IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

ORACLE CORPORATION and ORACLE U.S.A. INC.,	)	
Plaintiffs/Counter-Defendants,	)	C.A. No. 06-cv-414 (SLR)
v.	)	
EPICREALM LICENSING, LP,	)	
Defendant/Counter-Plaintiff.	)	

DECLARATION OF BENJAMIN J. BRADFORD IN SUPPORT OF EPICREALM'S RESPONSE IN OPPOSITION TO ORACLE'S MOTION FOR LEAVE TO AMEND ITS DECLARATORY JUDGMENT COMPLAINT

I, Benjamin J. Bradford, declare as follows:

1. I am a member of good standing of the State Bar of Illinois and an attorney at the law firm of Jenner & Block, LLP, counsel for Defendant and Counter-Plaintiff, epicRealm Licensing, LP ("epicRealm"). I have personal knowledge of the matters stated and could competently testify to them if I were called as a witness.

2. Attached hereto as Exhibit 1 is a true and correct copy of U.S. Patent No. 5,894,554 entitled "System for Managing Dynamic Web Page Generation Requests by Intercepting Request at Web Server and Routing to Page Server Thereby Releasing Web Server to Process Other Requests."

3. Attached hereto as Exhibit 2 is a true and correct copy of U.S. Patent No. 6,415,335 entitled "System and Method for Managing Dynamic Web Page Generation Requests."

4. Attached hereto as Exhibit 3 is a true and correct copy of U.S. Patent No. 5,761,673 entitled "Method and Apparatus for Generating Dynamic Web Pages by Invoking a Predefined Procedural Package Stored in a Database."

5. Attached hereto as Exhibit 4 is a true and correct copy of an Office Action and Notice of References cited for U.S. Patent Application 08/594,686 which issued as the '673 patent.

6. Attached hereto as Exhibit 5 is true and correct copy of a Notice of Allowability and Notice of References cited for U.S. Patent Application 08/594,686 which issued as the '673 patent.

7. Attached hereto as Exhibit 6 is a true and correct copy of a USPTO search results print-out of the Oracle patents that the law firm of Blakely, Sokoloff, Taylor & Zafman LLP prosecuted.

8. Attached hereto as Exhibit 7 is a true and correct copy of an e-mail dated September 13, 2007, from George S. Bosy, counsel for epicRealm, to Robert J. Artuz and Theodore T. Herhold, counsel for Oracle.

9. Attached hereto as Exhibit 8 is a true and correct copy of a letter dated October 27, 2007, from Paul D. Margolis, counsel for epicRealm, to Theodore T. Herhold.

10. Attached hereto as Exhibit 9 is a true and correct copy of an e-mail dated November 1, 2007, from Theodore T. Herhold to George S. Bosy, counsel for epicRealm.

11. Attached hereto as Exhibit 10 is a true and correct copy of an e-mail dated November 2, 2007, from George S. Bosy to Theodore T. Herhold.

12. Attached hereto as Exhibit 11 is a true and correct copy of an e-mail and attachment dated November 8, 2007, from Theodore T. Herhold to Patrick L. Patras and George S. Bosy.

13. Attached hereto as Exhibit 12 is a true and correct copy of an e-mail dated November 30, 2007, from George S. Bosy to Theodore T. Herhold.

14. Attached hereto as Exhibit 13 is a true and correct copy of an e-mail dated December 13, 2007, from George S. Bosy to Theodore T. Herhold.

15. Attached hereto as Exhibit 14 is a true and correct copy of a letter dated December 14, 2007, from Gloria Torres, paralegal at Townsend and Townsend and Crew LLP, to Benjamin J. Bradford.

16. Attached hereto as Exhibit 15 is a true and correct copy of a letter dated December 19, 2007, from C. Daniel Trevino, paralegal at Townsend and Townsend and Crew LLP, to Benjamin J. Bradford.

17. Attached hereto as Exhibit 16 is a true and correct copy of a letter dated December 20, 2007, from C. Daniel Trevino, paralegal at Townsend and Townsend and Crew LLP, to Benjamin J. Bradford.

18. Attached hereto as Exhibit 17 is a true and correct copy of a letter dated December 21, 2007, from C. Daniel Trevino, paralegal at Townsend and Townsend and Crew LLP, to Benjamin J. Bradford.

19. Attached hereto as Exhibit 18 is a true and correct copy of an e-mail dated January 2, 2008, from Robert J. Artuz, counsel for Oracle, to Patrick L. Patras.

20. Attached hereto as Exhibit 19 is a true and correct copy of an e-mail dated January 3, 2008, from Patrick L. Patras to Robert J. Artuz.

21. Attached hereto as Exhibit 20 is a true and correct copy of a letter dated January 4, 2008, from Gloria Torres, paralegal at Townsend and Townsend and Crew LLP, to Benjamin J. Bradford.

22. Attached hereto as Exhibit 21 is a true and correct copy of a letter dated January 10, 2008, from C. Daniel Trevino, paralegal at Townsend and Townsend and Crew LLP, to Benjamin J. Bradford.

23. Attached hereto as Exhibit 22 is a true and correct copy of a letter dated January 11, 2008, from Robert J. Artuz to George S. Bosy.

24. Attached hereto as Exhibit 23 is a true and correct copy of Plaintiff's Supplemental Response to Defendant's Interrogatory No. 3 dated January 14, 2008.

25. Attached hereto as Exhibit 24 is a true and correct copy of a letter dated January 14, 2008, from David R. Bennett, counsel for epicRealm to Theodore T. Herhold.

26. Attached hereto as Exhibit 25 is a true and correct copy of a letter dated January 14, 2008, from Patrick L. Patras to Theodore T. Herhold.

27. Attached hereto as Exhibit 26 is a true and correct copy of an e-mail dated January 14, 2008, from Theodore T. Herhold to Patrick L. Patras.

28. Attached hereto as Exhibit 27 is a true and correct copy of an e-mail dated January 17, 2008 from Robert J. Artuz to Patrick L. Patras.

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct and that this Declaration was executed on this 18th day of January, 2008 at Chicago, Illinois.

By: _____

Benjamin J. Bradford
Jenner & Block LLP
330 North Wabash Avenue
Chicago, Illinois 60611
Tel: (312) 840-7224
bbradford@jenner.com

Attorney for Defendant
epicRealm Licensing, LP

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

CERTIFICATE OF SERVICE

I, David E. Moore, hereby certify that on January 18, 2008, the attached document was electronically filed with the Clerk of the Court using CM/ECF which will send notification to the registered attorney(s) of record that the document has been filed and is available for viewing and downloading.

I further certify that on January 18, 2008, I have Electronically Mailed the document to the following person(s):

Mary B. Graham
Morris, Nichols, Arsht & Tunnell LLP
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
MGraham@MNAT.com

James G. Gilliland
Theodore T. Herhold
Chad E. King
Robert J. Artuz
Eric M. Hutchins
Townsend and Townsend and Crew LLP
379 Lytton Avenue
Palo Alto, CA 94301
OracleEpicrealm@townsend.com

/s/ David E. Moore
Richard L. Horwitz
David E. Moore
Potter Anderson & Corroon LLP
Hercules Plaza – Sixth Floor
1313 North Market Street
P.O. Box 951
Wilmington, DE 19899-0951
(302) 984-6000
rhorwitz@potteranderson.com
dmoore@potteranderson.com

Exhibit 1

US005894554A

United States Patent [19]

Lowery et al.

[11] Patent Number: 5,894,554

[45] Date of Patent: Apr. 13, 1999

[54] **SYSTEM FOR MANAGING DYNAMIC WEB PAGE GENERATION REQUESTS BY INTERCEPTING REQUEST AT WEB SERVER AND ROUTING TO PAGE SERVER THEREBY RELEASING WEB SERVER TO PROCESS OTHER REQUESTS**

[75] Inventors: **Keith Lowery**, Richardson; **Andrew B. Levine**, Plano; **Ronald L. Howell**, Rowlett, all of Tex.

[73] Assignee: **InfoSpinner, Inc.**, Richardson, Tex.

[21] Appl. No.: **08/636,477**

[22] Filed: **Apr. 23, 1996**

[51] **Int. Cl.**[6] **G06F 13/14**; G06F 13/20

[52] **U.S. Cl.** **395/200.33**; 395/200.68; 395/200.75; 395/680; 707/10; 707/104

[58] **Field of Search** 358/400; 395/800, 395/700, 200.68, 200.75, 200.53, 680, 200.33; 707/104, 10

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,866,706	9/1989	Christophersen et al.	370/85.7
5,341,499	8/1994	Doragh	395/700
5,392,400	2/1995	Berkowitz et al.	395/200
5,404,522	4/1995	Carmon et al.	395/650
5,404,523	4/1995	DellaFera et al.	395/650
5,404,527	4/1995	Irwin et al.	395/700
5,452,460	9/1995	Distelberg et al.	395/700
5,532,838	7/1996	Barbari	358/400
5,751,956	5/1998	Kirsch	395/200.33
5,761,673	6/1998	Bookman et al.	707/104

OTHER PUBLICATIONS

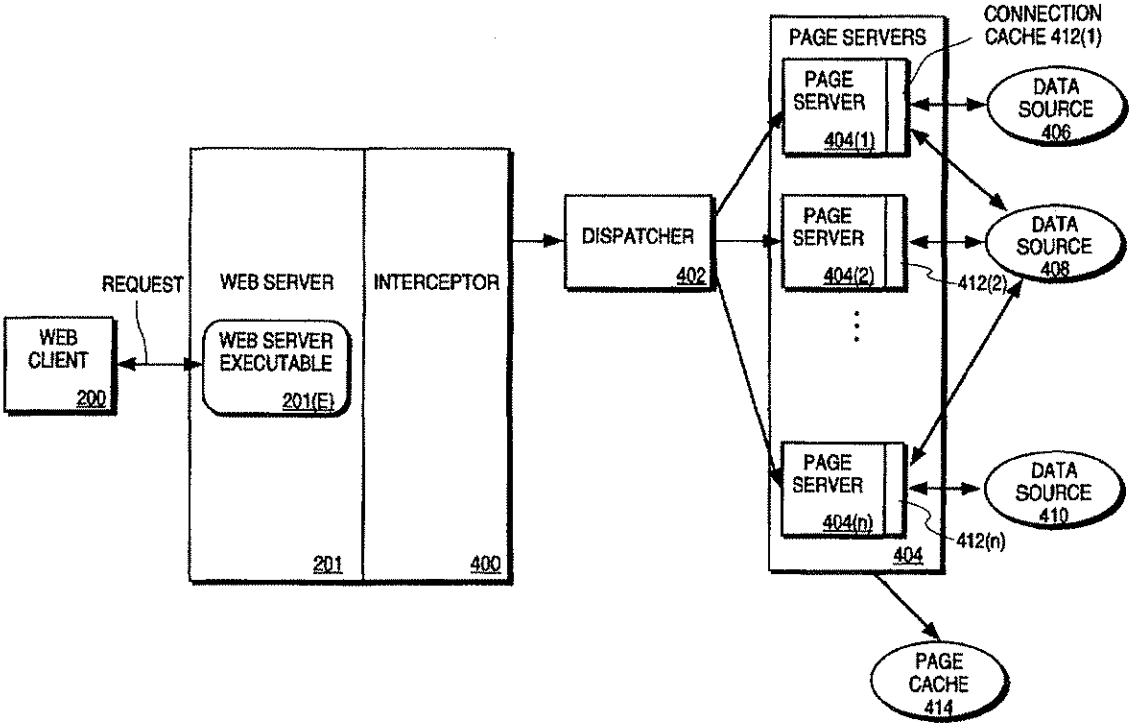
"Beyond the Web: Excavating the Real World Via Mosaic", Goldberg et al. Second International WWW, Oct. 17, 1994.
PCT International Search Report, Aug. 21, 1997.

Primary Examiner—Thomas C. Lee
Assistant Examiner—Rehana Perveen
Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman LLP

[57] **ABSTRACT**

The present invention teaches a method and apparatus for creating and managing custom Web sites. Specifically, one embodiment of the present invention claims a computer-implemented method for managing a dynamic Web page generation request to a Web server, the computer-implemented method comprising the steps of routing the request from the Web server to a page server, the page server receiving the request and releasing the Web server to process other requests, processing the request, the processing being performed by the page server concurrently with the Web server, as the Web server processes the other requests, and dynamically generating a Web page in response to the request, the Web page including data dynamically retrieved from one or more data sources.

11 Claims, 5 Drawing Sheets



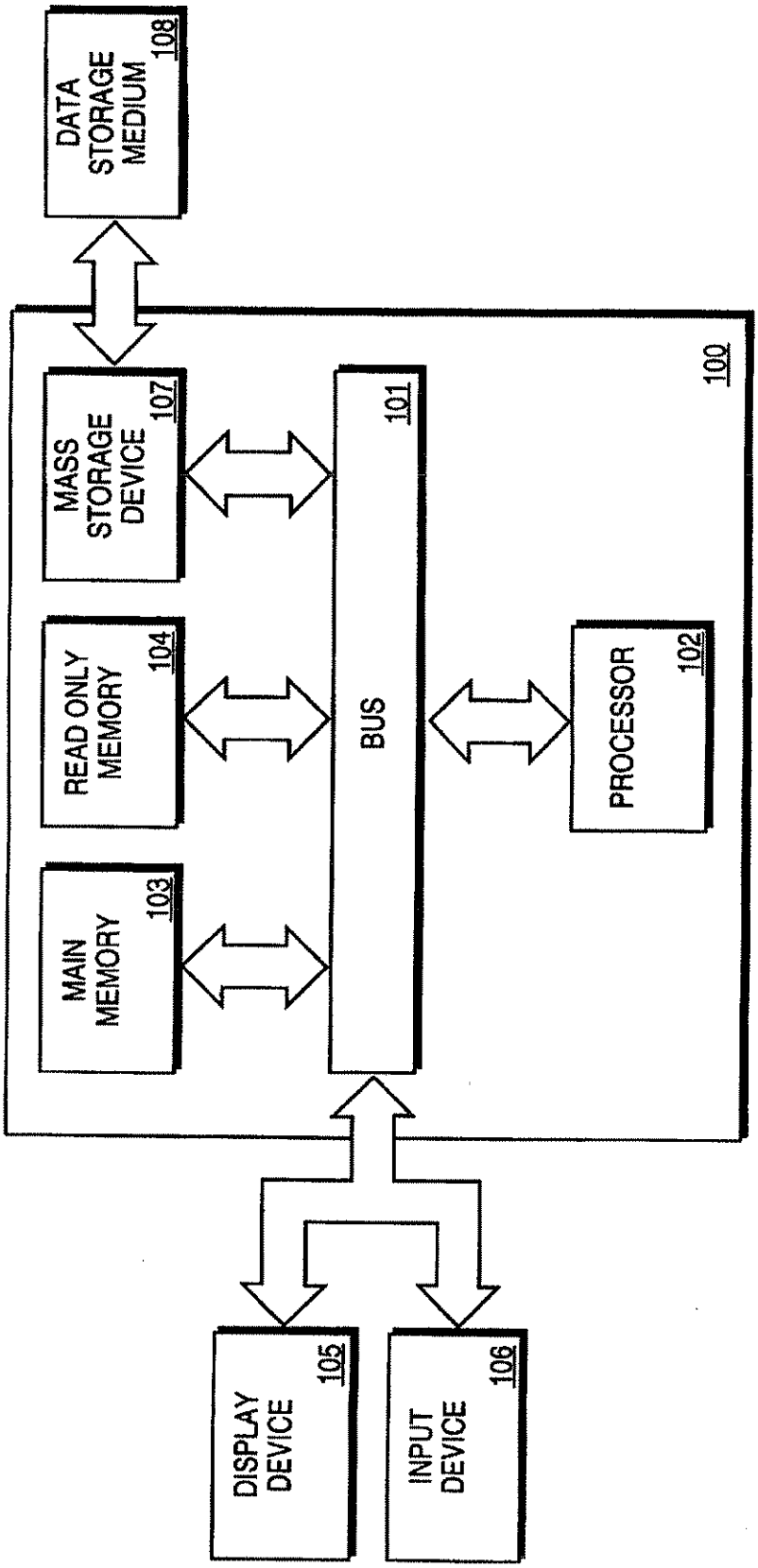

DATA STORAGE MEDIUM
108
100
MASS STORAGE DEVICE
107
101
READ ONLY MEMORY
104
BUS
PROCESSOR
102
MAIN MEMORY
103
DISPLAY DEVICE
105
INPUT DEVICE
106

