[C]
 [M]Two[C]

This list is converted to the following:

10
 <li start=1>[M]One[C]
 <li start=2>[M]Two[C]

15 Without this added attribute, if a comment were added to the first list item, the resulting HTML would display as two single item lists ("One" and "Two") both starting with the number 1, interleaved with the comment text. To avoid this, the list is re-written in the linked list to maintain the ordering of the source document, despite interleaved text additions for the comments. This is because the ordered list gets closed before the comment and reopened after it.

20 The linked list now consists of a set of text, tag, end tag, comment, whitespace, marker, and comment display objects that can be used to generate views of the document dynamically, with comments being shown wherever they are added and all tags properly closed.

As shown in box 42 of Figure 2, the linked list is written to a "markup" file, which is a binary file

CA9-98-038

that can later be read to quickly recreate the linked list. Characteristics of the markup file are as follows:

- For each object, its type, length, and content are stored in a sequence of one or more bytes.
- An object's type (HTML comment, tag, etc.) is identified by the top four bits of its first
5 byte. This allows for sixteen different object types, of which the following are used:

- Simple tag - a tag object with no attributes (e.g. <html>)
- Tag - a tag object with attributes (e.g. <body bgcolor="#FFFFFF">)
- End tag
- Text
- 10 ● Comment Insertion Marker
- Comment Display
- Complex whitespace (combinations of spaces, newlines, tabs, etc.)
- Series of spaces
- Series of newlines

- 15 ● HTML Comment

- Ordered list item tag (different from regular tag so that display functions know when to display the start attribute on the list item if a comment is inserted)

- Anchor tag used for links. (Different from regular tag so that, when the file is
20 displayed, the system of the preferred embodiment knows to add its own CGI
information to the link that gets generated. This allows links from one document to

CA9-98-038

another to invoke the system of the preferred embodiment to request the file.)

●An object's length is identified by the bottom four bits of its first byte if the length is under 15 bytes, or by the next one or two bytes if the length is over 15 bytes.

●An object's content is stored, to the extent required. For HTML tags, the leading "<" is not included because the object's type, "Tag", presupposes a starting "<". For HTML end tags, the leading "</" characters are not included because the object's type, "End Tag", presupposes a starting "</".

●Because a typical HTML start or end tag is shorter than 15 characters, the typical overhead of storing the linked list, versus storing the raw HTML, is negligible. For example, the 6-character text string "<html>" becomes a six-character binary string in the markup file, broken down as follows:

●First four bits of first byte identify the object as a Tag

●Last four bits of first byte identify the length as 6 characters

●Remaining bytes are "html>", since the leading "<" is not needed given that the object is known to be a Tag.

●For end tags, there is even a space savings, as almost all HTML tags are shorter than 15 characters and the "</" can therefore be converted into a single byte.

●For multi-character objects consisting entirely of spaces or entirely of newlines, the entire whitespace can usually be represented in one or two bytes. For example, a sequence of 12 spaces is represented in one byte as follows:

●The top 4 bits identify the object as a set of 1 to 16 spaces.

●The bottom 4 bits identify the number of spaces, from 1 to 16.

CA9-98-038

- For comment insertion markers and comment display objects, the numeric value (from 1 to N) of the object is stored. In most cases, a given marker object shares the same numeric value as the next comment display object that follows it.

5 The markup file record structure is such that a typical markup file is within 10% of the size of the HTML source file it was generated from, and in some cases is smaller. This expected size is achieved despite the overheads of:

- 64 bytes of header information at the start of the markup file
- The addition of comment insertion markers and comment display objects, including the display objects' open tags and the numeric values for both markers and display objects
- The addition of HTML tags required for normalization

15 Once the markup file is written, the system of the preferred embodiment also generates an empty comment file (the step of box 44 in Figure 2). This file contains a header record with the file's title and other information. When comments are added by reviewers, they are appended to the end of this comment file.

20 The comment file contains variable-length records for each comment. Each comment record contains data representing the comment status, type, disposition, user number of the user creating the comment, date (in seconds format), comment text length, line number, comment number on this line, and the text of the comment. The comment record also has pointers to the logical next and previous comments.

CA9-98-038

Order is maintained through these pointers, within each comment record, to the logical previous and next records. Thus, when a comment is inserted at line 12, for example, and the immediately preceding comment is on line 9, with the immediately following one on line 17, the new comment gets added to the end of the comment file, its Previous pointer points to line 9, and its Next pointer points to line 17. The Next pointer of the comment on line 9, and the Previous pointer for the comment on line 17, are also updated to point to the new comment. By iterating through the file starting with the virtual first record and using each record's Next to determine what record to read next, sequential order can be preserved without requiring constant resorting either at comment addition time or file display time.

Turning again to the architecture of the system as shown in Figure 1, the way in which reviewers access a file in a given review may be described. A reviewer uses a browser to view an HTML document made available by the server. The HTML document contains a list of different reviews. The reviewer selects a particular review from the HTML page displayed by the reviewer's browser. The server then displays (by another HTML document) the files available for review by that reviewer in the selected review. The reviewer selects a file. If the file has been accessed by a reviewer before, a markup file and a comment file will have been created. If not, the two files are created, as is described above. By selecting a file to be displayed, the HTML document displayed by the browser of the reviewer, gives rise to a CGI call to the C++ code resident in the server. This in turn translates the markup file data into the linked list which corresponds to the HTML markup for the source document. The link list is then translated into an HTML document which is then sent to the browser of the reviewer. From the perspective of the reviewer seeing the document as displayed by the HTML browser, this document corresponds to the source document, except that the document contains comment insertion markers.

CA9-98-038

From the perspective of reviewers, the system will function as follows:

1. Each reviewer logs on to system and selects a review from the list of reviews shown to the reviewer.
2. Once the reviewer has selected a review, the reviewer can select a page (or file) within the review, and begin reading. The page will contain numbered links at the start of every sentence, paragraph, list item, and unformatted text line.
3. At any part of the document where the reviewer wants to request a change or make a suggestion, the reviewer clicks on the numbered link closest to the pertinent sentence or line.
4. A comment form appears, allowing the reviewer to type in a comment and describe the priority level of the suggestion or change by selecting a comment type (see below). The reviewer can also choose to send a message to the author or to another reviewer to draw their attention to the comment.