FIG. 1

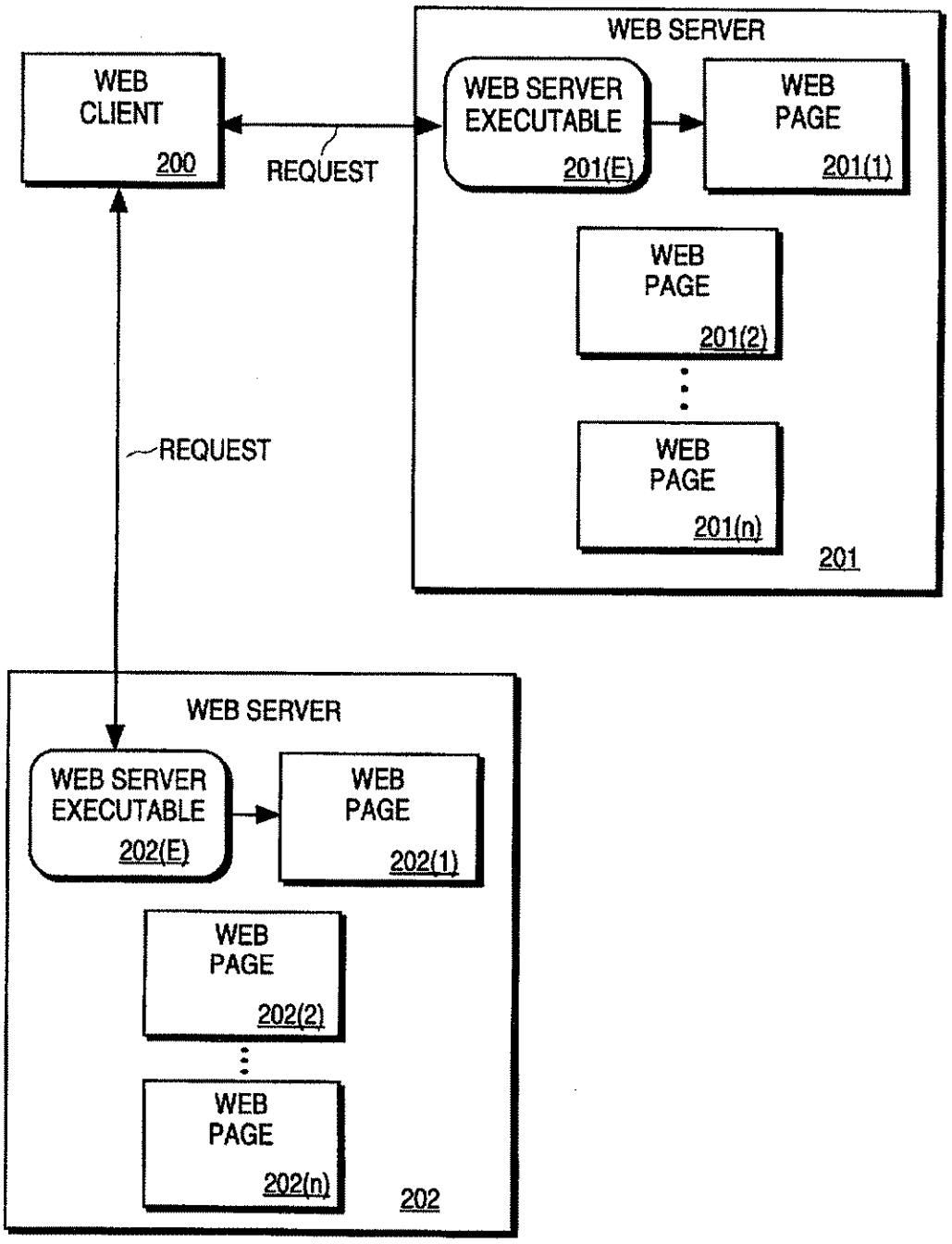


FIG. 2 (PRIOR ART)

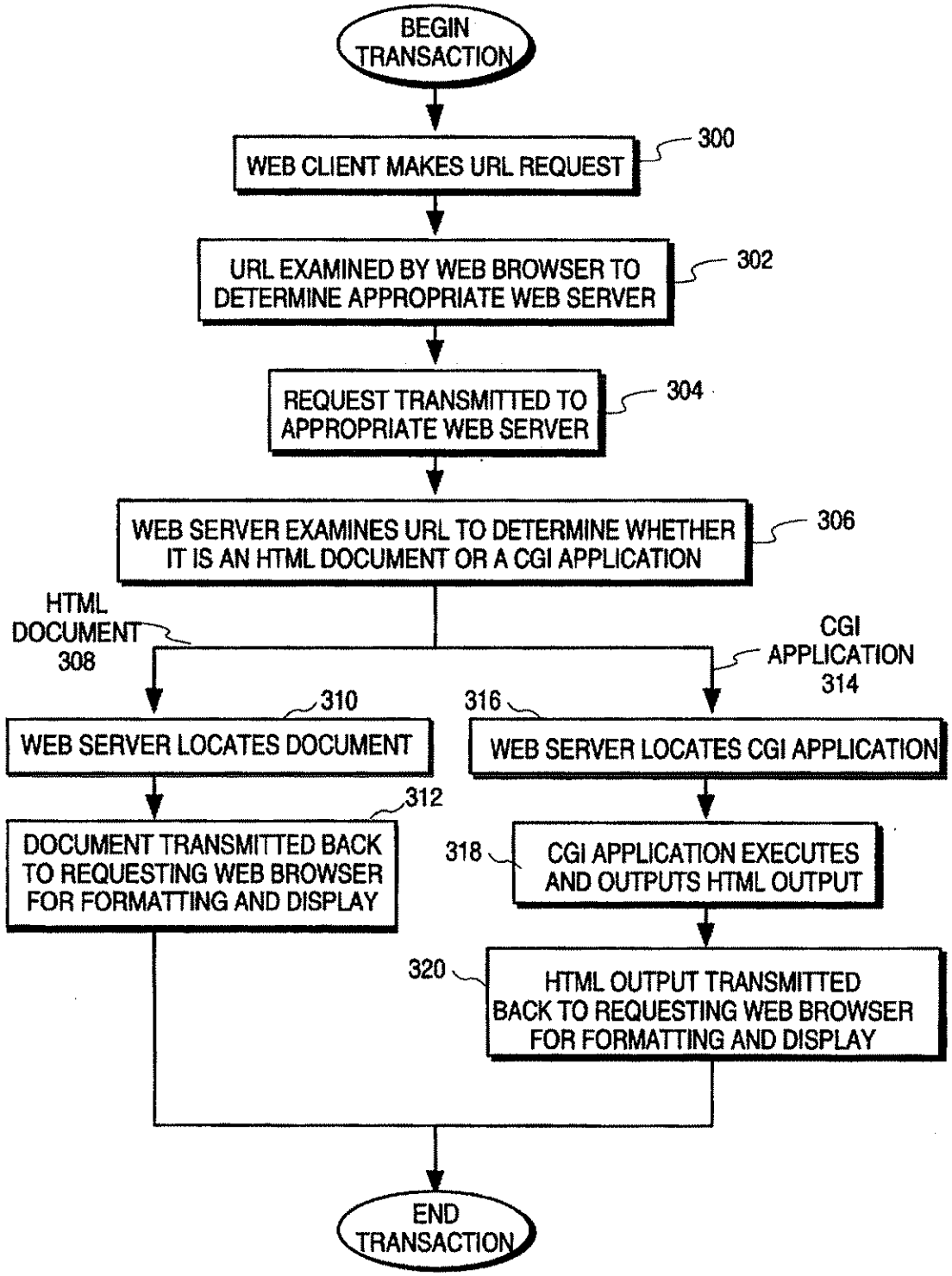


FIG. 3 (PRIOR ART)

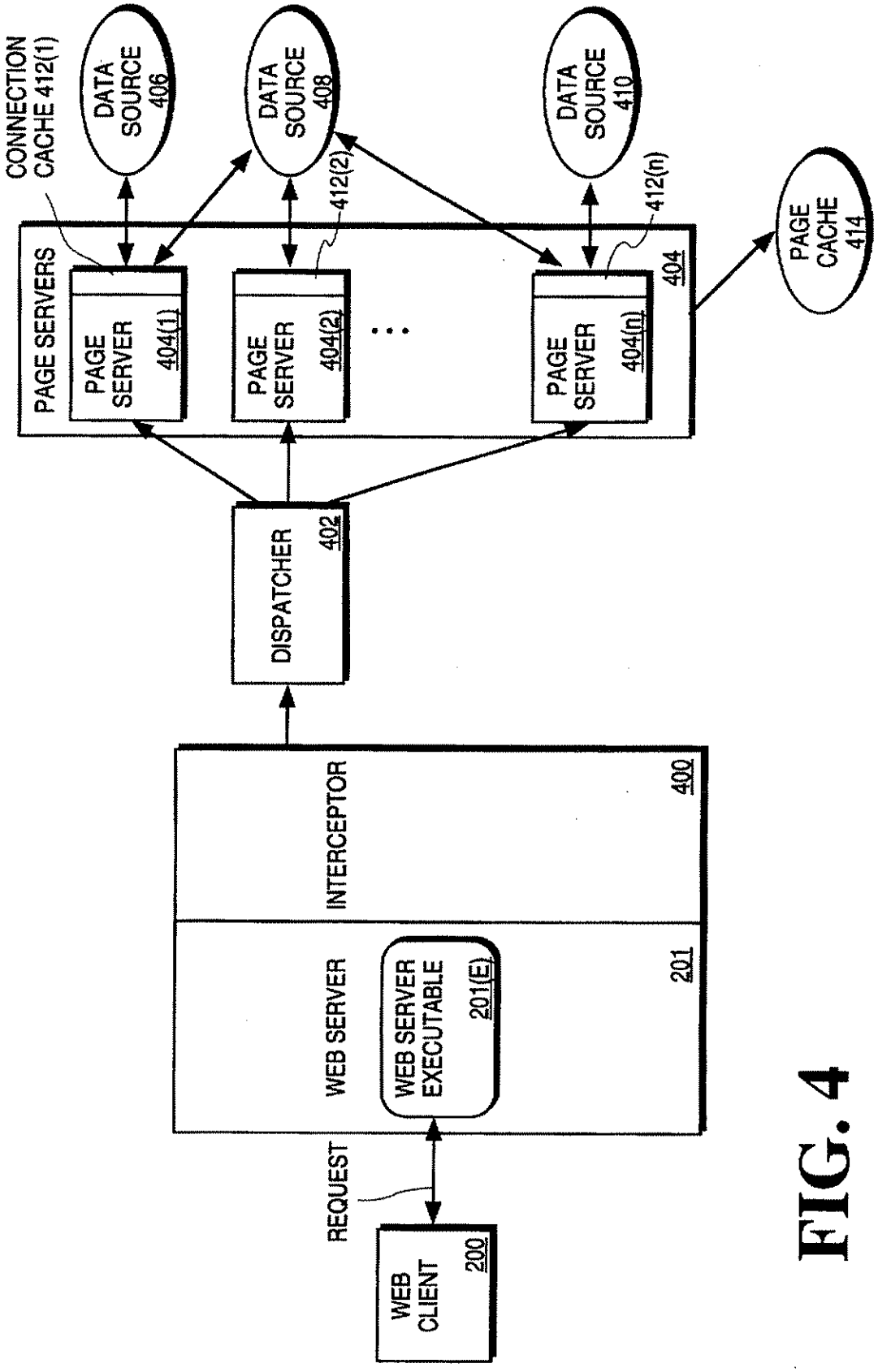

CONNECTION CACHE 412(1)
DATA SOURCE 406
DATA SOURCE 408
DATA SOURCE 410
412(2)
412(n)
PAGE CACHE 414
PAGE SERVERS
PAGE SERVER 404(1)
PAGE SERVER 404(2)
PAGE SERVER 404(n)
404
DISPATCHER 402
INTERCEPTOR 400
WEB SERVER
WEB SERVER EXECUTABLE 201(E)
201
REQUEST
WEB CLIENT 200

FIG. 4

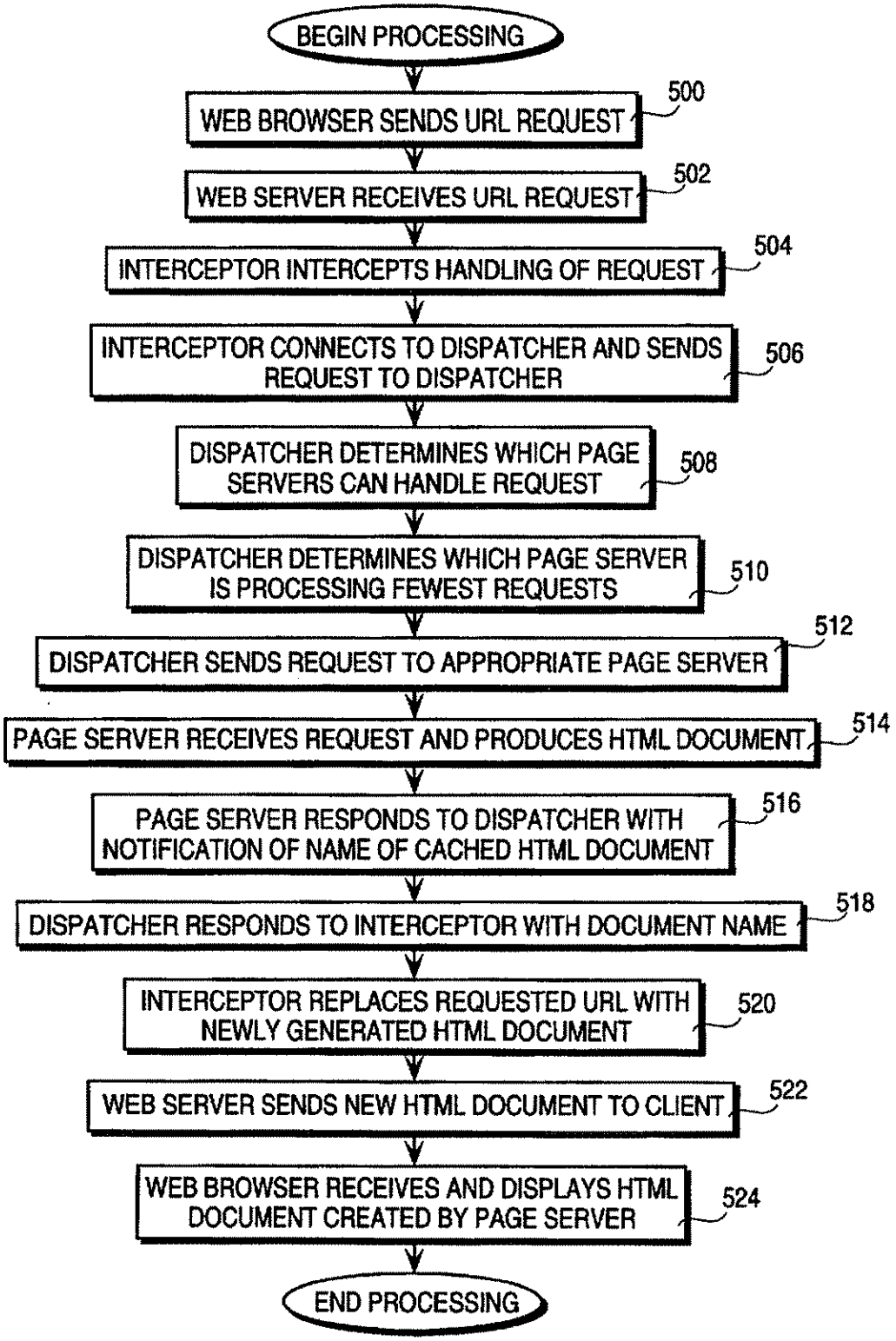

BEGIN PROCESSING
WEB BROWSER SENDS URL REQUEST
500
WEB SERVER RECEIVES URL REQUEST
502
INTERCEPTOR INTERCEPTS HANDLING OF REQUEST
504
INTERCEPTOR CONNECTS TO DISPATCHER AND SENDS REQUEST TO DISPATCHER
506
DISPATCHER DETERMINES WHICH PAGE SERVERS CAN HANDLE REQUEST
508
DISPATCHER DETERMINES WHICH PAGE SERVER IS PROCESSING FEWEST REQUESTS
510
DISPATCHER SENDS REQUEST TO APPROPRIATE PAGE SERVER
512
PAGE SERVER RECEIVES REQUEST AND PRODUCES HTML DOCUMENT
514
PAGE SERVER RESPONDS TO DISPATCHER WITH NOTIFICATION OF NAME OF CACHED HTML DOCUMENT
516
DISPATCHER RESPONDS TO INTERCEPTOR WITH DOCUMENT NAME
518
INTERCEPTOR REPLACES REQUESTED URL WITH NEWLY GENERATED HTML DOCUMENT
520
WEB SERVER SENDS NEW HTML DOCUMENT TO CLIENT
522
WEB BROWSER RECEIVES AND DISPLAYS HTML DOCUMENT CREATED BY PAGE SERVER
524
END PROCESSING

FIG. 5

SYSTEM FOR MANAGING DYNAMIC WEB PAGE GENERATION REQUESTS BY INTERCEPTING REQUEST AT WEB SERVER AND ROUTING TO PAGE SERVER THEREBY RELEASING WEB SERVER TO PROCESS OTHER REQUESTS

FIELD OF THE INVENTION

The present invention relates to the field of Internet technology. Specifically, the present invention relates to the creation and management of custom World Wide Web sites.

DESCRIPTION OF RELATED ART

The World Wide Web (the Web) represents all of the computers on the Internet that offer users access to information on the Internet via interactive documents or Web pages. These Web pages contain hypertext links that are used to connect any combination of graphics, audio, video and text, in a non-linear, non-sequential manner. Hypertext links are created using a special software language known as HyperText Mark-Up Language (HTML).

Once created, Web pages reside on the Web, on Web servers or Web sites. A Web site can contain numerous Web pages. Web client machines running Web browsers can access these Web pages at Web sites via a communications protocol known as HyperText Transport Protocol (HTTP). Web browsers are software interfaces that run on World Wide Web clients to allow access to Web sites via a simple user interface. A Web browser allows a Web client to request a particular Web page from a Web site by specifying a Uniform Resource Locator (URL). A URL is a Web address that identifies the Web page and its location on the Web. When the appropriate Web site receives the URL, the Web page corresponding to the requested URL is located, and if required, HTML output is generated. The HTML output is then sent via HTTP to the client for formatting on the client's screen.