Thus from the perspective of the reviewer, the system is simple and intuitive to use. The system also operates in close to real time.

Because of the efficient storage structure of a markup file, it can be read very quickly back into a new linked list each time the document is requested, and an output can be generated for the linked list. By using a linked list both to create the markup file, and to display views of it each time the document is requested, the system of the preferred embodiment can perform tasks such as dynamically inserting comments and numbered markers, or adding informational messages at the start or end of the body of the file, with minimal overhead and almost no text searching.

Another feature of the markup handling of the system of the preferred embodiment is a sections feature, which allows users to view only a portion of a large file at one time. The sections feature is

CA9-98-038

only activated for a markup file when a user who has chosen to view a file in sections first accesses the file. The following algorithm is used when such a user requests a file:

1.If the markup file contains fewer than a threshold number of comment insertion markers (e.g. 40), the markup file header is modified to indicate that the document is not eligible for division into sections. The entire file is displayed and this algorithm is abandoned.

2.If the markup file is eligible for sectioning, the system of the preferred embodiment scans the document to determine how frequently heading tags occur. As referred to above, this can be done quickly because the markup file is stored in a linked list. If heading tags occur on average at least once for every 40 comment insertion markers, the system of the preferred embodiment uses a heading strategy to section the file. Otherwise the system of the preferred embodiment uses an insertion marker strategy.

3.For either strategy, the system of the preferred embodiment creates a sections file, associated with the marker file. This sections file contains records identifying the starting and ending offsets in the markup file for each of the following sections:

- The header section: Everything from <html> up to and including the <body> tag. This section is always part of the HTML that gets sent back to a user's browser.
- The body sections
- The footer section: Everything from </body> to the end of the file. This section is always part of the HTML that gets sent back to a user's browser.

4.The heading strategy involves breaking the body sections down every time:

- (a) A heading is found, AND
- (b) The last section starting boundary is at least five comment insertion markers before

CA9-98-038

the current heading.

5. The insertion marker strategy involves breaking the body sections down every time:

- (a) At least 40 comment insertion markers have occurred since the last section starting boundary, AND
- 5 (b) The current comment display object has no Open Tag values associated with it. This is used to guarantee, for example, that a section does not start or end in the middle of a list or table.

10 Once the sections file is created, the system of the preferred embodiment determines how many sections to display of the file at one time, based on the user's section-viewing preferences. It then creates a linked list consisting of the header section, the required body sections, and the footer section. Only those portions of the markup file that are required to create the requested sections are read; other parts are scanned over. This makes it much faster to display a section of a large file, than would be accomplished by reading the entire markup file into a linked list and then splicing out unneeded sections.

15

As is set out above, the system of the preferred embodiment employs a linked list to represent the files in a review. A linked list is a standard computational device for creating an easily manipulated, sequential set of objects. The standard doubly-linked list has the following characteristics and functions:

20

- Each object contains a data portion and pointers to its previous and next elements
- Functions are provided to iterate forwards or backwards through the list to the next or prior object, or the first or last object.

CA9-98-038

- Functions are provided to add an object or another linked list of objects to a particular object in the list. The attachment can occur immediately before or after the selected object, or at the start or the end of the list

- Functions are provided to remove one or more objects from the list.

5 The linked list implementation which the system of the preferred embodiment uses in markup files includes the following enhancements:

- Different objects in the linked list are actually different-typed objects that are all derived from a common ListElement class. Thus the linked list can consist of tags, end tags, text objects, markers, etc. all linked together.

10 ●Functions are provided that allow each object to be tested for its type and contents.

- Functions are provided to move forwards or backwards in the linked list to the next or prior object of a given type or of the same type as the current object. This allows for fast searches for particular objects, such as a start tag with a data value of "<p>". For any object that is not a start tag, the string representation of the object is never parsed, because the object's type does not match the requested type.

15

The algorithm's pseudocode is as follows:

```

Function: getNextOfType(T)
Assign the current object's Next pointer to a
temporary object X
20 while (X is valid, AND
(X does not match the type of T, OR
X does not match the contents of T))
Assign X's Next pointer to X
return X

```

25 The performance benefit is achieved because the check for a type match ("X does not match the type of T") is performed before the content match ("X does not match the contents of T"),

5

10

15

20

- Functions are provided for specific types of objects, to manipulate the objects' contents or display the object. For example, functions are provided to set, read, or change attributes on Tag objects, so that the user can determine the value of the bgcolor attribute of a <body> tag. Functions are also provided to display each object when the HTML for a document is

CA9-98-038

displayed.

5 As is referred to above, each time a user requests a file in a review (a document), the system of the preferred embodiment reads in the markup file (or those sections that apply, if the user has asked to view a large file in sections), and opens the comment file for the document. The system dynamically generates the HTML corresponding to the file (or section) which has been requested by the user.

10 Before displaying anything back to the user (i.e. before forwarding an HTML document for display by the browser of the user), the system of the preferred embodiment searches for the first comment in the comment file that matches the filter settings of the user. Filter settings are described in further detail below. For example, a filter can be created so that a user can request that only comments created after a certain date be displayed. The system determines the line number of this comment.

15 The generation of an HTML document which will be forwarded to the user's browser for display proceeds as follows, the system travelling the linked list:

- For each object reached in the linked list, determine its type.
- Call the display function for that type. For most object types this simply means writing out to the HTML document what was read in from the original markup file. The system will, for example, add the appropriate HTML tag for objects defined to have a particular HTML tag type. The contents of the tag, if any will generally correspond to the contents of the object in the linked list. However, for link objects, additional work may be required to add CGI call arguments to the link, and for links that are tags, the link to the image needs to be updated to point to the image CGI that the system of the preferred embodiment uses

CA9-98-038

to display images.

- If the type of the object in the linked list is a comment display type, check its line number.

If the line number corresponds to the next comment to display, do the following:

- Place into the HTML document the matching closing tags for each of the tags stored in the comment display object's OpenTags attribute. Closing tags are displayed in reverse order. This turns off any tags that are currently active.

- Insert the comment in the HTML document.

- Scan the comment file for the next comment and determine its line number. While the line number is the same as the current line number, insert the comment in the HTML document.

- Insert into the HTML document the tags in the OpenTags attribute to turn the necessary tags back on.

The above describes the system of the preferred embodiment as it relates to the creation of collections of files for review. The system permits reviewers to add comments to dynamically generated HTML documents which will retain the original formatting of the source documents but which will permit reviewers to add comments which will be displayed within the documents of the review for the author and for other users of the system. As will be appreciated from the above description, the HTML document which is generated by the system will include within it CGI calls such that most actions taken by a reviewer or other user will result in the regeneration of the document. As a result of the architecture of the system, newly added comments are displayed on the next regeneration of the document. This results in the system behaving in a "real time" manner, where comments being added by multiple reviewers are available for the reviewers, and the author, almost as soon as they are added to the file.

CA9-98-038

The system also permits the reviewer to move between different files in a review, should the files contain links to each other. The system also supports the reviewer following links from files in the review to files outside the review, although it is not possible to include comments in files which are outside the review.

5

A further feature of the system which arises from its design as set out above is that reviews may be created incrementally. The review defined by an author may be added to at any time without interfering with the previously defined review. In this way a review may be made available to reviewers in stages, or releases. The newer files in the release may use links to files included in the review previously or the newer files may be independent of the previously included files.

10

The system of the preferred embodiment also includes other features which are made possible by the representation of the documents in the review.

15

Comments in the system of the preferred embodiment are assigned a severity (called a "type") and a current status (called a "disposition"). Default types and dispositions are provided by the system but an author may modify the types and dispositions for each review. A comment's type may be changed. A comment's disposition is expected to be changed to reflect the action taken on the comment by the author, for example to indicate that the comment has been entered into the source document.

20

The following four types of comments are available as the default in the system of the preferred embodiment:

CA9-98-038

- Typo: A typing error in code, text or markup
- Comment: Simple comments that may not be necessary fixes
- Problem: Something is wrong: either the text cannot be understood, or the information in this part of the text is inaccurate
- 5 ●Issue: Something is wrong on a larger scale. Problems in the text may indicate a larger misunderstanding of information, or may point to other areas where more work is needed before the text can be corrected.

The following comment dispositions are made available by the system of the preferred embodiment:

- 10 ●Accepted: the author has read the comment and made the changes recommended.
- Rejected: the author has read the comment but is not going to make the recommended changes
- Open: the author has read the comment but has not resolved the problem because more information is needed, or, the author accepted or rejected the comment but a reviewer disagrees with how the comment was handled, and so re-opens the comment.
- 15 ●Minor: the author has accepted the comment but it was something minor, like a typo.
- Suggestion: the author agrees with the reviewer's comment, but cannot make the change for this draft, or there are more important issues to deal with first.
- Duplicate: the comment has already been made somewhere else in the review or file by the reviewer or by another reviewer.
- 20 ●Unassigned: new comments are automatically given an unassigned disposition until an

CA9-98-038

author for the review, or an administrator, changes their disposition to indicate that they have read the new comments and decided what to do with them.

5 The system permits users to filter the files in a review. The filters are user-selectable and so the users may change the filters to see different sets of comments for the same document.

10 The user can tell the system to show the user only certain types of comments by filtering. The user can filter out comments with certain dispositions, comment types, comments created before a specific date or modified after another date, comments created or appended by particular users, or comments containing particular text. For example, if the user wants to see only the pages containing comments the user has not yet addressed, the user can select a filter to hide all the comments that have a disposition of "Accepted".

15 The system also permits each review to have a predefined access restrictions associated with it. These are referred to as authority levels.

20 The system implements authority levels at the server level and for each review. There are four levels of authority: Reader, Reviewer, Author, and Administrator. Each user is defined to have a level of authority at the server level. The server level authority may be thought of as a default authority. A user's authority at the server level does not, however, guarantee that the user will have the same level of authority within a particular review. For example, a user who has a default or server authority of Reader can be made an Author for a specific review. Similarly, a user with Author authority may be prevented from doing some Author-level tasks if a review's access is restricted.

CA9-98-038

The system default for each level of authority entitles a user to do the following:

- Readers can read the contents of the review, search and filter comments, and view statistics, but cannot add or respond (append) to comments.

5

- Reviewers can do what readers can do, but can also add or append to comments; they can edit or delete their own comments or appended comments, provided theirs is the most recent comment; and they can mark a file as read.

10

- Authors can do what reviewers can do, can change the disposition of comments, and can edit or delete anyone's comments. They can also add releases (new groups of files) to a review and delete reviews they created; they can change the cutoff dates (see below) for reviews they own; and they can change the access privileges of other users to their reviews.

- Administrators can do everything, including creating and deleting reviews, and creating and deleting users.

15

Each review has an access level. The default access level determines what users can do in the review if they have not been granted special individual access. In the list below, the authority levels Reader through Administrator refer to a user's authority at the server level.

- In a Read-Only review, all users can read the review, but none can enter comments.

- In a Restricted Editing review, all users can read the review, but only Reviewers and Authors can enter comments.

- In an Open Editing review, all users can read the review and enter comments.

20

- In a Private review, no one can read the review or enter comments unless they are individually authorized to do so.

However, in addition to the default access levels, individual users' access to a review can be

CA9-98-038

increased or decreased, regardless of their default authority level. For example, the system can prevent a certain user from accessing an Open Editing review, even if he or she has a default, or server, authority level of Author; or the system can make a Reader an Author for a Private review.

Outside of a review, users whose server-level authority is Reader, Reviewer, or Author all have the same capabilities, except that the following tasks can only be performed by users with a server-level authority of Author:

- Creating new reviews.
- Deleting reviews for which the user is an author.
- Creating new users. New users created by an Author can only have a default authority of Reader (only an Administrator can change the authority level).

The system of the preferred embodiment includes the ability to impose cutoff dates on comments being added to releases of files in a review. This permits reviews to have releases which have been finalized, but which are related to releases which are continuing to undergo modifications and for which comments continue to be accepted.