Although Web pages and Web sites are extremely simple to create, the proliferation of Web sites on the Internet highlighted a number of problems. The scope and ability of a Web page designer to change the content of the Web page was limited by the static nature of Web pages. Once created, a Web page remained static until it was manually modified. This in turn limited the ability of Web site managers to effectively manage their Web sites.

The Common Gateway Interface (CGI) standard was developed to resolve the problem of allowing dynamic content to be included in Web pages. CGI "calls" or procedures enable applications to generate dynamically created HTML output, thus creating Web pages with dynamic content. Once created, these CGI applications do not have to be modified in order to retrieve "new" or dynamic data. Instead, when the Web page is invoked, CGI "calls" or procedures are used to dynamically retrieve the necessary data and to generate a Web page.

CGI applications also enhanced the ability of Web site administrators to manage Web sites. Administrators no longer have to constantly update static Web pages. A number of vendors have developed tools for CGI based development, to address the issue of dynamic Web page generation. Companies like Spider™ and Bluestone™, for example, have each created development tools for CGI-based Web page development. Another company, Haht Software™, has developed a Web page generation tool that uses a BASIC-like scripting language, instead of a CGI scripting language.

Tools that generate CGI applications do not, however, resolve the problem of managing numerous Web pages and requests at a Web site. For example, a single company may maintain hundreds of Web pages at their Web site. Current Web server architecture also does not allow the Web server to efficiently manage the Web page and process Web client requests. Managing these hundreds of Web pages in a coherent manner and processing all requests for access to the Web pages is thus a difficult task. Existing development tools are limited in their capabilities to facilitate dynamic Web page generation, and do not address the issue of managing Web requests or Web sites.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a method and apparatus for creating and managing custom Web sites. Specifically, the present invention claims a method and apparatus for managing dynamic web page generation requests.

In one embodiment, the present invention claims a computer-implemented method for managing a dynamic Web page generation request to a Web server, the computer-implemented method comprising the steps of routing the request from the Web server to a page server, the page server receiving the request and releasing the Web server to process other requests, processing the request, the processing being performed by the page server concurrently with the Web server, as the Web server processes the other requests, and dynamically generating a Web page in response to the request, the Web page including data dynamically retrieved from one or more data sources. Other embodiments also include connection caches to the one or more data sources, page caches for each page server, and custom HTML extension templates for configuring the Web page.

Other objects, features and advantages of the present invention will be apparent from the accompanying drawings and from the detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates a typical computer system in which the present invention operates.

FIG. **2** illustrates a typical prior art Web server environment.

FIG. **3** illustrates a typical prior art Web server environment in the form of a flow diagram.

FIG. **4** illustrates one embodiment of the presently claimed invention.

FIG. **5** illustrates the processing of a Web browser request in the farm of a flow diagram, according to one embodiment of the presently claimed invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention relates to a method and apparatus for creating and managing custom Web sites. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent to one of ordinary skill in the art, however, that these specific details need not be used to practice the present invention. In other instances, well-known structures, interfaces and processes have not been shown in detail in order not to unnecessarily obscure the present invention.

FIG. **1** illustrates a typical computer system **100** in which the present invention operates. The preferred embodiment of

the present invention is implemented on an IBM™ Personal Computer manufactured by IBM Corporation of Armonk, N.Y. An alternate embodiment may be implemented on an RS/6000™ Workstation manufactured by IBM Corporation of Armonk, N.Y. It will be apparent to those of ordinary skill in the art that other computer system architectures may also be employed.

In general, such computer systems as illustrated by FIG. 1 comprise a bus **101** for communicating information, a processor **102** coupled with the bus **101** for processing information, main memory **103** coupled with the bus **101** for storing information and instructions for the processor **102**, a read-only memory **104** coupled with the bus **101** for storing static information and instructions for the processor **102**, a display device **105** coupled with the bus **101** for displaying information for a computer user, an input device **106** coupled with the bus **101** for communicating information and command selections to the processor **102**, and a mass storage device **107**, such as a magnetic disk and associated disk drive, coupled with the bus **101** for storing information and instructions. A data storage medium **108** containing digital information is configured to operate with mass storage device **107** to allow processor **102** access to the digital information on data storage medium **108** via bus **101**.

Processor **102** may be any of a wide variety of general purpose processors or microprocessors such as the Pentium™ microprocessor manufactured by Intel™ Corporation or the RS/6000™ processor manufactured by IBM Corporation. It will be apparent to those of ordinary skill in the art, however, that other varieties of processors may also be used in a particular computer system. Display device **105** may be a liquid crystal device, cathode ray tube (CRT), or other suitable display device. Mass storage device **107** may be a conventional hard disk drive, floppy disk drive, CD-ROM drive, or other magnetic or optical data storage device for reading and writing information stored on a hard disk, a floppy disk, a CD-ROM a magnetic tape, or other magnetic or optical data storage medium. Data storage medium **108** may be a hard disk, a floppy disk, a CD-ROM, a magnetic tape, or other magnetic or optical data storage medium.

In general, processor **102** retrieves processing instructions and data from a data storage medium **108** using mass storage device **107** and downloads this information into random access memory **103** for execution. Processor **102**, then executes an instruction stream from random access memory **103** or read-only memory **104**. Command selections and information input at input device **106** are used to direct the flow of instructions executed by processor **102**. Equivalent input device **106** may also be a pointing device such as a conventional mouse or trackball device. The results of this processing execution are then displayed on display device **105**.

The preferred embodiment of the present invention is implemented as a software module, which may be executed on a computer system such as computer system **100** in a conventional manner. Using well known techniques, the application software of the preferred embodiment is stored on data storage medium **108** and subsequently loaded into and executed within computer system **100**. Once initiated, the software of the preferred embodiment operates in the manner described below.

FIG. **2** illustrates a typical prior art Web server environment. Web client **200** can make URL requests to Web server **201** or Web server **202**. Web servers **201** and **202** include Web server executables, **201**(E) and **202**(E) respectively, that perform the processing of Web client requests. Each Web server may have a number of Web pages **201**(1)–(n) and **202**(1)–(n). Depending on the URL specified by the Web client **200**, the request may be routed by either Web server executable **201**(E) to Web page **201** (1), for example, or from Web server executable **202**(E) to Web page **202** (1). Web client **200** can continue making URL requests to retrieve other Web pages. Web client **200** can also use hyperlinks within each Web page to "jump" to other Web pages or to other locations within the same Web page.

FIG. **3** illustrates this prior art Web server environment in the form of a flow diagram. In processing block **300**, the Web client makes a URL request. This URL request is examined by the Web browser to determine the appropriate Web server to route the request to in processing block **302**. In processing block **304** the request is then transmitted from the Web browser to the appropriate Web server, and in processing block **306** the Web server executable examines the URL to determine whether it is a HTML document or a CGI application. If the request is for an HTML document **308**, then the Web server executable locates the document in processing block **310**. The document is then transmitted back through the requesting Web browser for formatting and display in processing block **312**.

If the URL request is for a CGI application **314**, however, the Web server executable locates the CGI application in processing block **316**. The CGI application then executes and outputs HTML output in processing block **318** and finally, the HTML output is transmitted back to requesting Web browser for formatting and display in processing block **320**.

This prior art Web server environment does not, however, provide any mechanism for managing the Web requests or the Web sites. As Web sites grow, and as the number of Web clients and requests increase, Web site management becomes a crucial need.

For example, a large Web site may receive thousands of requests or "hits" in a single day. Current Web servers process each of these requests on a single machine, namely the Web server machine. Although these machines may be running "multi-threaded" operating systems that allow transactions to be processed by independent "threads," all the threads are nevertheless on a single machine, sharing a processor. As such, the Web executable thread may hand off a request to a processing thread, but both threads will still have to be handled by the processor on the Web server machine. When numerous requests are being simultaneously processed by multiple threads on a single machine, the Web server can slow down significantly and become highly inefficient. The claimed invention addresses this need by utilizing a partitioned architecture to facilitate the creation and management of custom Web sites and servers.

FIG. **4** illustrates one embodiment of the presently claimed invention. Web client **200** issues a URL request that is processed to determined proper routing. In this embodiment, the request is routed to Web server **201**. Instead of Web server executable **201**(E) processing the URL request, however, Interceptor **400** intercepts the request and routes it to Dispatcher **402**. In one embodiment, Interceptor **400** resides on the Web server machine as an extension to Web server **201**. This embodiment is appropriate for Web servers such as Netsite™ from Netscape, that support such extensions. A number of public domain Web servers, such as NCSA™ from the National Center for Supercomputing Applications at the University of Illinois, Urbana-Champaign, however, do not provide support for

this type of extension. Thus, in an alternate embodiment, Interceptor **400** is an independent module, connected via an "intermediate program" to Web server **201**. This intermediate program can be a simple CGI application program that connects Interceptor **400** to Web server **201**. Alternate intermediate programs the perform the same functionality can also be implemented.

In one embodiment of the invention, Dispatcher **402** resides on a different machine than Web server **201**. This embodiment overcomes the limitation described above, in prior art Web servers, wherein all processing is performed by the processor on a single machine. By routing the request to Dispatcher **402** residing on a different machine than the Web server executable **201**(E), the request can then be processed by a different processor than the Web server executable **201**(E). Web server executable **201**(E) is thus free to continue servicing client requests on Web server **201** while the request is processed "off-line," at the machine on which Dispatcher **402** resides.

Dispatcher **402** can, however, also reside on the same machine as the Web server. The Web site administrator has the option of configuring Dispatcher **402** on the same machine as Web server **201**, taking into account a variety of factors pertinent to a particular Web site, such as the size of the Web site, the number of Web pages and the number of hits at the Web site. Although this embodiment will not enjoy the advantage described above, namely off-loading the processing of Web requests from the Web server machine, the embodiment does allow flexibility for a small Web site to grow. For example, a small Web site administrator can use a single machine for both Dispatcher **402** and Web server **201** initially, then off-load Dispatcher **402** onto a separate machine as the Web site grows. The Web site can thus take advantage of other features of the present invention regardless of whether the site has separate machines configured as Web servers and dispatchers.

Dispatcher **402** receives the intercepted request and then dispatches the request to one of a number of Page servers **404** (1)–(n). For example, if Page server **404** (1) receives the dispatched request, it processes the request and retrieves the data from an appropriate data source, such as data source **406**, data source **408**, or data source **410**. Data sources, as used in the present application, include databases, spreadsheets, files and any other type of data repository. Page server **404** (1) can retrieve data from more than one data source and incorporate the data from these multiple data sources in a single Web page.

In one embodiment, each Page server **404**(1)–(n) resides on a separate machine on the network to distribute the processing of the request. Dispatcher **402** maintains a variety of information regarding each Page server on the network, and dispatches requests based on this information. For example, Dispatcher **402** retains dynamic information regarding the data sources that any given Page server can access. Dispatcher **402** thus examines a particular request and determines which Page servers can service the URL request. Dispatcher **402** then hands off the request to the appropriate Page server.

For example, if the URL request requires financial data from data source **408**, dispatcher **402** will first examine an information list. Dispatcher **402** may determine that Page server **404**(3), for example, has access to the requisite data in data source **408**. Dispatcher **402** will thus route the URL request to Page server **404**(3). This "connection caching" functionality is described in more detail below, under the heading "Performance."

Alternately, Dispatcher **402** also has the ability to determine whether a particular Page server already has the necessary data cached in the Page server's page cache (described in more detail below, under the heading "Performance"). Dispatcher **402** may thus determine that Page server **404**(1) and **404**(2) are both logged into Data source **408**, but that Page server **404**(2) has the financial information already cached in Page server **404**(2)'s page cache. In this case, Dispatcher **402** will route the URL request to Page server **404**(2) to more efficiently process the request.

Finally, Dispatcher **402** may determine that a number or all Page servers **404**(1)–(n) are logged into Data source **408**. In this scenario, Dispatcher **402** can examine the number of requests that each Page server is servicing and route the request to the least busy page server. This "load balancing" capability can significantly increase performance at a busy Web site and is discussed in more detail below, under the heading "Scalability".

If, for example, Page server **404**(2), receives the request, Page server **404**(2) will process the request. While Page server **404**(2) is processing the request, Web server executable **201**(E) can concurrently process other Web client requests. This partitioned architecture thus allows both Page server **404**(2) and Web server executable **201**(E) to simultaneously process different requests, thus increasing the efficiency of the Web site. Page server **404**(2) dynamically generates a Web page in response to the Web client request, and the dynamic Web page is then either transmitted back to requesting Web client **200** or stored on a machine that is accessible to Web server **201**, for later retrieval.

One embodiment of the claimed invention also provides a Web page designer with HTML extensions, or "dyna" tags. These dyna tags provide customized HTML functionality to a Web page designer, to allow the designer to build customized HTML templates that specify the source and placement of retrieved data. For example, in one embodiment, a "dynatext" HTML extension tag specifies a data source and a column name to allow the HTML template to identify the data source to log into and the column name from which to retrieve data. Alternatively, "dyna-anchor" tags allow the designer to build hyperlink queries while "dynablock" tags provide the designer with the ability to iterate through blocks of data. Page servers use these HTML templates to create dynamic Web pages. Then, as described above, these dynamic Web pages are either transmitted back to requesting Web client **200** or stored on a machine that is accessible to Web server **201**, for later retrieval.

The presently claimed invention provides numerous advantages over prior art Web servers, including advantages in the areas of performance, security, extensibility and scalability.

Performance

One embodiment of the claimed invention utilizes connection caching and page caching to improve performance. Each Page server can be configured to maintain a cache of connections to numerous data sources. For example, as illustrated in FIG. **4**, Page server **404**(1) can retrieve data from data source **406**, data source **408** or data source **410**. Page server **404**(1) can maintain connection cache **412**(1), containing connections to each of data source **406**, data source **408** and data source **410**, thus eliminating connect times from the Page servers to those data sources.

Additionally, another embodiment of the present invention supports the caching of finished Web pages, to optimize

the performance of the data source being utilized. This "page caching" feature, illustrated in FIG. **4** as Page cache **414**, allows the Web site administrator to optimize the performance of data sources by caching Web pages that are repeatedly accessed. Once the Web page is cached, subsequent requests or "hits" will utilize the cached Web page rather than re-accessing the data source. This can radically improve the performance of the data source.

Security

The present invention allows the Web site administrator to utilize multiple levels of security to manage the Web site. In one embodiment, the Page server can utilize all standard encryption and site security features provided by the Web server. In another embodiment, the Page server can be configured to bypass connection caches **412**(1)–(n), described above, for a particular data source and to require entry of a user-supplied identification and password for the particular data source the user is trying to access.

Additionally, another embodiment of the presently claimed invention requires no real-time access of data sources. The Web page caching ability, described above, enables additional security for those sites that want to publish non-interactive content from internal information systems, but do not want real-time Internet accessibility to those internal information systems. In this instance, the Page server can act as a "replication and staging gent" and create Web pages in batches, rather than in real-time. These "replicated" Web pages are then "staged" for access at a later time, and access o the Web pages in this scenario is possible even if the Page server and dispatcher are not present later.

In yet another embodiment, the Page server can make a single pass through a Web library, and compile a Web site that exists in the traditional form of separately available files. A Web library is a collection of related Web books and Web pages. More specifically, the Web library is a hierarchical organization of Web document templates, together with all the associated data source information. Information about an entire Web site is thus contained in a single physical file, thus simplifying the problem of deploying Web sites across multiple Page servers. The process of deploying the Web site in this embodiment is essentially a simple copy of a single file.

Extensibility

One embodiment of the present invention provides the Web site administrator with Object Linking and Embedding (OLE) 2.0 extensions to extend the page creation process. These OLE 2.0 extensions also allow information submitted over the Web to be processed with user-supplied functionality. Utilizing development tools such as Visual Basic, Visual C++ or PowerBuilder that support the creation of OLE 2.0 automation, the Web site administrator can add features and modify the behavior of the Page servers described above. This extensibility allows one embodiment of the claimed invention to be incorporated with existing technology to develop an infinite number of custom web servers.

For example, OLE 2.0 extensions allow a Web site administrator to encapsulate existing business rules in an OLE 2.0 automation interface, to be accessed over the Web. One example of a business rule is the steps involved in the payoff on an installment or mortgage loan. The payoff may involve, for example, taking into account the current balance, the date and the interest accrued since the last payment. Most organizations already have this type of business rule implemented using various applications, such as Visual Basic for client-server environments, or CICS programs on mainframes. If these applications are OLE 2.0 compliant, the Page server "dynaobject" HTML extension tag can be used to encapsulated the application in an OLE 2.0 automation interface. The Page server is thus extensible, and can incorporate the existing application with the new Page server functionality.

Scalability

One embodiment of the claimed invention allows "plug and play" scalability. As described above, referring to FIG. **4**, Dispatcher **402** maintains information about all the Page servers configured to be serviced by Dispatcher **402**. Any number of Page servers can thus be "plugged" into the configuration illustrated in FIG. **4**, and the Page servers will be instantly activated as the information is dynamically updated in Dispatcher **402**. The Web site administrator can thus manage the overhead of each Page server and modify each Page server's load, as necessary, to improve performance. In this manner, each Page server will cooperate with other Page servers within a multi-server environment. Dispatcher **402** can examine the load on each Page server and route new requests according to each Page server's available resources. This "load-balancing" across multiple Page servers can significantly increase a Web site's performance.