To change the cutoff date of a release a user must be an author for the review, or an administrator. Cutoff dates are entered in the system of the preferred embodiment in the following way:

- 1.The user enters the review in which a cutoff date is to be defined.
- 2.The user is provided with a list of releases and their cutoff dates (if any). If the user is an author for the review, or an administrator, the cutoff dates are shown as hypertext links. The user clicks on the date location to be entered.

CA9-98-038

3.The user fills in the resulting form to enter a new date.

5 Reviews themselves do not have cutoff dates. Cutoff dates displayed in the list of reviews are the latest new-comment cutoff dates for any release within those reviews. To change the cutoff dates for an entire review, the cutoff dates must be changed for every release in the review.

10 The system also permits statistics relating to a review to be simply and powerfully displayed for users. The system provides a Statistics page that is accessible from any page in a review. It displays a table showing comments sorted by disposition and type, and a table showing comments sorted by user.

15 Statistics in the Statistics page are, for appropriate statistics, represented as links to lists of the files containing matching comments. These links provide an efficient way of performing simple searches for comments in files in a review. It is possible for a user to narrow down all files and comments in a review to a list of comments of a specific type or from an individual reviewer.

20 The Statistics page is dynamically generated each time a user accesses it, so any changes made to comments after viewing statistics will be reflected in the summaries the next time the Statistics page is viewed.

As long as the review documents remain in the server's directory structure, statistics can be generated for the review even after all cutoff dates have passed.

CA9-98-038

As will be apparent from the description of the system set out above, it is possible to navigate through files in a review by moving sequentially through the files, by following links found in the files, or by navigating through the comments which are included in the files of the review. Users may also select how a file is to be displayed. Comment insertion markers may be turned off so that
5 they are not displayed for a particular file and comments themselves may be displayed in different formats, as defined by the system and the user.

The features described above follow from the system design which permits the dynamic generation of the HTML document for communication to the browser of the user. This dynamic generation,
10 in which the comments are stored separately from the representation of the document itself, results in the ability for the user to customize how the document appears or how it may be navigated.

Although a preferred embodiment of the present invention has been described here in detail, it will be appreciated by those skilled in the art, that variations may be made thereto, without departing
15 from the spirit of the invention or the scope of the appended claims.

CA9-98-038

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR
PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

- 1 1. A file review system for storing and managing a set of comments associated with a source file,
2 comprising
3 means for accepting data from the source file and storing a representation of the source file as
4 a markup file,
5 means for creating a comment file containing data representing the set of comments associated
6 with the source file,
7 means for accepting new comments for inclusion in the set of comments associated with the
8 source file and for updating the comment file to correspond to the complete set of comments,
9 means for generating a hypertext document from the markup file and from the comment file, the
10 hypertext document corresponding to the source file and including portions corresponding to one
11 or more of the set of comments associated with the source file,
12 means for communicating the hypertext document to a user for display.
- 1 2. The file review system of claim 1 in which the means for accepting data from the source file and
2 storing the source file as a markup file further comprises means for representing the source file
3 as a linked list of objects corresponding to a hypertext representation of the source file, wherein
4 the markup file is a binary file representation of the linked list.
- 1 3. The file review system of claim 2 in which the means for representing the source file as a linked
2 list of objects further comprises means for inserting comment insertion objects into the linked list
3 of objects and in which the means for generating a hypertext document further comprises means to
4 include hypertext data in the hypertext document to display one or more symbols representing each
5 comment insertion object.

CA9-98-038

1 4. The file review system of claim 3 in which the one or more symbols representing each comment
2 insertion object comprise a hypertext link to a hypertext form for user input of a comment.

1 5. The file review system of claims 3 or 4 in which the means for representing the source file as
2 a linked list of objects further comprises means for inserting comment display objects into the linked
3 list of objects, each comment display object being associated with a comment insertion object and
4 in which the means for generating a hypertext document further comprises means to include
5 hypertext data in the hypertext document to display comments in accordance with the corresponding
6 comment display objects.

1 6. The file review system of claim 5 in which each one of the comment display objects comprises
2 open tag data representing the open hypertext tags, at the location of the comment display
3 object, in the linked list of objects
4 and in which the means for generating a hypertext document further comprises
5 means to insert closing hypertext data in the hypertext document, prior to a selected one
6 of the comments being displayed in the hypertext document, and means to insert
7 reopening hypertext data in the hypertext document, following the selected one of the
8 comments being displayed in the hypertext document,
9 the closing hypertext data and the reopening hypertext data both being defined by the
10 open tag data for the comment display object associated with the selected one of the
11 comments being displayed.

1 7. The file review system of claim 2 in which the means for representing the source file as a linked
2 list of objects further comprises a means for normalizing the linked list of objects whereby
3 objects representing text are enclosed by objects representing paragraph tags,
4 end of sentence indicators in objects in the linked list are represented in text,

CA9-98-038

5 objects representing hypertext end tags are selectively located in the linked list to close all
6 hypertext tags, and
7 objects representing unnecessary hypertext tags are removed from the linked list.

1 8. The file review system of claim 2 in which the linked list of objects comprises objects
2 appropriately selected from the set of objects comprising simple tag objects, tag objects, end tag
3 objects, text objects, comment insertion objects, comment display objects, complex whitespace
4 objects, spaces objects, newlines objects, hypertext comment objects, ordered list item tag
5 objects, and link tag objects.

1 9. The file review system of claim 1 in which the means for generating a hypertext document
2 comprises a common gateway interface computer program.

1 10. The file review system of claim 1 in which the means for generating a hypertext document
2 further comprises means to generate a hypertext document which corresponds to a predefined
3 subsection of the markup file and the associated subsection of the comment file.

1 11. The file review system of claim 2 in which
2 each object in the linked list of objects has a type, and comprises contents derived from a
3 portion of the source file,
4 each comment insertion object and each comment display object comprises a contents line
5 identifier, representing the line number defining the location of the object in relation to the
6 source file,
7 the comment file consists of comment line identifier data associated with the set of
8 comments representing the line numbers in the source file with which each comment is
9 related,
10 and in which the means for generating a hypertext document further comprises

CA9-98-038

11 means to traverse the linked list of objects comprising, and means to select a display
 12 object from the linked list of objects,
 13 means to determine the type of the display object,
 14 means to generate hypertext data to correspond to the type of the display object and the
 15 contents of the display object,
 16 means to identify the contents line identifier for the display object and for selecting
 17 comment display data from the comment file based on the contents line identifier and the
 18 comment line identifiers of the comment file,
 19 means to generate hypertext data from the comment file representing the selected one of
 20 the set of comments corresponding to the comment line identifier associated with the
 21 contents line identifier.

1 12. The file review system of claim 1 in which each comment in the set of comments has
 2 associated with it a severity and in which the comment file consists of data representing the severity
 3 of each comment, and in which the means for accepting new comments for inclusion in the set of
 4 comments associated with the source file and for updating the comment file to correspond to the
 5 complete set of comments, comprises means for accepting data representing the severity of the new
 6 comments, and means for updating the comment file to add data representing the severity of the new
 7 comments.

1 13. The file review system of claim 2, in which the means for generating a hypertext document
 2 comprises means for selectively excluding portions of the comment file corresponding to one or
 3 more comments in the set of comments, whereby a selectively defined subset of the set of comments
 4 is represented in the hypertext document.

1 14. The file review system of claim 13, in which comments comprise comment attributes, the
 2 means for selectively excluding portions of the comment file further comprising means for

CA9-98-038....

3 selectively defining the subset of the set of comments by selection based on comment attributes.

1 15. The file review system of claim 14 in which the comment attributes comprise one or more of
2 comment type, comment severity, comment date, comment status, comment author, and comment
3 content.

1 16. The file review system of claim 1 in which system users are uniquely identified in the system
2 and in which the means for accepting new comments further comprises means for selectively
3 restricting the inclusion of new comments in the set of comments based on the unique identity of
4 system users.

1 17. The file review system of claim 1 in which the source file is associated with a defined cutoff
2 date and in which the means for accepting new comments further comprises means for selectively
3 restricting the inclusion of new comments on dates past the defined cutoff date.

1 18. The file review system of claim 1 further comprising means to navigate the comment file to
2 provide user access to comments in user-defined sequenced order.

1 19. The file review system of claim 1 further comprising means to collect and display statistics
2 relating to the set of comments.

1 20. The file review system of claim 19 in which the means to collect and display statistics further
2 comprises means for displaying hypertext links in the statistics display whereby a user may access
3 a selected subset of the set of comments by means of the hypertext link.

1 21. A web-based file review system for storing and managing comments from a plurality of
2 reviewers, the comments being associated with one or more webs of source files, comprising

CA9-98-038

3 a parser to parse a selected one of the set of source files into a linked list of objects
4 corresponding to a hypertext representation of the selected source file, the linked list further
5 comprising comment insertion objects and comment display objects, the parser writing the
6 linked list of objects to a binary markup file representing the linked list of objects and
7 corresponding to the selected one of the set of source files, each comment display object
8 being capable of being associated with one or more comments,

9 a set of comment files, each comment file being associated with a one of the set of source
10 files and comprising data representing comments associated with the one of the set of source
11 files,

12 common gateway interface program code means for accepting new comments for inclusion
13 in the set of comments associated with a reviewer-defined source file and for updating the
14 associated comment file,

15 common gateway interface program code means for generating a hypertext document from
16 a markup file corresponding to reviewer-selected source file and from the associated
17 comment file, the hypertext document corresponding to the reviewer-selected source file and

18 the hypertext document including portions corresponding to one or more of the set of
19 comments associated with the reviewer-selected source file, the hypertext data for each
20 portion relating to a comment to be displayed being defined by the associated comment
21 display object,

22 the hypertext document selectively including hypertext links representing comment
23 insertion objects, the hypertext links providing reviewers with forms for reviewer entry
24 of comments,

CA9-98-038

25 the hypertext document selectively including hypertext data for calling the common
26 gateway interface program for generating a hypertext document and the hypertext
27 document selectively including hypertext data for calling the common gateway interface
28 program for accepting new comments,

29 means for communicating the hypertext document to a browser for display.

1 22. The system of claim 21, in which each one of the comment display objects comprises
2 open tag data representing the open hypertext tags, at the location of the comment display
3 object, in the linked list of objects,

4 and in which the common gateway interface program code means for generating a hypertext
5 document further comprises

6 means to insert closing hypertext data in the hypertext document, prior to a selected one
7 of the comments being displayed in the hypertext document, and means to insert
8 reopening hypertext data in the hypertext document, following the selected one of the
9 comments being displayed in the hypertext document,

10 the closing hypertext data and the reopening hypertext data both being defined by the
11 open tag data for the comment display object associated with the selected one of the
12 comments being displayed.

1 23. The system of claim 21 in which the linked list of objects comprises objects appropriately
2 selected from the set of objects comprising simple tag objects, tag objects, end tag objects, text
3 objects, comment insertion objects, comment display objects, complex whitespace objects, spaces
4 objects, newlines objects, hypertext comment objects, ordered list item tag objects, and link tag

CA9-98-038

5 objects.

1 24. The system of claim 21 in which

2 each object in the linked list of objects has a type, and comprises contents derived from a
3 portion of the source file,

4 each comment insertion object and each comment display object comprises a contents line
5 identifier, representing the line number defining the location of the object in relation to the
6 source file,

7 the comment file consists of comment line identifier data associated with the set of
8 comments representing the line numbers in the source file with which each comment is
9 related,

10 and in which the common gateway interface program code means for generating a hypertext
11 document further comprises

12 means to traverse the linked list of objects, and means to select a display object from the
13 linked list of objects,

14 means to determine the type of the display object,

15 means to generate hypertext data to correspond to the type of the display object and the
16 contents of the display object,

17 means to identify the contents line identifier for the display object and for selecting
18 comment display data from the comment file based on the contents line identifier and the

CA9-98-038

19 comment line identifiers of the comment file, and

20 means to generate hypertext data from the comment file representing the selected one of
21 the set of comments corresponding to the comment line identifier associated with the
22 contents line identifier.