FIG. **5** illustrates the processing of a Web browser request in the form of a flow diagram, according to one embodiment of the presently claimed invention. A Web browser sends a URL request to a Web server in processing block **500**. In processing block **502**, the Web server receives the URL request, and an interceptor then intercepts the handling of the request in processing block **504**. The interceptor connects to a dispatcher and sends the URL request to the dispatcher in processing block **506**. In processing block **508**, the dispatcher determines which Page servers can handle the request. The dispatcher also determines which Page server is processing the fewest requests in processing block **510**, and in processing block **512**, the dispatcher sends the URL request to an appropriate Page server. The Page server receives the request and produces an HTML document in processing block **514**. The Page server then responds to the dispatcher with notification of the name of the cached HTML document in processing block **516**. In processing block **518**, the dispatcher responds to the interceptor with the document name, and the interceptor then replaces the requested URL with the newly generated HTML document in processing block **520**. The Web server then sends the new HTML document to the requesting client in processing block **522**. Finally, the Web browser receives and displays the HTML document created by the Page server at processing block **524**.

Thus, a method and apparatus for creating and managing custom Web sites is disclosed. These specific arrangements and methods described herein are merely illustrative of the principles of the present invention. Numerous modifications in form and detail may be made by those of ordinary skill in the art without departing from the scope of the present invention. Although this invention has been shown in relation to a particular preferred embodiment, it should not be considered so limited. Rather, the present invention is limited only by the scope of the appended claims.

We claim:

1. A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:

routing said request from said Web server to a page server, said page server receiving said request and releasing

said Web server to process other requests, wherein said routing step further includes the steps of intercepting said request at said Web server, routing said request from said Web server to a dispatcher, and dispatching said request to said page server;

processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and

dynamically generating a Web page in response to said request, said Web page including data dynamically retrieved from one or more data sources.

2. The computer-implemented method in claim **1** wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.

3. The computer-implemented method in claim **2** wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.

4. The computer-implemented method in claim **3** wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.

5. The computer-implemented method in claim **3** wherein said step of processing said request includes the step of logging into said one or more data sources.

6. The computer-implemented method in claim **3** wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.

7. The computer-implemented method in claim **3** wherein said page server includes custom HTML extension templates for configuring said Web page.

8. The computer-implemented method in claim **7** wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said custom HTML extension templates.

9. A networked system for managing a dynamic Web page generation request, said system comprising:

one or more data sources;

a page server having a processing means;

a first computer system including means for generating said request; and

a second computer system including means for receiving said request from said first computer, said second computer system also including a router, said router routing said request from said second computer system to said page server, wherein said routing further includes intercepting said request at said second computer, routing said request from said second computer to a dispatcher, and dispatching said request to said page server said page server receiving said request and releasing said second computer system to process other requests, said page server processing means processing said request and dynamically generating a Web page in response to said request, said Web page including data dynamically retrieved from said one or more data sources.

10. The networked system in claim **9** wherein said router in said second computer system includes:

an interceptor intercepting said request at said second computer system and routing said request; and

a dispatcher receiving said routed request from said interceptor and dispatching said request to said page server.

11. A machine readable medium having stored thereon data representing sequences of instructions, which when executed by a computer system, cause said computer system to perform the steps of:

routing a dynamic Web page generation request from a Web server to a page server, said page server receiving said request and releasing said Web server to process other requests wherein said routing step further includes the steps of intercepting said request at said Web server, routing said request from said Web server to a dispatcher, and dispatching said request to said page server;

processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and

dynamically generating a Web page, said Web page including data retrieved from one or more data sources.

* * * * *

Exhibit 2

US006415335B1

(12) **United States Patent**
Lowery et al.

(10) **Patent No.: US 6,415,335 B1**
(45) **Date of Patent: *Jul. 2, 2002**

(54) **SYSTEM AND METHOD FOR MANAGING DYNAMIC WEB PAGE GENERATION REQUESTS**

(75) Inventors: **Keith Lowery**, Richardson; **Andrew B. Levine**, Plano; **Ronald L. Howell**, Rowlett, all of TX (US)

(73) Assignee: **epicRealm Operating Inc.**, Richardson, TX (US)

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

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **09/234,048**

(22) Filed: **Jan. 19, 1999**

Related U.S. Application Data

(62) Division of application No. 08/636,477, filed on Apr. 23, 1996, now Pat. No. 5,894,554.

(51) **Int. Cl.**[7] **G06F 13/14**; G06F 13/20

(52) **U.S. Cl.** **710/5**; 710/7; 709/219; 709/223; 709/238

(58) **Field of Search** 709/238, 223, 709/219; 710/5, 7, 20–21

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,866,706	A		9/1989	Christophersen et al. .. 370/85.7
5,341,499	A		8/1994	Doragh 395/700
5,392,400	A		2/1995	Berkowitz et al. 395/200
5,404,522	A		4/1995	Carmon et al. 395/650
5,404,523	A		4/1995	DellaFera et al. 395/650
5,404,527	A	*	4/1995	Irwin et al. 395/700
5,452,460	A		9/1995	Distelberg et al. 395/700
5,532,838	A		7/1996	Barbari 358/400
5,701,463	A	*	12/1997	Malcolm 395/610
5,751,956	A		5/1998	Kirsch 395/200.33
5,752,246	A	*	5/1998	Rogers et al. 707/10
5,754,772	A	*	5/1998	Leaf 395/200.33
5,761,673	A	*	6/1998	Bookman et al. 707/104
5,774,660	A		6/1998	Brendel et al. 395/200.31
5,774,668	A		6/1998	Choquier et al. 395/200.53

OTHER PUBLICATIONS

Hoffner, 'Inter–operability and distributed application platform design', Web URL:http://www.ansa.co.uk/, 1995, pp. 342–356.*

Mourad et al, 'Scalable Web Server Architectures', IEEE, Jun. 1997, pp. 12–16.*

Dias et al, 'A Scalable and Highly Available Web Server', IEEE, 1996, pp. 85–92.*

'Single System Image and Load Balancing for Network Access to a Loosely Coupled Complex', IBM TDB, vol. 34, Feb. 1992, pp. 464–467.*

Dias, Daniel M., et al.; A Scalable and Highly Available Web Server; IBM Research Division; T.J. Watson Research Center; 7 pages.

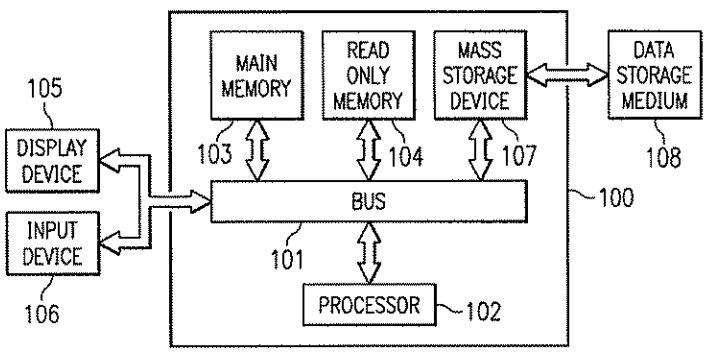
(List continued on next page.)

Primary Examiner—Jeffrey Gaffin
Assistant Examiner—Rehana Perveen
(74) *Attorney, Agent, or Firm*—Baker Botts L.L.P.

(57) **ABSTRACT**

The present invention teaches a method and apparatus for creating and managing custom Web sites. Specifically, one embodiment of the present invention claims a computer-implemented method for managing a dynamic Web page generation request to a Web server, the computer-implemented method comprising the steps of routing the request from the Web server to a page server, the page server receiving the request and releasing the Web server to process other requests, processing the request, the processing being performed by the page server concurrently with the Web server, as the Web server processes the other requests, and dynamically generating a Web page in response to the request, the Web page including data dynamically retrieved from one or more data sources.

29 Claims, 4 Drawing Sheets



OTHER PUBLICATIONS

Andresen, Daniel, Et Al.; Scalability Issues for High Performance Digital Libraries on the World Wide Web; Department of Computer Science; University of California at Santa Barbara; 10 pages.

Andresen, Daniel, Et Al.; SWEB: Towards a Scalable World Wide Web Server on Multicomputers; Department of Computer Science; University of California at Santa Barbara; 7 pages.

Holmedahl, Vegard; Et Al.; Cooperative Caching of Dynamic Content on a Distributed Web Server; Department of Computer Science; University of California at Santa Barbara; 8 pages.

Overson, Nicole; NeXT Ships WebObjects—On Time—As Promised; Deja.com: NeXT Ships WebObjects—On Time—As Promishttp://X28..deja.com/=dnc/ST_m=ps...EXT=927585438. 1744765032&hitnum=33.

International Search Report; 7 pages; dated Aug. 21, 1997.

Birman, Kenneth P. and van Renesse, Robbert; Software for Reliable Networks; Scientific American; May 1996; pp. 64–69.

"Beyond the Web: Excavating the Real World Via Mosaic"; Goldberg et al.; Second International WWW Conference; Oct. 17, 1994.

* cited by examiner

FIG. 1

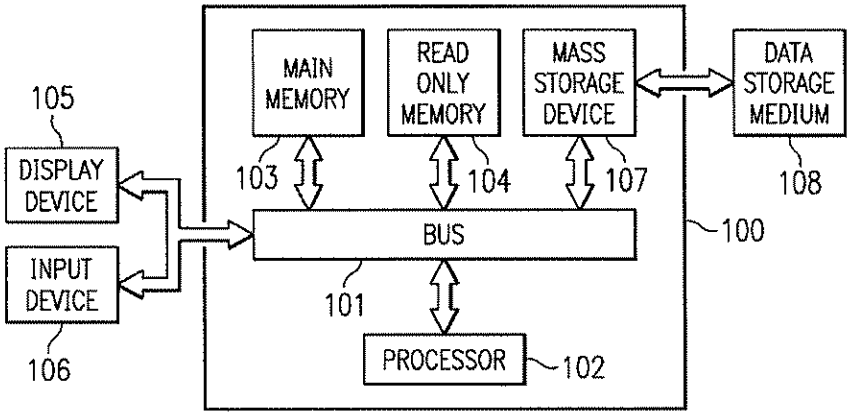


FIG. 2
(PRIOR ART)

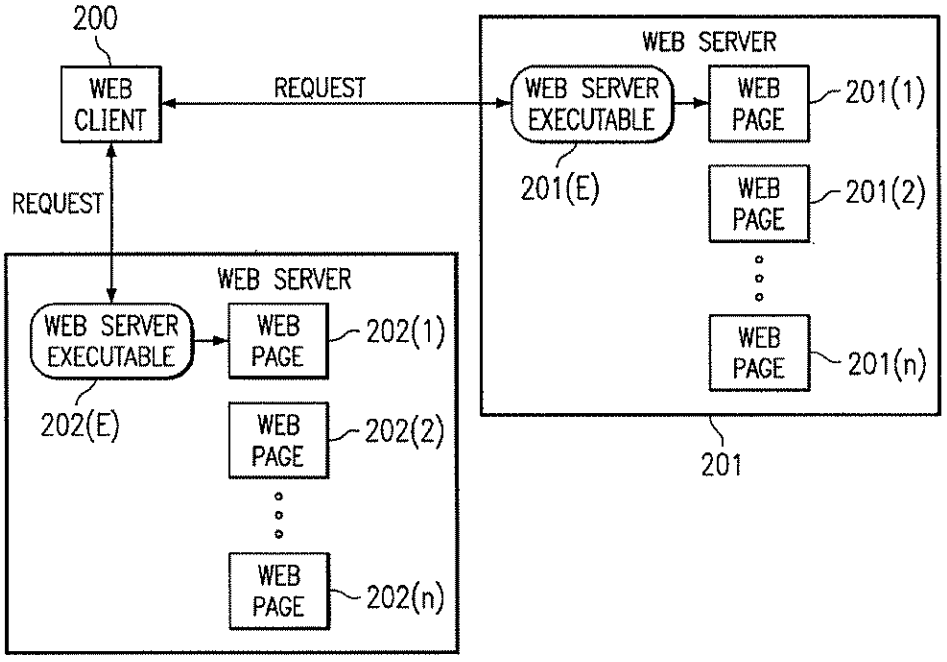
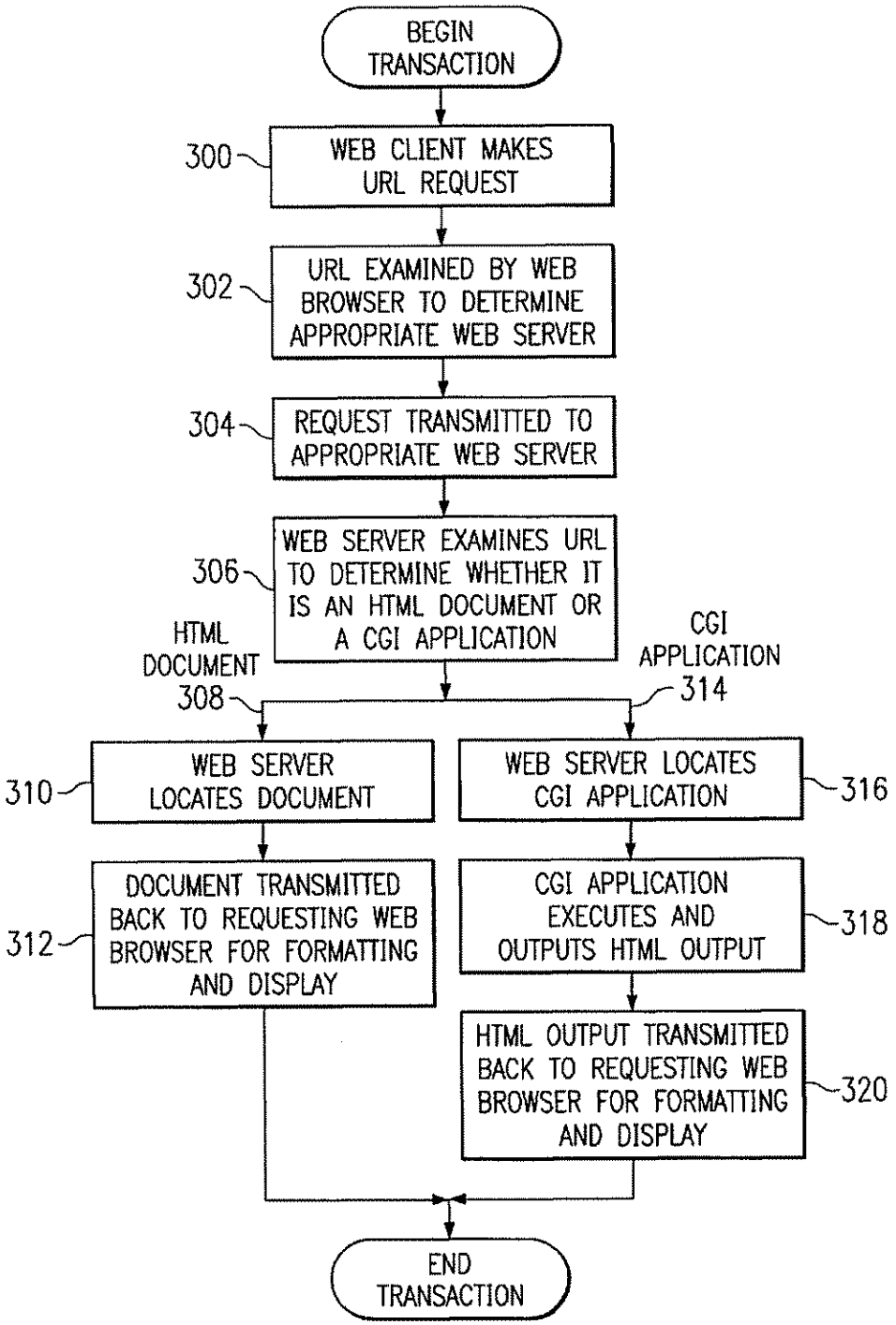


FIG. 3
(PRIOR ART)