1 25. An article of manufacture comprising:

2 a computer usable medium having computer readable program code means embodied therein for
3 causing the storage and management of comments in a web-based file review system, the
4 comments being from a plurality of reviewers, and being associated with one or more webs of
5 source files, the computer readable program code means in the article of manufacture comprising

6 computer readable program code means for causing a computer to parse a selected one of the
7 set of source files into a linked list of objects corresponding to a hypertext representation of
8 the selected source file, the linked list further comprising comment insertion objects and
9 comment display objects, the parser writing the linked list of objects to a binary markup file
10 representing the linked list of objects and corresponding to the selected one of the set of
11 source files, each comment display object being capable of being associated with one or
12 more comments,

13 computer readable program code means for causing the computer to create and manage a set
14 of comment files, each comment file being associated with a one of the set of source files and
15 comprising data representing comments associated with the one of the set of source files,
16 computer readable program code means for causing the computer to accept new comments
17 for inclusion in the set of comments associated with a reviewer-defined source file and to
18 update the associated comment file,

CA9-98-038

19 computer readable program code means for causing the computer to generate a hypertext
20 document from a markup file corresponding to reviewer-selected source file and from the
21 associated comment file, the hypertext document corresponding to the reviewer-selected
22 source file and

23 the hypertext document including portions corresponding to one or more of the set of
24 comments associated with the reviewer-selected source file, the hypertext data for each
25 portion relating to a comment to be displayed being defined by the associated comment
26 display object,

27 the hypertext document selectively including hypertext links representing comment
28 insertion objects, the hypertext links providing reviewers with forms for reviewer entry
29 of comments,

30 the hypertext document selectively including hypertext data for calling the common
31 gateway interface program for generating a hypertext document and the hypertext
32 document selectively including hypertext data for calling the common gateway interface
33 program for accepting new comments,

34 computer readable program code means for causing the computer to communicate the
35 hypertext document to a browser for display.

1 26. The article of manufacture of claim 25, in which each one of the comment display objects
2 comprises
3 open tag data representing the open hypertext tags, at the location of the comment display
4 object, in the linked list of objects,

CA9-98-038

5 and in which the computer readable program code means for causing the computer to
6 generate a hypertext document further comprises

7 means to insert closing hypertext data in the hypertext document, prior to a selected one
8 of the comments being displayed in the hypertext document, and means to insert
9 reopening hypertext data in the hypertext document, following the selected one of the
10 comments being displayed in the hypertext document,

11 the closing hypertext data and the reopening hypertext data both being defined by the
12 open tag data for the comment display object associated with the selected one of the
13 comments being displayed.

1 27. The article of manufacture of claim 25 in which the linked list of objects comprises objects
2 appropriately selected from the set of objects comprising simple tag objects, tag objects, end tag
3 objects, text objects, comment insertion objects, comment display objects, complex whitespace
4 objects, spaces objects, newlines objects, hypertext comment objects, ordered list item tag objects,
5 and link tag objects.

1 28. The article of manufacture of claim 25 in which
2 each object in the linked list of objects has a type, and comprises contents derived from a
3 portion of the source file,

4 each comment insertion object and each comment display object comprises a contents line
5 identifier, representing the line number defining the location of the object in relation to the
6 source file,

7 the comment file consists of comment line identifier data associated with the set of

CA9-98-038

8 comments representing the line numbers in the source file with which each comment is
9 related,

10 and in which the computer readable program code means for causing the computer to
11 generate a hypertext document further comprises

12 means to traverse the linked list of objects, and means to select a display object from the
13 linked list of objects,

14 means to determine the type of the display object,

15 means to generate hypertext data to correspond to the type of the display object and the
16 contents of the display object,

17 means to identify the contents line identifier for the display object and for selecting
18 comment display data from the comment file based on the contents line identifier and the
19 comment line identifiers of the comment file, and

20 means to generate hypertext data from the comment file representing the selected one of
21 the set of comments corresponding to the comment line identifier associated with the
22 contents line identifier.

1 29. A computer program product for use with a hypertext server, the computer program product
2 comprising:

3 a computer usable medium having computer readable program code means embodied in the
4 medium for causing the storage and management of comments in a web-based file review
5 system, the comments being from a plurality of reviewers, and being associated with one or more

CA9-98-038

6 webs of source files, the computer program product having:

7 computer readable program code means for causing a computer to parse a selected one of the
8 set of source files into a linked list of objects corresponding to a hypertext representation of
9 the selected source file, the linked list further comprising comment insertion objects and
10 comment display objects, the parser writing the linked list of objects to a binary markup file
11 representing the linked list of objects and corresponding to the selected one of the set of
12 source files, each comment display object being capable of being associated with one or
13 more comments,

14 computer readable program code means for causing the computer to create and manage a set
15 of comment files, each comment file being associated with a one of the set of source files and
16 comprising data representing comments associated with the one of the set of source files,
17 computer readable program code means for causing the computer to accept new comments
18 for inclusion in the set of comments associated with a reviewer-defined source file and to
19 update the associated comment file,

20 computer readable program code means for causing the computer to generate a hypertext
21 document from a markup file corresponding to reviewer-selected source file and from the
22 associated comment file, the hypertext document corresponding to the reviewer-selected
23 source file and

24 the hypertext document including portions corresponding to one or more of the set of
25 comments associated with the reviewer-selected source file, the hypertext data for each
26 portion relating to a comment to be displayed being defined by the associated comment
27 display object,

CA9-98-038

28 the hypertext document selectively including hypertext links representing comment
29 insertion objects, the hypertext links providing reviewers with forms for reviewer entry
30 of comments,

31 the hypertext document selectively including hypertext data for calling the common
32 gateway interface program for generating a hypertext document and the hypertext
33 document selectively including hypertext data for calling the common gateway interface
34 program for accepting new comments,

35 computer readable program code means for causing the computer to communicate the
36 hypertext document to a browser for display.