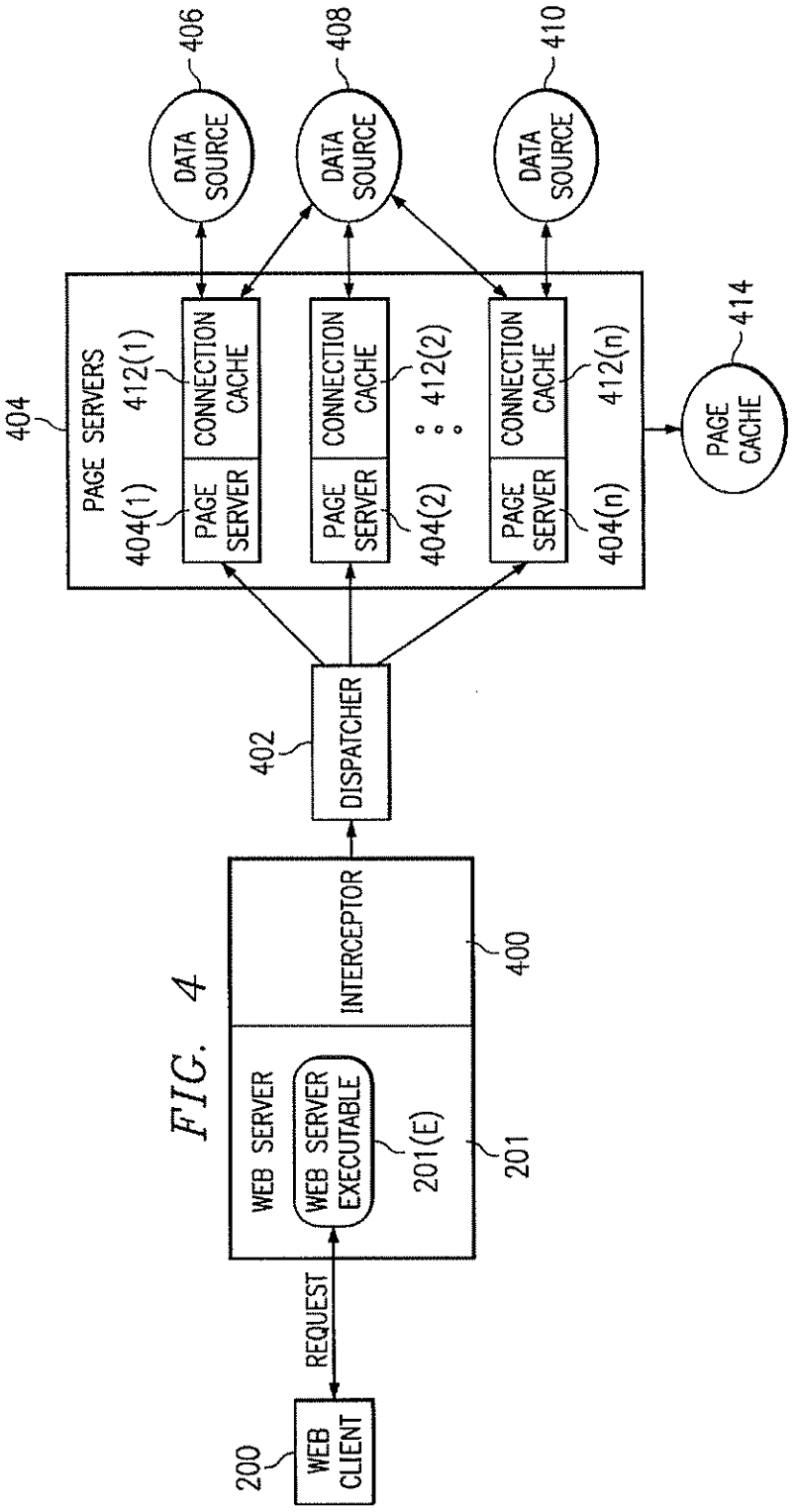
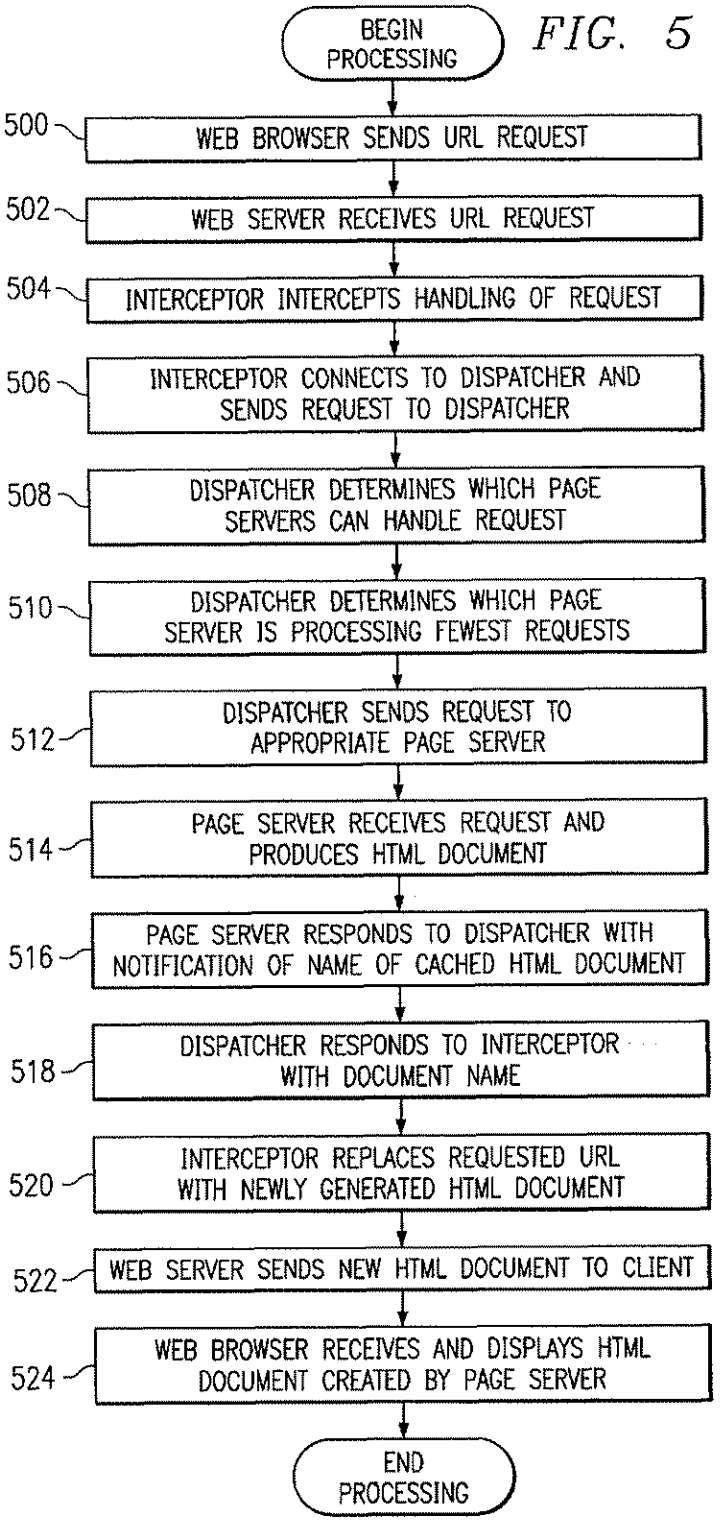

406
408
410
DATA SOURCE
DATA SOURCE
DATA SOURCE
404
PAGE SERVERS
412(1)
CONNECTION CACHE
PAGE SERVER
404(1)
CONNECTION CACHE
PAGE SERVER
404(2)
412(2)
CONNECTION CACHE
PAGE SERVER
404(n)
412(n)
414
PAGE CACHE
402
DISPATCHER
INTERCEPTOR
400
WEB SERVER
WEB SERVER EXECUTABLE
201(E)
201
REQUEST
200
WEB CLIENT

FIG. 4

FIG. 5


BEGIN PROCESSING
500
WEB BROWSER SENDS URL REQUEST
502
WEB SERVER RECEIVES URL REQUEST
504
INTERCEPTOR INTERCEPTS HANDLING OF REQUEST
506
INTERCEPTOR CONNECTS TO DISPATCHER AND SENDS REQUEST TO DISPATCHER
508
DISPATCHER DETERMINES WHICH PAGE SERVERS CAN HANDLE REQUEST
510
DISPATCHER DETERMINES WHICH PAGE SERVER IS PROCESSING FEWEST REQUESTS
512
DISPATCHER SENDS REQUEST TO APPROPRIATE PAGE SERVER
514
PAGE SERVER RECEIVES REQUEST AND PRODUCES HTML DOCUMENT
516
PAGE SERVER RESPONDS TO DISPATCHER WITH NOTIFICATION OF NAME OF CACHED HTML DOCUMENT
518
DISPATCHER RESPONDS TO INTERCEPTOR WITH DOCUMENT NAME
520
INTERCEPTOR REPLACES REQUESTED URL WITH NEWLY GENERATED HTML DOCUMENT
522
WEB SERVER SENDS NEW HTML DOCUMENT TO CLIENT
524
WEB BROWSER RECEIVES AND DISPLAYS HTML DOCUMENT CREATED BY PAGE SERVER
END PROCESSING

SYSTEM AND METHOD FOR MANAGING DYNAMIC WEB PAGE GENERATION REQUESTS

This application is a division of Ser. No. 08/636,477, filed Apr. 23, 1996, now U.S. Pat. No. 5,894,554.

FIELD OF THE INVENTION

The present invention relates to the field of Internet technology. Specifically, the present invention relates to the creation and management of custom World Wide Web sites.

DESCRIPTION OF RELATED ART

The World Wide Web (the Web) represents all of the computers on the Internet that offer users access to information on the Internet via interactive documents or Web pages. These Web pages contain hypertext links that are used to connect any combination of graphics, audio, video and text, in a non-linear, non-sequential manner. Hypertext links are created using a special software language known as HyperText Mark-Up Language (HTML).

Once created, Web pages reside on the Web, on Web servers or Web sites. A Web site can contain numerous Web pages. Web client machines running Web browsers can access these Web pages at Web sites via a communications protocol known as HyperText Transport Protocol (HTTP). Web browsers are software interfaces that run on World Wide Web clients to allow access to Web sites via a simple user interface. A Web browser allows a Web client to request a particular Web page from a Web site by specifying a Uniform Resource Locator (URL). A URL is a Web address that identifies the Web page and its location on the Web. When the appropriate Web site receives the URL, the Web page corresponding to the requested URL is located, and if required, HTML output is generated. The HTML output is then sent via HTTP to the client for formatting on the client's screen.

Although Web pages and Web sites are extremely simple to create, the proliferation of Web sites on the Internet highlighted a number of problems. The scope and ability of a Web page designer to change the content of the Web page was limited by the static nature of Web pages. Once created, a Web page remained static until it was manually modified. This in turn limited the ability of Web site managers to effectively manage their Web sites.

The Common Gateway Interface (CGI) standard was developed to resolve the problem of allowing dynamic content to be included in Web pages. CGI "calls" or procedures enable applications to generate dynamically created HTML output, thus creating Web pages with dynamic content. Once created, these CGI applications do not have to be modified in order to retrieve "new" or dynamic data. Instead, when the Web page is invoked, CGI "calls" or procedures are used to dynamically retrieve the necessary data and to generate a Web page.

CGI applications also enhanced the ability of Web site administrators to manage Web sites. Administrators no longer have to constantly update static Web pages. A number of vendors have developed tools for CGI based development, to address the issue of dynamic Web page generation. Companies like Spider™ and Bluestone™, for example, have each created development tools for CGI-based Web page development. Another company, Haht Software™, has developed a Web page generation tool that uses a BASIC-like scripting language, instead of a CGI scripting language.

Tools that generate CGI applications do not, however, resolve the problem of managing numerous Web pages and requests at a Web site. For example, a single company may maintain hundreds of Web pages at their Web site. Current Web server architecture also does not allow the Web server to efficiently manage the Web page and process Web client requests. Managing these hundreds of Web pages in a coherent manner and processing all requests for access to the Web pages is thus a difficult task. Existing development tools are limited in their capabilities to facilitate dynamic Web page generation, and do not address the issue of managing Web requests or Web sites.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a method and apparatus for creating and managing custom Web sites. Specifically, the present invention claims a method and apparatus for managing dynamic web page generation requests.

In one embodiment, the present invention claims a computer-implemented method for managing a dynamic Web page generation request to a Web server, the computer-implemented method comprising the steps of routing the request from the Web server to a page server, the page server receiving the request and releasing the Web server to process other requests, processing the request, the processing being performed by the page server concurrently with the Web server, as the Web server processes the other requests, and dynamically generating a Web page in response to the request, the Web page including data dynamically retrieved from one or more data sources. Other embodiments also include connection caches to the one or more data sources, page caches for each page server, and custom HTML extension templates for configuring the Web page.

Other objects, features and advantages of the present invention will be apparent from the accompanying drawings and from the detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates a typical computer system in which the present invention operates.

FIG. **2** illustrates a typical prior art Web server environment.

FIG. **3** illustrates a typical prior art Web server environment in the form of a flow diagram.

FIG. **4** illustrates one embodiment of the presently claimed invention.

FIG. **5** illustrates the processing of a Web browser request in the form of a flow diagram, according to one embodiment of the presently claimed invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention relates to a method and apparatus for creating and managing custom Web sites. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent to one of ordinary skill in the art, however, that these specific details need not be used to practice the present invention. In other instances, well-known structures, interfaces and processes have not been shown in detail in order not to unnecessarily obscure the present invention.

FIG. **1** illustrates a typical computer system **100** in which the present invention operates. The preferred embodiment of

the present invention is implemented on an IBM™ Personal Computer manufactured by IBM Corporation of Armonk, New York. An alternate embodiment may be implemented on an RS/6000™ Workstation manufactured by IBM Corporation of Armonk, New York. It will be apparent to those of ordinary skill in the art that other computer system architectures may also be employed.

In general, such computer systems as illustrated by FIG. **1** comprise a bus **101** for communicating information, a processor **102** coupled with the bus **101** for processing information, main memory **103** coupled with the bus **101** for storing information and instructions for the processor **102**, a read-only memory **104** coupled with the bus **101** for storing static information and instructions for the processor **102**, a display device **105** coupled with the bus **101** for displaying information for a computer user, an input device **106** coupled with the bus **101** for communicating information and command selections to the processor **102**, and a mass storage device **107**, such as a magnetic disk and associated disk drive, coupled with the bus **101** for storing information and instructions. A data storage medium **108** containing digital information is configured to operate with mass storage device **107** to allow processor **102** access to the digital information on data storage medium **108** via bus **101**.

Processor **102** may be any of a wide variety of general purpose processors or microprocessors such as the Pentium™ microprocessor manufactured by Intel™ Corporation or the RS/6000™ processor manufactured by IBM Corporation. It will be apparent to those of ordinary skill in the art, however, that other varieties of processors may also be used in a particular computer system. Display device **105** may be a liquid crystal device, cathode ray tube (CRT), or other suitable display device. Mass storage device **107** may be a conventional hard disk drive, floppy disk drive, CD-ROM drive, or other magnetic or optical data storage device for reading and writing information stored on a hard disk, a floppy disk, a CD-ROM a magnetic tape, or other magnetic or optical data storage medium. Data storage medium **108** may be a hard disk, a floppy disk, a CD-ROM, a magnetic tape, or other magnetic or optical data storage medium.

In general, processor **102** retrieves processing instructions and data from a data storage medium **108** using mass storage device **107** and downloads this information into random access memory **103** for execution. Processor **102**, then executes an instruction stream from random access memory **103** or read-only memory **104**. Command selections and information input at input device **106** are used to direct the flow of instructions executed by processor **102**. Equivalent input device **106** may also be a pointing device such as a conventional mouse or trackball device. The results of this processing execution are then displayed on display device **105**.

The preferred embodiment of the present invention is implemented as a software module, which may be executed on a computer system such as computer system **100** in a conventional manner. Using well known techniques, the application software of the preferred embodiment is stored on data storage medium **108** and subsequently loaded into and executed within computer system **100**. Once initiated, the software of the preferred embodiment operates in the manner described below.

FIG. **2** illustrates a typical prior art Web server environment. Web client **200** can make URL requests to Web server **201** or Web server **202**. Web servers **201** and **202** include Web server executables, **201**(E) and **202**(E) respectively, that perform the processing of Web client requests. Each Web server may have a number of Web pages **201**(1)–(n) and **202**(1)–(n). Depending on the URL specified by the Web client **200**, the request may be routed by either Web server executable **201**(E) to Web page **201** (1), for example, or from Web server executable **202**(E) to Web page **202** (1). Web client **200** can continue making URL requests to retrieve other Web pages. Web client **200** can also use hyperlinks within each Web page to "jump" to other Web pages or to other locations within the same Web page.

FIG. **3** illustrates this prior art Web server environment in the form of a flow diagram. In processing block **300**, the Web client makes a URL request. This URL request is examined by the Web browser to determine the appropriate Web server to route the request to in processing block **302**. In processing block **304** the request is then transmitted from the Web browser to the appropriate Web server, and in processing block **306** the Web server executable examines the URL to determine whether it is a HTML document or a CGI application. If the request is for an HTML document **308**, then the Web server executable locates the document in processing block **310**. The document is then transmitted back through the requesting Web browser for formatting and display in processing block **312**.

If the URL request is for a CGI application **314**, however, the Web server executable locates the CGI application in processing block **316**. The CGI application then executes and outputs HTML output in processing block **318** and finally, the HTML output is transmitted back to requesting Web browser for formatting and display in processing block **320**.

This prior art Web server environment does not, however, provide any mechanism for managing the Web requests or the Web sites. As Web sites grow, and as the number of Web clients and requests increase, Web site management becomes a crucial need.

For example, a large Web site may receive thousands of requests or "hits" in a single day. Current Web servers process each of these requests on a single machine, namely the Web server machine. Although these machines may be running "multi-threaded" operating systems that allow transactions to be processed by independent "threads," all the threads are nevertheless on a single machine, sharing a processor. As such, the Web executable thread may hand off a request to a processing thread, but both threads will still have to be handled by the processor on the Web server machine. When numerous requests are being simultaneously processed by multiple threads on a single machine, the Web server can slow down significantly and become highly inefficient. The claimed invention addresses this need by utilizing a partitioned architecture to facilitate the creation and management of custom Web sites and servers.

FIG. **4** illustrates one embodiment of the presently claimed invention. Web client **200** issues a URL request that is processed to determined proper routing. In this embodiment, the request is routed to Web server **201**. Instead of Web server executable **201**(E) processing the URL request, however, Interceptor **400** intercepts the request and routes it to Dispatcher **402**. In one embodiment, Interceptor **400** resides on the Web server machine as an extension to Web server **201**. This embodiment is appropriate for Web servers such as Netsite™ from Netscape, that support such extensions. A number of public domain Web servers, such as NCSA™ from the National Center for Supercomputing Applications at the University of Illinois, Urbana-Champaign, however, do not provide support for

this type of extension. Thus, in an alternate embodiment, Interceptor **400** is an independent module, connected via an "intermediate program" to Web server **201**. This intermediate program can be a simple CGI application program that connects Interceptor **400** to Web server **201**. Alternate intermediate programs the perform the same functionality can also be implemented.