1 30. The computer program product of claim 29, in which each one of the comment display objects
2 comprises

3 open tag data representing the open hypertext tags, at the location of the comment display
4 object, in the linked list of objects,
5 and in which the computer readable program code means for causing the computer to
6 generate a hypertext document further comprises

7 means to insert closing hypertext data in the hypertext document, prior to a selected one
8 of the comments being displayed in the hypertext document, and means to insert
9 reopening hypertext data in the hypertext document, following the selected one of the
10 comments being displayed in the hypertext document,

11 the closing hypertext data and the reopening hypertext data both being defined by the
12 open tag data for the comment display object associated with the selected one of the
13 comments being displayed.

CA9-98-038

1 31. The computer program product of claim 29 in which the linked list of objects comprises
2 objects appropriately selected from the set of objects comprising simple tag objects, tag objects, end
3 tag objects, text objects, comment insertion objects, comment display objects, complex whitespace
4 objects, spaces objects, newlines objects, hypertext comment objects, ordered list item tag objects,
5 and link tag objects.

1 32. The computer program product of claim 29 in which
2 each object in the linked list of objects has a type, and comprises contents derived from a
3 portion of the source file,

4 each comment insertion object and each comment display object comprises a contents line
5 identifier, representing the line number defining the location of the object in relation to the
6 source file,

7 the comment file consists of comment line identifier data associated with the set of
8 comments representing the line numbers in the source file with which each comment is
9 related,

10 and in which the computer readable program code means for causing the computer to
11 generate a hypertext document further comprises

12 means to traverse the linked list of objects, and means to select a display object from the
13 linked list of objects,

14 means to determine the type of the display object,

CA9-98-038

15 means to generate hypertext data to correspond to the type of the display object and the
16 contents of the display object,

17 means to identify the contents line identifier for the display object and for selecting
18 comment display data from the comment file based on the contents line identifier and the
19 comment line identifiers of the comment file, and

20 means to generate hypertext data from the comment file representing the selected one of
21 the set of comments corresponding to the comment line identifier associated with the
22 contents line identifier.

1 33. A method of storing and managing a set of comments associated with a source file, in a file
2 review system, the method comprising the steps of

3 accepting data from the source file and storing a representation of the source file as a markup
4 file,

5 creating a comment file containing data representing the set of comments associated with the
6 source file,

7 responding to user input to accept new comments for inclusion in the set of comments associated
8 with the source file and updating the comment file to correspond to the complete set of
9 comments,

10 responding to user input to dynamically generate a hypertext document from the markup file and
11 from the comment file, the hypertext document corresponding to the source file and including
12 portions corresponding to one or more of the set of comments associated with the source file,

CA9-98-038

13 communicating the hypertext document to a user for display.

1 34. The method of claim 33 in which the step of accepting data from the source file and storing
2 the source file as a markup file comprises the step of representing the source file as a linked list of
3 objects corresponding to a hypertext representation of the source file, wherein the markup file is a
4 binary file representation of the linked list.

1 35. The method of claim 34 in which the step of representing the source file as a linked list of
2 objects further comprises the step of inserting comment insertion objects into the linked list of
3 objects and in which step of generating a hypertext document further comprises the step of including
4 hypertext data in the hypertext document to display one or more symbols representing each comment
5 insertion object.

1 36. The method of claim 35 in which the step of including hypertext data in the hypertext
2 document to display one or more symbols representing each comment insertion object further
3 comprises the step of including a hypertext link to a hypertext form for user input.

1 37. The method of claims 35 or 36 in step of representing the source file as a linked list of objects
2 further comprises the step of inserting comment display objects into the linked list of objects, each
3 comment display object being associated with a comment insertion object and in which the step of
4 generating a hypertext document further comprises the step of including hypertext data in the
5 hypertext document to display comments in accordance with the corresponding comment display
6 objects.

1 38. The method of claim 37 in which each one of the comment display objects comprises

2 open tag data representing the open hypertext tags, at the location of the comment display

CA9-98-038

3 object, in the linked list of objects

4 and in which the step of generating a hypertext document further comprises the steps of

5 inserting closing hypertext data in the hypertext document, prior to a selected one of the
6 comments being displayed in the document, and inserting reopening hypertext data in the
7 hypertext document, following the selected one of the comments being displayed in the
8 hypertext document,

9 the closing hypertext data and the reopening hypertext data both being defined by the
10 open tag data for the comment display object associated with the selected one of the
11 comments being displayed.

1 39. A method for storing and managing comments in a web-based file review system, the
2 comments being from a plurality of reviewers and being associated with one or more webs of source
3 files, comprising the steps of

4 parsing a selected one of the set of source files into a linked list of objects corresponding to
5 a hypertext representation of the selected source file, the linked list further comprising
6 comment insertion objects and comment display objects, the parser writing the linked list of
7 objects to a binary markup file representing the linked list of objects and corresponding to
8 the selected one of the set of source files, each comment display object being capable of
9 being associated with one or more comments,

10 on review request, accepting new comments for inclusion in the set of comments associated
11 with a reviewer-defined source file and for updating an associated comment file, the
12 comment file being associated with a one of the set of source files and comprising data

CA9-98-038

13 representing comments associated with the one of the set of source file,

14 dynamically generating a hypertext document from a markup file corresponding to reviewer-

15 selected source file and from the associated comment file, the hypertext document

16 corresponding to the reviewer-selected source file and

17 the hypertext document including portions corresponding to one or more of the set of

18 comments associated with the reviewer-selected source file, the hypertext data for each

19 portion relating to a comment to be displayed being defined by the associated comment

20 display object,

21 the hypertext document selectively including hypertext links representing comment

22 insertion objects, the hypertext links providing reviewers with forms for reviewer entry

23 of comments,

24 the hypertext document selectively including hypertext data for calling the common

25 gateway interface program for generating a hypertext document and the hypertext

26 document selectively including hypertext data for calling the common gateway interface

27 program for accepting new comments,

28 communicating the hypertext document to a browser for display.

1 40. A program storage device readable by a machine, tangibly embodying a program of

2 instructions executable by the machine to perform method steps for storing and managing comments

3 in a web-based file review system, the method steps comprising the steps of claims 33, 34, 35, 36,

4 37, or 38.

CA9-98-038

- 1 41. A program storage device readable by a machine, tangibly embodying a program of
- 2 instructions executable by the machine to perform method steps for storing and managing comments
- 3 in a web-based file review system, the method steps comprising the steps of claim 39.

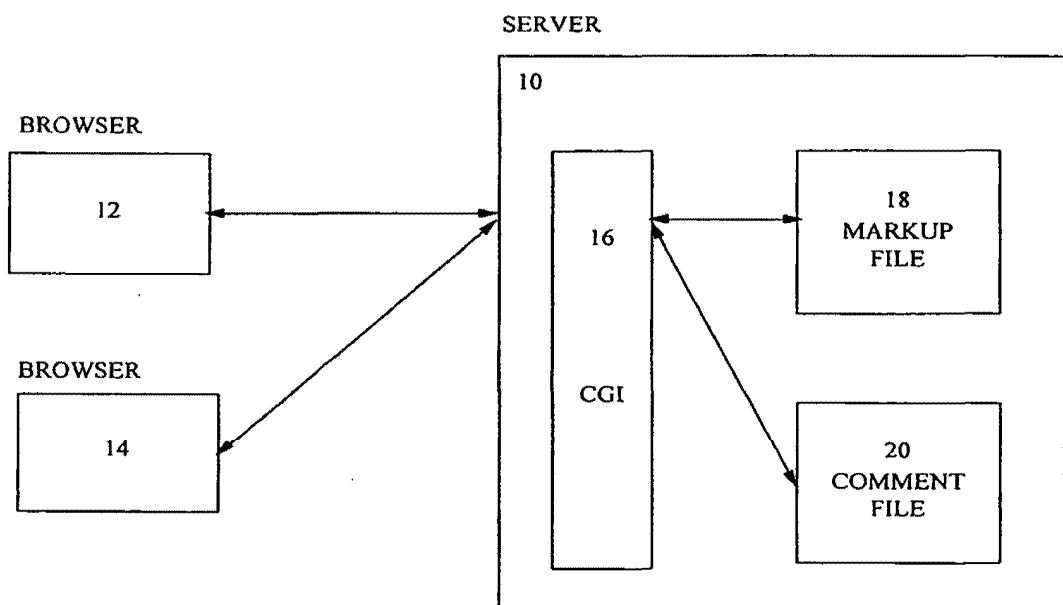


FIGURE 1

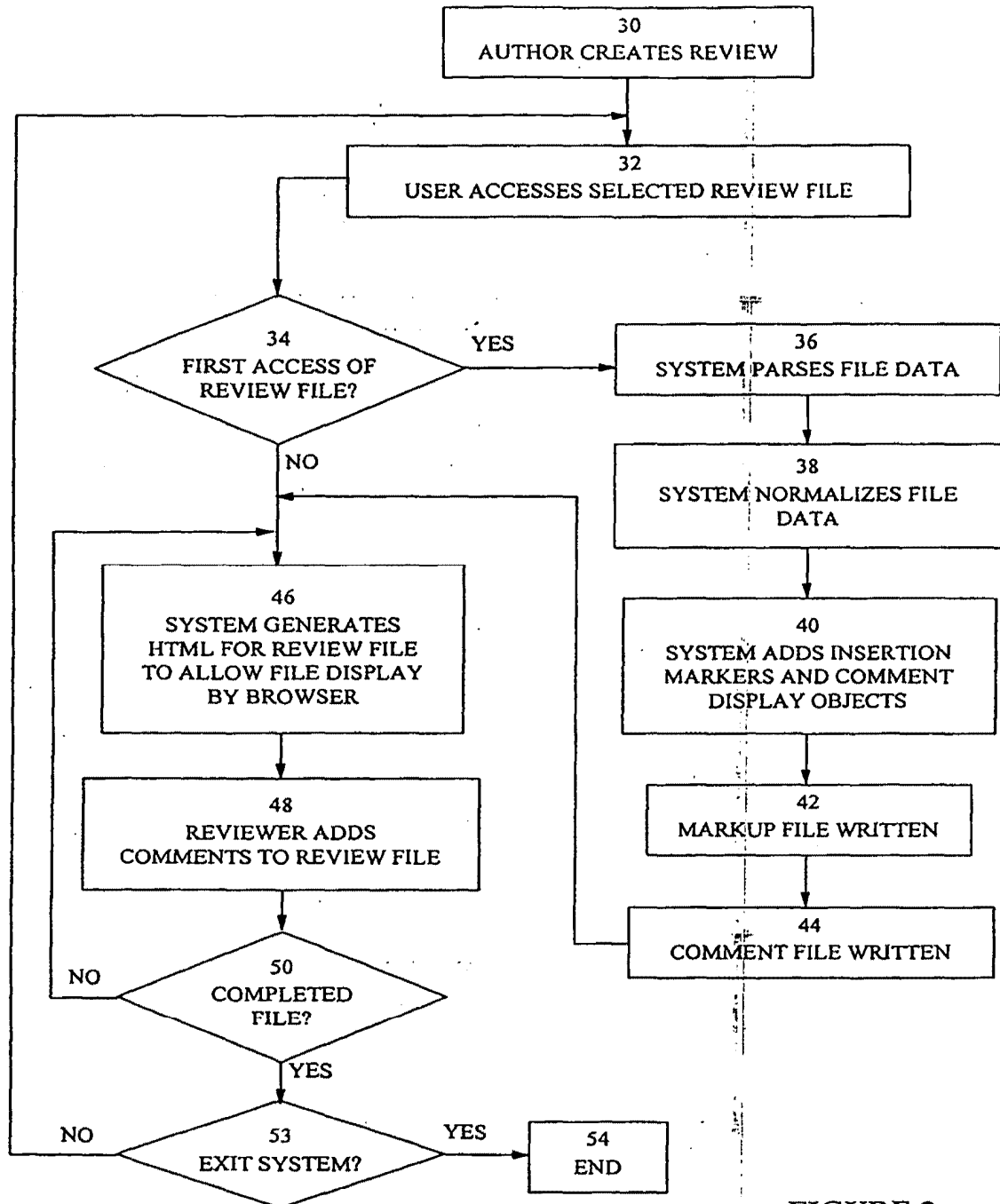


FIGURE 2