In one embodiment of the invention, Dispatcher **402** resides on a different machine than Web server **201**. This embodiment overcomes the limitation described above, in prior art Web servers, wherein all processing is performed by the processor on a single machine. By routing the request to Dispatcher **402** residing on a different machine than the Web server executable **201**(E), the request can then be processed by a different processor than the Web server executable **201**(E). Web server executable **201**(E) is thus free to continue servicing client requests on Web server **201** while the request is processed "off-line," at the machine on which Dispatcher **402** resides.

Dispatcher **402** can, however, also reside on the same machine as the Web server. The Web site administrator has the option of configuring Dispatcher **402** on the same machine as Web server **201**, taking into account a variety of factors pertinent to a particular Web site, such as the size of the Web site, the number of Web pages and the number of hits at the Web site. Although this embodiment will not enjoy the advantage described above, namely off-loading the processing of Web requests from the Web server machine, the embodiment does allow flexibility for a small Web site to grow. For example, a small Web site administrator can use a single machine for both Dispatcher **402** and Web server **201** initially, then off-load Dispatcher **402** onto a separate machine as the Web site grows. The Web site can thus take advantage of other features of the present invention regardless of whether the site has separate machines configured as Web servers and dispatchers.

Dispatcher **402** receives the intercepted request and then dispatches the request to one of a number of Page servers **404 (1)**–(n). For example, if Page server **404 (1)** receives the dispatched request, it processes the request and retrieves the data from an appropriate data source, such as data source **406**, data source **408**, or data source **410**. Data sources, as used in the present application, include databases, spreadsheets, files and any other type of data repository. Page server **404 (1)** can retrieve data from more than one data source and incorporate the data from these multiple data sources in a single Web page.

In one embodiment, each Page server **404(1)**–(n) resides on a separate machine on the network to distribute the processing of the request. Dispatcher **402** maintains a variety of information regarding each Page server on the network, and dispatches requests based on this information. For example, Dispatcher **402** retains dynamic information regarding the data sources that any given Page server can access. Dispatcher **402** thus examines a particular request and determines which Page servers can service the URL request. Dispatcher **402** then hands off the request to the appropriate Page server.

For example, if the URL request requires financial data from data source **408**, dispatcher **402** will first examine an information list. Dispatcher **402** may determine that Page server **404(3)**, for example, has access to the requisite data in data source **408**. Dispatcher **402** will thus route the URL request to Page server **404(3)**. This "connection caching" functionality is described in more detail below, under the heading "Performance." Alternately, Dispatcher **402** also has the ability to determine whether a particular Page server already has the necessary data cached in the Page server's page cache (described in more detail below, under the heading "Performance"). Dispatcher **402** may thus determine that Page server **404(1)** and **404(2)** are both logged into Data source **408**, but that Page server **404(2)** has the financial information already cached in Page server **404(2)**'s page cache. In this case, Dispatcher **402** will route the URL request to Page server **404(2)** to more efficiently process the request.

Finally, Dispatcher **402** may determine that a number or all Page servers **404(1)**–(n) are logged into Data source **408**. In this scenario, Dispatcher **402** can examine the number of requests that each Page server is servicing and route the request to the least busy page server. This "load balancing" capability can significantly increase performance at a busy Web site and is discussed in more detail below, under the heading "Scalability".

If, for example, Page server **404(2)**, receives the request, Page server **404(2)** will process the request. While Page server **404(2)** is processing the request, Web server executable **201**(E) can concurrently process other Web client requests. This partitioned architecture thus allows both Page server **404(2)** and Web server executable **201**(E) to. simultaneously process different requests, thus increasing the efficiency of the Web site. Page server **404(2)** dynamically generates a Web page in response to the Web client request, and the dynamic Web page is then either transmitted back to requesting Web client **200** or stored on a machine that is accessible to Web server **201**, for later retrieval.

One embodiment of the claimed invention also provides a Web page designer with HTML extensions, or "dyna" tags. These dyna tags provide customized HTML functionality to a Web page designer, to allow the designer to build customized HTML templates that specify the source and placement of retrieved data. For example, in one embodiment, a "dynatext" HTML extension tag specifies a data source and a column name to allow the HTML template to identify the data source to log into and the column name from which to retrieve data. Alternatively, "dyna-anchor" tags allow the designer to build hyperlink queries while "dynablock" tags provide the designer with the ability to iterate through blocks of data. Page servers use these HTML templates to create dynamic Web pages. Then, as described above, these dynamic Web pages are either transmitted back to requesting Web client **200** or stored on a machine that is accessible to Web server **201**, for later retrieval.

The presently claimed invention provides numerous advantages over prior art Web servers, including advantages in the areas of performance, security, extensibility and scalability

Performance

One embodiment of the claimed invention utilizes connection caching and page caching to improve performance. Each Page server can be configured to maintain a cache of connections to numerous data sources. For example, as illustrated in FIG. **4**, Page server **404(1)** can retrieve data from data source **406**, data source **408** or data source **410**. Page server **404(1)** can maintain connection cache **412(1)**, containing connections to each of data source **406**, data source **408** and data source **410**, thus eliminating connect times from the Page servers to those data sources.

Additionally, another embodiment of the present invention supports the caching of finished Web pages, to optimize the performance of the data source being utilized. This "page

caching" feature, illustrated in FIG. **4** as Page cache **414**, allows the Web site administrator to optimize the performance of data sources by caching Web pages that are repeatedly accessed. Once the Web page is cached, subsequent requests or "hits" will utilize the cached Web page rather than re-accessing the data source. This can radically improve the performance of the data source.

Security

The present invention allows the Web site administrator to utilize multiple levels of security to manage the Web site. In one embodiment, the Page server can utilize all standard encryption and site security features provided by the Web server. In another embodiment, the Page server can be configured to bypass connection caches **412(1)**–(n), described above, for a particular data source and to require entry of a user-supplied identification and password for the particular data source the user is trying to access.

Additionally, another embodiment of the presently claimed invention requires no real-time access of data sources. The Web page caching ability, described above, enables additional security for those sites that want to publish non-interactive content from internal information systems, but do not want real-time Internet accessibility to those internal information systems. In this instance, the Page server can act as a "replication and staging agent" and create Web pages in batches, rather than in real-time. These "replicated" Web pages are then "staged" for access at a later time, and access to the Web pages in this scenario is possible even if the Page server and dispatcher are not present later.

In yet another embodiment, the Page server can make a single pass through a Web library, and compile a Web site that exists in the traditional form of separately available files. A Web library is a collection of related Web books and Web pages. More specifically, the Web library is a hierarchical organization of Web document templates, together with all the associated data source information. Information about an entire Web site is thus contained in a single physical file, thus simplifying the problem of deploying Web sites across multiple Page servers. The process of deploying the Web site in this embodiment is essentially a simple copy of a single file.

Extensibility

One embodiment of the present invention provides the Web site administrator with Object Linking and Embedding (OLE) 2.0 extensions to extend the page creation process. These OLE 2.0 extensions also allow information submitted over the Web to be processed with user-supplied functionality. Utilizing development tools such as Visual Basic, Visual C++ or PowerBuilder that support the creation of OLE 2.0 automation, the Web site administrator can add features and modify the behavior of the Page servers described above. This extensibility allows one embodiment of the claimed invention to be incorporated with existing technology to develop an infinite number of custom web servers.

For example, OLE 2.0 extensions allow a Web site administrator to encapsulate existing business rules in an OLE 2.0 automation interface, to be accessed over the Web. One example of a business rule is the steps involved in the payoff on an installment or mortgage loan. The payoff may involve, for example, taking into account the current balance, the date and the interest accrued since the last payment. Most organizations already have this type of business rule implemented using various applications, such as Visual Basic for client-server environments, or CICS programs on mainframes. If these applications are OLE 2.0 compliant, the Page server "dynaobject" HTML extension tag can be used to encapsulated the application in an OLE 2.0 automation interface. The Page server is thus extensible, and can incorporate the existing application with the new Page server functionality.

Scalability

One embodiment of the claimed invention allows "plug and play" scalability. As described above, referring to FIG. **4**, Dispatcher **402** maintains information about all the Page servers configured to be serviced by Dispatcher **402**. Any number of Page servers can thus be "plugged" into the configuration illustrated in FIG. **4**, and the Page servers will be instantly activated as the information is dynamically updated in Dispatcher **402**. The Web site administrator can thus manage the overhead of each Page server and modify each Page server's load, as necessary, to improve performance. In this manner, each Page server will cooperate with other Page servers within a multi-server environment. Dispatcher **402** can examine the load on each Page server and route new requests according to each Page server's available resources. This "load-balancing" across multiple Page servers can significantly increase a Web site's performance.

FIG. **5** illustrates the processing of a Web browser request in the form of a flow diagram, according to one embodiment of the presently claimed invention. A Web browser sends a URL request to a Web server in processing block **500**. In processing block **502**, the Web server receives the URL request, and an interceptor then intercepts the handling of the request in processing block **504**. The interceptor connects to a dispatcher and sends the URL request to the dispatcher in processing block **506**. In processing block **508**, the dispatcher determines which Page servers can handle the request. The dispatcher also determines which Page server is processing the fewest requests in processing block **510**, and in processing block **512**, the dispatcher sends the URL request to an appropriate Page server. The Page server receives the request and produces an HTML document in processing block **514**. The Page server then responds to the dispatcher with notification of the name of the cached HTML document in processing block **516**. In processing block **518**, the dispatcher responds to the interceptor with the document name, and the interceptor then replaces the requested URL with the newly generated HTML document in processing block **520**. The Web server then sends the new HTML document to the requesting client in processing block **522**. Finally, the Web browser receives and displays the HTML document created by the Page server at processing block **524**.

Thus, a method and apparatus for creating and managing custom Web sites is disclosed. These specific arrangements and methods described herein are merely illustrative of the principles of the present invention. Numerous modifications in form and detail may be made by those of ordinary skill in the art without departing from the scope of the present invention. Although this invention has been shown in relation to a particular preferred embodiment, it should not be considered so limited. Rather, the present invention is limited only by the scope of the appended claims.

We claim:

1. A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:

routing a request from a Web server to a page server, said page server receiving said request and releasing said

Web server to process other requests wherein said routing step further includes the steps of:

intercepting said request at said Web server and routing said request to said page server;

processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and

dynamically generating a Web page in response to said request, said Web page including data dynamically retrieved from one or more data sources.

2. The computer-implemented method in claim 1 wherein said step of routing said request includes the steps of:

routing said request from said Web server to a dispatcher; and

dispatching said request to said page server.

3. The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.

4. The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.

5. The computer-implemented method in claim 1 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.

6. The computer-implemented method in claim 1 wherein said step of processing said request includes the step of logging into said one or more data sources.

7. The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.

8. The computer-implemented method in claim 1 wherein said page server includes tag-based text templates for configuring said Web page.

9. The computer-implemented method in claim 8 wherein said step of processing said request further includes the step of inserting said-dynamically retrieved data from said one or more data sources into said tag-based text templates.

10. The computer-implemented method in claim 8 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request.

11. The computer-implemented method in claim 8 wherein said tag-based text templates include HTML templates.

12. The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources.

13. The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of processing an object handling extension.

14. The computer-implemented method in claim 13 wherein said object handling extension is an OLE extension.

15. A computer-implemented method comprising the steps of:

transferring a request from an HTTP-compliant device to a page server, said page server receiving said request and releasing said HTTP-compliant device to process other requests wherein said transferring step further includes the steps of:

intercepting said request at said HTTP-compliant device and transferring said request to said page server;

processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and

dynamically generating a page in response to said request, said page including data dynamically retrieved from one or more data sources.

16. The computer-implemented method in claim 15 wherein said step of transferring said request includes the steps of:

transferring said request from said HTTP-compliant device to a dispatcher; and

dispatching said request to said page server.

17. The computer-implemented method in claim 15 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.

18. The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of dynamically retrieving said data from said one or more data sources.

19. The computer-implemented method in claim 15 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.

20. The computer-implemented method in claim 15 wherein said step of processing said request includes the step of logging into said one or more data sources.

21. The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of maintaining a page cache containing said page.

22. The computer-implemented method in claim 15 wherein said page server includes tag-based text templates for configuring said page.

23. The computer-implemented method in claim 22 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said tag-based text templates.

24. The computer-implemented method in claim 22 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request.

25. The computer-implemented method in claim 22 wherein said tag-based text templates include HTML templates.

26. The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources.

27. The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of processing an object handling extension.

28. The computer-implemented method in claim 27 wherein said object handling extension is an OLE extension.

29. A computer-implemented method comprising the steps of:

transferring a request from an HTTP-compliant device to a dispatcher;

maintaining dynamic information regarding data sources a given page server may access;

dispatching said request to an appropriate page server based on said request and based on said dynamic information, said page server receiving said request and

releasing said HTTP-compliant device to process other requests;

processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and

dynamically generating a page in response to said request, said page including data dynamically retrieved from one or more data sources.

* * * * *

Exhibit 3

US005761673A

United States Patent [19]

Bookman et al.

[11] **Patent Number: 5,761,673**

[45] **Date of Patent: Jun. 2, 1998**

[54] **METHOD AND APPARATUS FOR GENERATING DYNAMIC WEB PAGES BY INVOKING A PREDEFINED PROCEDURAL PACKAGE STORED IN A DATABASE**

[75] Inventors: **Matthew Bookman**, Los Gatos; **John Francis Haverty**, La Honda; **Magnus Mard Lomroth**, Redwood City; **Teresita Katrina Rodriquez Montinola**; **Joseph Charles Pistritto**, both of Belmont, all of Calif.

[73] Assignee: **Oracle Corporation**, Redwood Shores, Calif.

[21] Appl. No.: **594,686**

[22] Filed: **Jan. 31, 1996**

[51] **Int. Cl.**[6] **G06F 17/30**

[52] **U.S. Cl.** **707/104**; 707/10; 395/680

[58] **Field of Search** 395/793, 603, 395/610, 184.01, 682, 356, 733, 357, 617, 680; 707/10, 104

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,530,852	6/1996	Meske, Jr. et al.	395/610
5,537,586	7/1996	Amram et al.	395/603
5,572,643	11/1996	Judson	395/793
5,701,451	12/1997	Rogers et al.	707/104
5,710,918	1/1998	Lagarde et al.	395/680

OTHER PUBLICATIONS

"Oracle Version 7", Computergram International, Jun. 17, 1992.

"Oracle Rides the Internet with Web", Computergram International, Feb. 10, 1995.

"Sun Java, Netscape Livescript Married as Javascript", Computergram International, Dec. 1995.

Marshall et al. "Industry Lines Up for Java—Vendors Set Stage for Dynamic", Commications Week, Dec. 1995.

"Oracle Unveils Oracle Websystem Internet Line", Computergram International, Nov. 1995.

Marshall, Martin "Sybase Sites to Get Web Links", Communciations Week, Nov. 1995.

(List continued on next page.)

Primary Examiner—Thomas G. Black

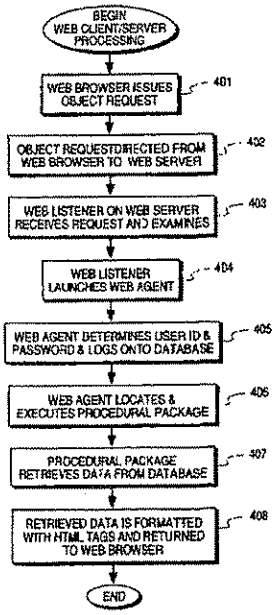
Assistant Examiner—Charles L. Rones

Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman LLP

[57] **ABSTRACT**

A method and apparatus for generating dynamic Web pages is disclosed. Specifically, the present invention claims a method and apparatus for generating dynamic Web pages on a Web server by invoking and executing predefined procedural packages stored in a database. The claimed invention receives an object request on the Web server and activates a Web agent on the Web server based on the object request. The Web agent invokes and executes the predefined procedural package to retrieve data from a data repository, and then formats the retrieved data as HTML output.

7 Claims, 4 Drawing Sheets



OTHER PUBLICATIONS

Karpinski, Richard "Oracle Readies Web Initiative–Businesses Can Tie Their Databases Directly to the Web", Interactive Age, n207, p. 1, Jan. 30, 1995.

"Hypercard–Based Oracle Card for Macintosh, Windows 3.0", Computergram International, Jan. 1991.

Karpinski, Richard "Web–Based Help Desks Debut", Communications Week, Oct. 1995.

"Next Leaps Aboard the Internet with Webobjects", Computergram International, Aug. 15, 1905.

Clinton et al. "Intranet Tools: Coprorations Seek Internal Web Applications—and Major Vendors are Happy to Help", Informationweek, n552, p14(2), Nov. 6, 1992.

Nash, Kim S. "Web Browser Makers Follw Modular Trend", News, Apr. 29, 1996.

"Netscape & Sun Announce Javascript", Newsbytes, Dec. 4, 1995.

"Oracle Challlenges Netscape with Free Powerbrowser", Computergram International, Dec. 1995.

"Oracle Set Powerbrowser", Computergram International, Dec. 1995.

"Oracle Begins Internet Initiative:Multimedia, Hypertext Development Tools", Seybold Report on Desktop Publishing, v9, n7, p27(1), Mar. 6, 1995.

Menefee, Craig "Oracle Intros Database Products", Newsbytes, Jan. 23, 1995.

"Sybase: Sybase Introduces Powersite Enterprise Web Development Environment", M2 Communications, Dec. 1997.

"ObjectShare IntroducesjKit/Grid; First Offering in a New Line of Advanced Java Components; Extensible New Software Provides Powerful Java Classes for Grids, Tables and More; Advanced Component Automatically Reacts to Data Changes", Buisness Wire, pp. 101, Oct. 1996.

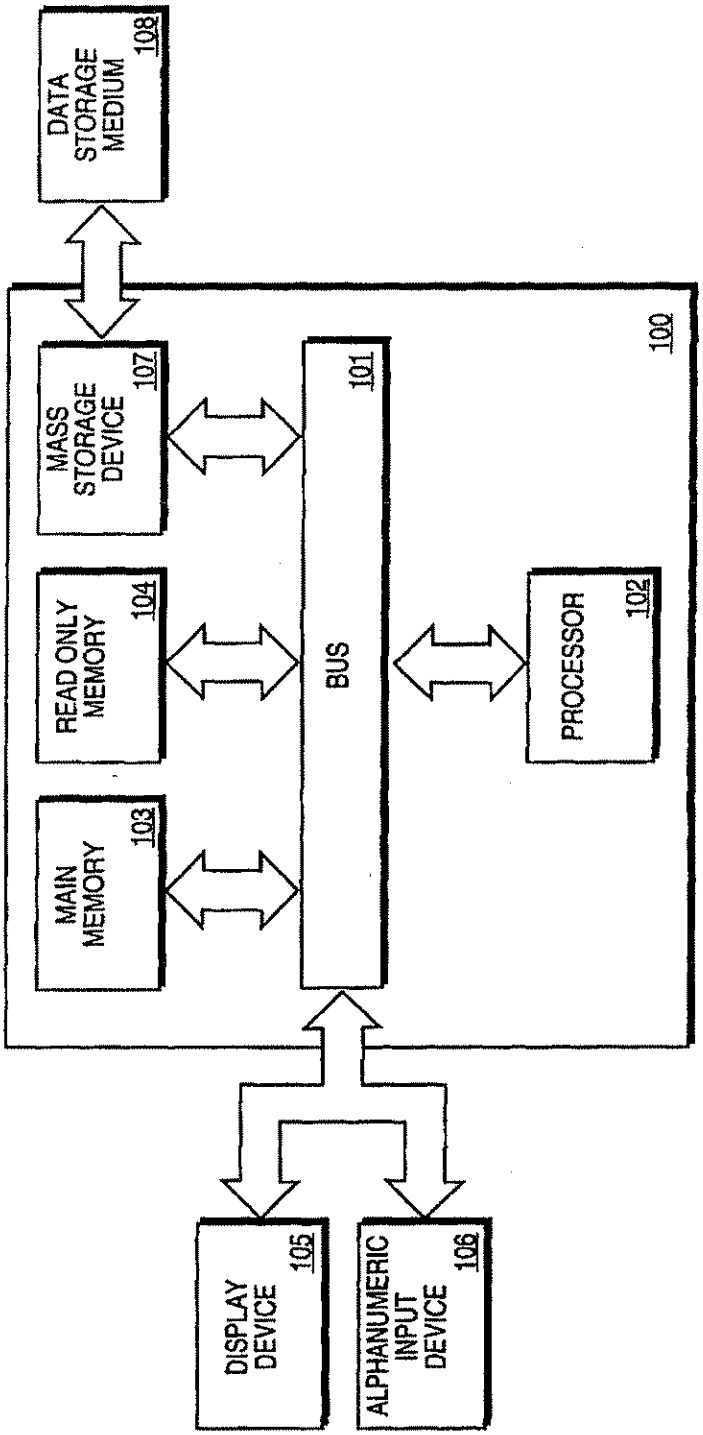

DATA STORAGE MEDIUM
108
100
MASS STORAGE DEVICE
107
101
READ ONLY MEMORY
104
BUS
PROCESSOR
102
MAIN MEMORY
103
DISPLAY DEVICE
105
ALPHANUMERIC INPUT DEVICE
106

FIG. 1

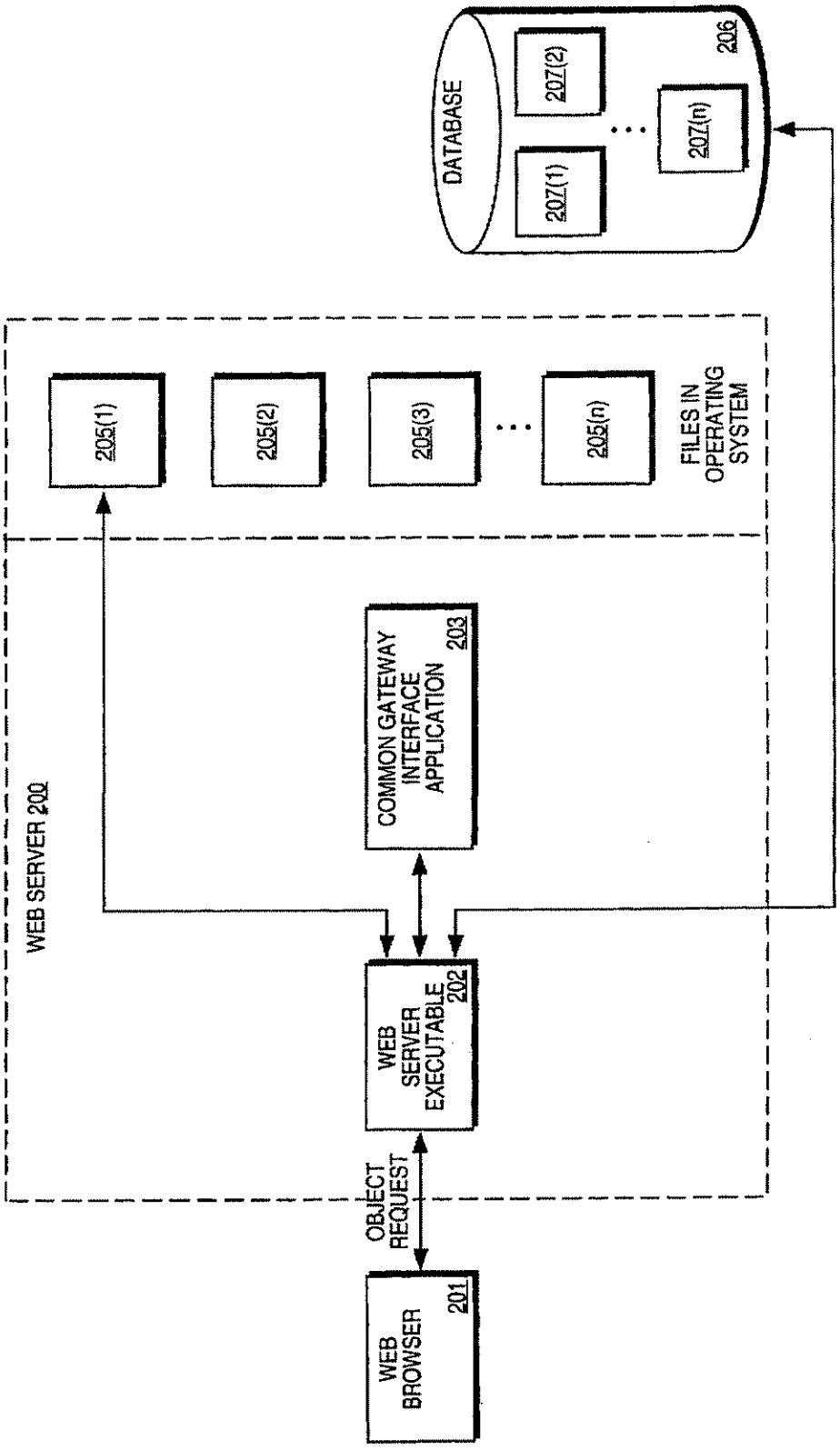


FIG. 2 (PRIOR ART)

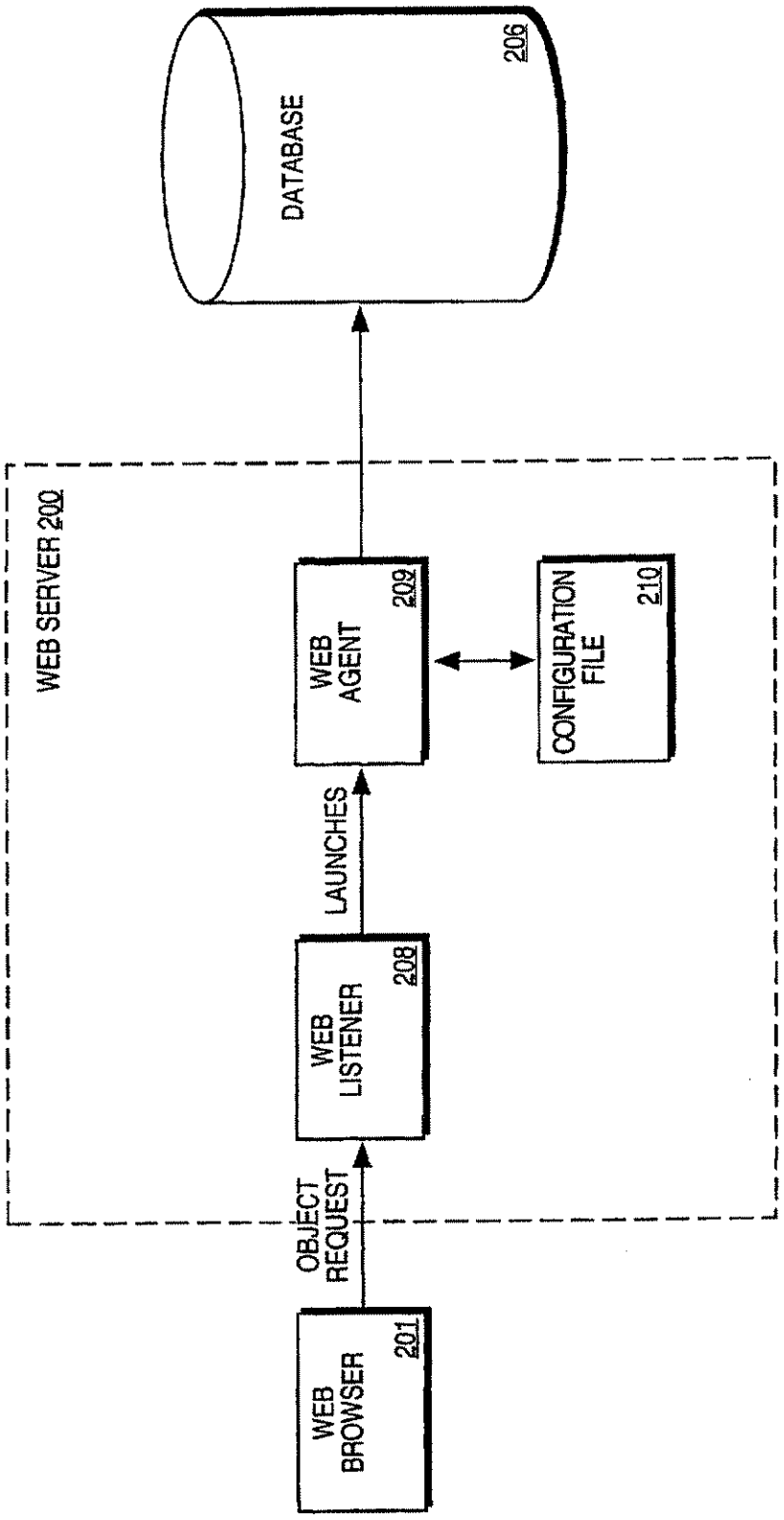

DATABASE
206
WEB SERVER 200
WEB AGENT
209
CONFIGURATION FILE
210
LAUNCHES
WEB LISTENER
208
OBJECT REQUEST
WEB BROWSER
201

FIG. 3

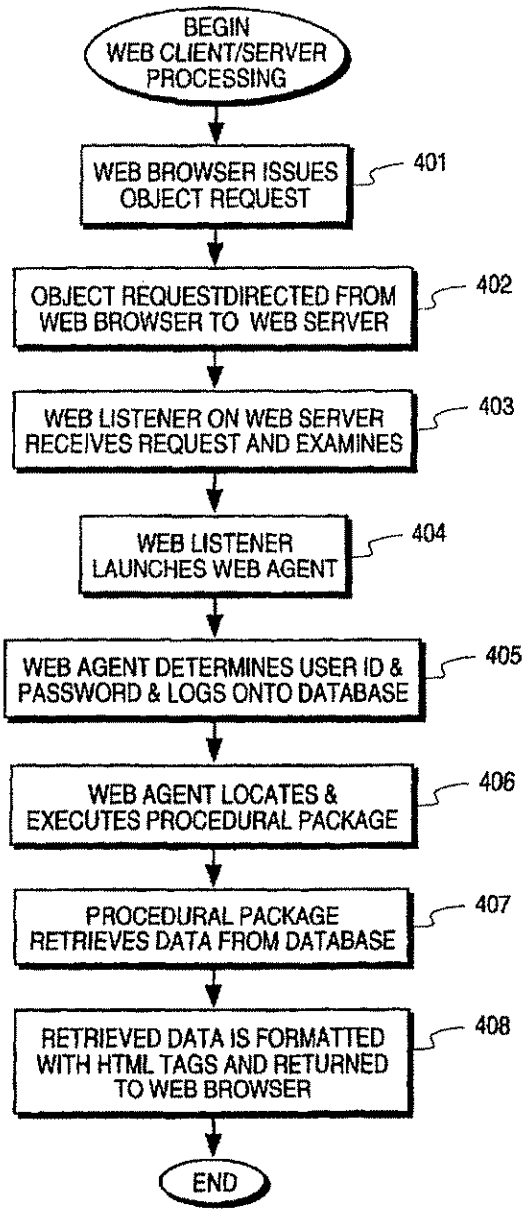

BEGIN WEB CLIENT/SERVER PROCESSING
WEB BROWSER ISSUES OBJECT REQUEST
401
OBJECT REQUESTDIRECTED FROM WEB BROWSER TO WEB SERVER
402
WEB LISTENER ON WEB SERVER RECEIVES REQUEST AND EXAMINES
403
WEB LISTENER LAUNCHES WEB AGENT
404
WEB AGENT DETERMINES USER ID & PASSWORD & LOGS ONTO DATABASE
405
WEB AGENT LOCATES & EXECUTES PROCEDURAL PACKAGE
406
PROCEDURAL PACKAGE RETRIEVES DATA FROM DATABASE
407
RETRIEVED DATA IS FORMATTED WITH HTML TAGS AND RETURNED TO WEB BROWSER
408
END

FIG. 4

METHOD AND APPARATUS FOR GENERATING DYNAMIC WEB PAGES BY INVOKING A PREDEFINED PROCEDURAL PACKAGE STORED IN A DATABASE

FIELD OF THE INVENTION

The present invention relates to the field of Internet server technology. Specifically, the present invention relates to a method and apparatus for generating dynamic Web pages.

DESCRIPTION OF RELATED ART

The World Wide Web ("the Web") represents all the computers on the Internet that offer users access to information and documentation on the Internet via interactive "hypermedia." Hypermedia describes a document in which "hypertext links" are used to connect any combination of graphics, audio, video and text in a non-linear, non-sequential manner. Words, phrases and icons in documents become links that enable a user to jump at will to a new location in a document or to a new document, either on the same computer or on a different computer on the Web.

Hypermedia authors use a special software language known as HyperText Markup Language (HTML) to create these hyperlinks and to create and format these hypermedia documents. These hypermedia documents are generally referred to as "Web pages" and are stored on Web servers as text files, containing the information encoded in HTML. The Web pages can also be stored in databases, in database tables. Examples of currently available Web servers include NETSITE™ from NETSCAPE™, CERN™ from the European Particle Physics Laboratory and NCSA™ from the National Center for Supercomputing Applications (NCSA™) at the University of Illinois, Urbana-Champaign.

Web client machines running "Web browsers" can access Web pages stored on Web servers via a communications protocol known as HyperText Transport Protocol (HTTP). Web browsers are software interfaces that run on Web clients to allow access to Web servers via a simple user interface. A variety of Web browsers are available today, the most popular ones being "NAVIGATOR™" which can be obtained from NETSCAPE™, and "MOSAIC™" which can be obtained from NCSA™.

A Web browser allows a Web client to request a particular hypermedia document from a Web server by specifying a Universal Resource Locator (URL). An URL is a "Web address" that identifies the Web page and its location on the Web. When the appropriate Web server receives the URL, the server locates the document corresponding to the requested URL and, if required, takes action to create HTML output.

Although Web pages were traditionally stored as static files on the Web server operating system, today Web pages can also be generated dynamically using the Common Gateway Interface (CGI). CGI is a standard interface for running external programs on a Web server. It allows Web servers to create dynamic documents when the server receives a request from the Web browser. When the Web server receives a request for a dynamic document, the Web server executes the appropriate CGI script and transmits the output of the execution back to the requesting Web browser. The Web browser does not differentiate between static and dynamic documents. It simply displays the output of the request.

With CGI, it is becoming increasingly common to extract dynamic information from databases as well as from text files. As Web sites continue to grow, storing Web pages as text files in the operating system becomes unwieldy and inefficient. Databases provide significant advantages over traditional file systems for storage of large amounts of information. Database capabilities may also be exploited in other ways to increase the efficiency of generating dynamic Web pages.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a more efficient method and apparatus for generating dynamic Web pages. Specifically, the present invention discloses a method and apparatus for generating dynamic Web pages on a Web server by invoking and executing predefined procedural packages stored in a database.

The claimed invention comprises the steps of receiving an object request on the Web server, activating a Web agent on the Web server based on the object request, the Web agent invoking the predefined procedural package, executing the predefined procedural package to retrieve data from a data repository, and formatting the retrieved data as HTML output.

Other objects, features and advantages of the present invention will be apparent from the accompanying drawings and from the detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates a typical prior art computer system.

FIG. 2 is an illustration of a typical prior art Web server environment.

FIG. 3 illustrates the interaction between the Web browser, the Web listener, the Web agent and the database, as claimed in the preferred embodiment of the present invention.

FIG. 4 illustrates the preferred embodiment of the presently claimed invention, in block diagram form

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention relates to a method and apparatus for generating dynamic Web pages on a Web server. Specifically, the present invention discloses a Web agent for generating dynamic Web pages by invoking and executing procedural packages that are stored in a database. A procedural package, for the purposes of this invention, includes procedural blocks such as Oracle™ Corporation PL/SQL™ blocks. It will, however, be apparent to one of ordinary skill in the art that other types of procedural packages may also be utilized. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent to one of ordinary skill in the art that these specific details need not be used to practice the present invention. In other instances, well-known structures, interfaces, and processes have not been shown in detail in order not to unnecessarily obscure the present invention.

FIG. 1 illustrates a typical computer system **100** in which the present invention operates. The preferred embodiment of the present invention is implemented on a SUN™ Workstation manufactured by SUN MICROSYSTEMS™ of Mountain View, Calif. Alternate embodiments may be implemented on an IBM™ Personal Computer manufactured by IBM Corporation of Armonk, N.Y. or a MACINTOSH™ computer manufactured by APPLE™ Computer, Incorporated of Cupertino, Calif. It will be apparent to those of

ordinary skill in the art that other computer system architectures may also be employed.

In general, such computer systems as illustrated by FIG. 1 comprise a bus **101** for communicating information, a processor **102** coupled with the bus **101** for processing information, main memory **103** coupled with the bus **101** for storing information and instructions for the processor **102**, a read-only memory **104** coupled with the bus **101** for storing static information and instructions for the processor **102**, a display device **105** coupled with the bus **101** for displaying information for a computer user, an alphanumeric input device **106** coupled with the bus **101** for communicating information and command selections to the processor **102**, and a mass storage device **107**, such as a magnetic disk and associated disk drive, coupled with the bus **101** for storing information and instructions. A data storage medium **108** containing digital information is configured to operate with data storage device **107** to allow processor **102** access to the digital information on data storage medium **108** via bus **101**. In addition, a CD-ROM drive (not shown) may also be used for the storage of high resolution images for display on the display device **105**.

Processor **102** may be any of a wide variety of general purpose processors or microprocessors such as the SPARC™ manufactured by SUN MICROSYSTEMS™ the MOTOROLA™ 68040 or POWERPC™ brand microprocessor manufactured by MOTOROLA™ Corporation or the PENTIUM® microprocessor manufactured by INTEL® Corporation. It will be apparent to those of ordinary skill in the art, however, that other varieties of processors may also be used in a particular computer system. Display device **105** may be a liquid crystal device, cathode ray tube (CRT), or other suitable display device. Mass storage device **107** may be a conventional hard disk drive, floppy disk drive, or other magnetic or optical data storage device for reading and writing information stored on a hard disk, a floppy disk, a magnetic tape, or other magnetic or optical data storage medium.

In general, processor **102** retrieves processing instructions and data from a data storage medium **108** using data storage device **107** and downloads this information into random access memory **103** for execution. Processor **102**, then executes an instruction stream from random access memory **103** or read-only memory **104**. Command selections and information input at alphanumeric input device **106** are used to direct the flow of instructions executed by processor **102**. Equivalent input device **106** may also be a pointing device such as a conventional mouse or trackball device. The results of this processing execution are then displayed on display device **105**.

The preferred embodiment of the present invention is implemented as a software module, which may be executed on a computer system such as computer system **100** in a conventional manner. Using well known techniques, the application software of the preferred embodiment is stored on data storage medium **108** and subsequently loaded into and executed within computer system **100**. Once initiated, the software of the preferred embodiment operates in the manner described below.

FIG. **2** illustrates a typical prior art Web server environment, containing conventional objects. An object includes HTML files, GIF or JPEG files, any files supported by Web browsers and Common Gateway Interface (CGI) scripts. Each object has an attribute associated with it that identifies the type of object. For example, a AVI extension on a Web object indicates that the object is a Microsoft™ Video object. Web browser **201** interacts with Web server **200** via Web server executable **202**. Web browser **201** makes an object request from Web server executable **202**. Web server executable **202** locates the object either in a text file on the operating system of Web server **200**, or a table in database **206**, based on the object request URL. If the object is an HTML file, a GIF file or other files supported by Web browsers, Web server executable **202** retrieves and returns object **205(1)** or **207(1)** to Web Browser **201**. If the object requested is a CGI script, however, CGI script **203** will generate dynamically created HTML output and transfer the output to Web server executable **202**, and back to requesting Web browser **201**.

The present invention discloses a method and apparatus for generating dynamic Web pages on a Web server by invoking and executing procedural packages such as ORACLE™ Corporation, PL/SQL™ that are stored in a database. For the purposes of this invention, a "database" is defined as an actual database or any repository that can be made to look like a database to the Web server. Procedural packages provide an object oriented approach to database programming. A package consists of a public header declaration and a private body containing the definitions. The public package header contains function and procedure prototypes as well as other procedural objects such as variables, cursors and exceptions. The private package body contains definitions for objects declared in the package header as well as private functions, procedures, cursors, variables and exceptions, which are only accessible from other functions and procedures in the package.

Each procedure or function may have zero or more parameters and each parameter may be defined as being read-only or writeable by the program unit. One advantage of packaging functions and procedures is an object oriented feature that allows multiple versions of the same function or procedure, with different "signatures." Signatures in this context are parameter lists which can differ in datatype and/or quantity. This feature greatly simplifies the coding effort for developers because they can use the same procedure for different types of data. It also improves code readability and makes code less prone to break when, for example, datatypes in tables are changed. Generating dynamic Web pages using procedural packages also benefits from the simple programming involved in creating procedural packages as opposed to traditional CGI scripts.

A virtual path is the pathname pointing to an HTML document presented as if it were located in the file system rather than in the database. The Web server maps the virtual path to the actual path based on a configuration file. The HTML document may be in a directory on the file system or in the database. Use of this virtual pathname ensures that the actual location of the HTML document will be transparent to a client Web browser that requests the document. The Web browser will be able to use the request an HTML document without knowing whether it is attempting to retrieve the HTML document from the file system or a database. The Web agent takes advantage of two special features available in all Web servers namely the ability to pass extra PATH information after the name of a CGI executable, and the ability to map virtual directories to the physical file system.

FIG. **3** illustrates the interaction between the Web browser **201**, Web listener **208**, Web agent **211** and Database **206**, in the preferred embodiment of the presently claimed invention. For clarity, Database **206** is illustrated as residing at a different location from Web server **200**. It will be apparent to one of ordinary skill in the art, however, that Database **206** may reside either on the same computer as Web Server **200** or on a different computer.

Web browser **201** first makes an object request from Web listener **208** on Web Server **200**. For example, if the object request specifies URL http://nhl.oracle.com/hr/owa/fireemp, the following actions will occur. The first part of the URL, http://nhl.oracle.com identifies the Web server **200**. The object request will therefore be directed from Web browser **201** to Web server **200**. Web listener **208** on Web server **200** receives the object request and examines the rest of the URL. /hr/owa, the first portion of the address, is defined on the Web server as a CGI application with a virtual path pointing to a directory containing Web agent **209**. Web listener **208** launches Web agent **209**. The preferred embodiment of the present invention implements the Web agent as an Oracle Call Interface program. It will be apparent to one of ordinary skill in the art, however, that any interface program that is similar to the Oracle Call Interface may be used to implement Web agent **209**.

After activation, Web agent **209** examines the portion of the URL that precedes "owa." In this example, Web agent **209** finds "/hr" and uses this name to determine which database service to use. A list of various database services will be stored on Web server **200**, in configuration file **210**, each service providing a list of userids and passwords for logging into Database **206**. Once Web agent **209** determines the appropriate userid and password to use for the database service, it logs into Database **206**. Web agent **209** then examines the URL to determine what follows "owa." In the example above, Web agent **209** finds "fireemp" and uses that to locate the predefined procedural package to execute. In the preferred embodiment of the presently claimed invention, "fireemp" is a PL/SQL package stored in an Oracle7™ database. It will be apparent to one of ordinary skill in the art, however, that other types of procedural packages and databases may also be utilized.

Web agent **210** then executes "fireemp" which retrieves data from Database **206**. Although in the preferred embodiment of the claimed invention, the data is retrieved from Database **206**, it will be apparent to one of ordinary skill in the art that the procedural package "fireemp" can also retrieve data from alternate data repositories. The retrieved data is then formatted with HTML tags. The formatting information may be specified in PL/SQL procedure "fireemp." Alternatively, the user can include in PL/SQL procedure "fireemp" a call to a standard set of PL/SQL packages called HTP (HyperText Procedure) and HTF (HyperText Function). HTP and HTP automatically generate HTML tags. The formatted output is then returned to requesting Web browser **201**.

FIG. **4** illustrates the preferred embodiment of the presently claimed invention, in block diagram form, using the same example as above. The Web browser issues an object request having URL http://nhl.oracle.com/hr/owa/fireemp in processing block **401**. The object request is then directed from Web browser to Web server in processing block **402**. The Web listener on the Web server receives the object request and examines the rest of the URL in processing block **403**. /hr/owa, the first portion of the address, is defined on the Web server as a CGI application with a virtual path pointing to a directory containing Web agent. In processing block **404**, the Web listener launches the Web agent "owa" in the directory corresponding to virtual path /hr. After activation, in processing block **405**, the Web agent determines an appropriate userid and password and logs into the database. The Web agent then locates and executes the procedural package "fireemp" in procedural package **406**. The procedural package retrieves data from the database in procedural block **407**. The retrieved data is then formatted with HTML tags and returned to requesting the Web browser in procedural block **408**.

Thus, a method and apparatus for generating dynamic Web pages on a Web server using predefined procedural packages stored in a database is disclosed. These specific arrangements and methods described herein are merely illustrative of the principles of the present invention. Numerous modifications in form and detail may be made by those of ordinary skill in the art without departing from the scope of the present invention. Although this invention has been shown in relation to a particular preferred embodiment, it should not be considered so limited. Rather, the present invention is limited only by the scope of the appended claims.

We claim:

1. A Web agent for generating dynamic Web pages comprising:

means for invoking a predefined procedural package stored in a database;

means for executing the predefined procedural package to retrieve data from a data repository; and

means for formatting the retrieved data to conform with a selected format.

2. A computer-readable medium having stored thereon a plurality of sequences of instructions, the plurality of sequences of instructions including sequences of instructions which, when executed by a processor, cause the processor to perform the steps of:

receiving a request on a Web server;

activating a Web agent on the Web server based on the request, the Web agent invoking a predefined procedural package stored in a database;

executing the predefined procedural package to retrieve data from a data repository; and

formatting the retrieved data to conform with a selected format.

3. A computer-implemented method for generating dynamic Web pages on a Web server using a predefined procedural package stored in a database, the computer-implemented method comprising the steps of:

receiving a request on the Web server;

activating a Web agent on the Web server based on the request, the Web agent invoking the predefined procedural package stored in the database;

executing the predefined procedural package to retrieve data from a data repository; and

formatting the retrieved data to conform with a selected format.

4. The computer-implemented method as described in claim **3** wherein the step of executing the predefined procedural package to retrieve data from the data repository retrieves data from the database.

5. The computer-implemented method as described in claim **4** wherein the step of activating the Web agent further includes the step of identifying a database service.

6. The computer-implemented method as described in claim **5** further including the step of logging into the database using the database service.

7. The computer-implemented method as described in claim **3** wherein the step of formatting the retrieved data is performed by the predefined procedural package.

* * * * *

Exhibit 4

Office Action Summary	Application No. 08/594,686	Applicant(s) BOOKMAN et al.
	Examiner Charles Rones	Group Art Unit 2307

☒ Responsive to communication(s) filed on *Jan 31, 1996*.

☐ This action is **FINAL.**

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

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

Disposition of Claims

☒ Claim(s) *1-7* is/are pending in the application.

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

☐ Claim(s) _____ is/are allowed.

☒ Claim(s) *1-7* is/are rejected.

☐ Claim(s) _____ is/are objected to.

☐ Claims _____ are subject to restriction or election requirement.

Application Papers

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

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

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

☐ The specification is objected to by the Examiner.

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

Priority under 35 U.S.C. § 119

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

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

☐ received.

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

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

*Certified copies not received: _____.

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

Attachment(s)

☒ Notice of References Cited, PTO-892

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

☐ Interview Summary, PTO-413

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

☐ Notice of Informal Patent Application, PTO-152

— SEE OFFICE ACTION ON THE FOLLOWING PAGES —

DETAILED ACTION

Specification

1. The use of the trademarks Microsoft (page 8, line 27), Oracle, & PL/SQL has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless --
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

3. Claims 1-2 & 5-7 are rejected under 35 U.S.C. 102(a) as anticipated by "SUN Java, Netscape Livescript Married as Javascript".

4. As to claims 1-2 & 5-7, "SUN Java, Netscape Livescript Married as Javascript" discloses

a. receiving an object request on the Web server (Netscape Server) wherein it is deemed to be inherent that the server received the request; See para. 1.

b. activating a Web agent on the Web server based on the object request, the Web agent invoking the predefined procedural package (applets); See para 1.

c. executing the predefined procedural package to retrieve data from a data repository (database); and

d. formatting the retrieved data as HTML output wherein the Web pages that are dynamically created are deemed to be in HTML format which is the format that the Web browser software interprets; See para. 1.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-2 & 5-7 are under 35 U.S.C. 103(a) as obvious over "Oracle Rides the Internet with Web", in view of "ORACLE VERSION 7".

7. As to claims 1 & 6-7, "Oracle Rides the Internet with Web" discloses

a. receiving an object request on the Web server wherein it is deemed to be inherent that the server received the request which resulted in a returned document to the user; See para. 1.

b. activating a Web agent on the Web server based on the object request, the Web agent invoking the predefined procedural package; See para 1.

c. formatting the retrieved data as HTML output wherein the Web pages that are dynamically created are deemed to be in HTML format; See para. 1.

8. "Oracle Rides the Internet with Web" discloses the claimed invention except executing a predefined procedural package to retrieve data from a data repository. "ORACLE VERSION 7" teaches that it is known to execute a predefined procedural package to retrieve data from a data repository. It would have been obvious to one having ordinary skill in the art at the time the invention was made to only execute a predefined procedural package to retrieve data from a data repository as taught by "ORACLE VERSION 7", since "ORACLE VERSION 7" states at the last paragraph that such a modification would allow the application to remain completely user programmable which can be explicitly invoked or triggered by an event.

9. Claims 3-4 are rejected under 35 U.S.C. 103(a) as anticipated by "Oracle Rides the Internet with Web", in view of "ORACLE VERSION 7" as applied to claim 2 above, in further view of Marshall et al. ("Industry Lines Up For Java--Vendors Set Stage for Dynamic").

10. As to claim 3-4, the modified in invention of "Oracle Rides the Internet with Web" discloses the claimed invention except for identifying a database service and logging into the database using the database service. Marshall et al. discloses that it is known to identify a database service and logging into the database using the database service. It would have been obvious to one having ordinary skill in the art at the time the invention was made to identify a database service and logging into the database using the database service as taught by Marshall et al., since Marshall et al. states on page 2, paras. 2-3, that such a modification would allow corporate data to present Web pages to individuals logging in on to commercial services ranging from banking to transportation and that it can be helpful in making sure that data entered from the outside world into the corporate databases are complete and properly formatted.

Conclusion

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Charles Rones whose telephone number is (703) 306-3030. The examiner can normally be reached on Monday through Thursday from 8 a.m. to 4 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thomas Black, can be reached on (703) 305-9707. The fax phone number for this Group is (703) 305-9731.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 308-9600.

Rones

April 14, 1997

THOMAS G. BLACK
SUPERVISORY PATENT EXAMINER
GROUP 2300

Notice of References Cited

Application No.	Applicant(s)		
08/594,686	BOOKMAN et al.		
Examiner		Group Art Unit	Page 1 of 5
Charles Rones		2307	

U.S. PATENT DOCUMENTS

	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
A	5,572,643	11-5-96	Judson	395	793
B	5,530,852	6-25-96	Meske, Jr. et al.	395	610
C	5,537,586	7-16-96	Amram et al.	395	603
D					
E					
F					
G					
H					
I					
J					
K					
L					
M					

FOREIGN PATENT DOCUMENTS

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

NON-PATENT DOCUMENTS

	DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
U	"ORACLE VERSION 7", Computergram International.	6-17-92
V	"ORACLE Rides the Internet with Web", Computergram International.	2-10-95
W	"Sun Java, Netscape Livescript Married as Javascript", Computergram International.	12-95
X	Marshall et al. "Industry Lines Up for Java -- Vendors Set Stage for Dynamic", Commications Week.	12-95

Notice of References Cited	Application No. 08/594,686	Applicant(s) BOOKMAN et al.	
	Examiner Charles Rones	Group Art Unit 2307	Page 2 of 5

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
	A					
	B					
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

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

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	U	"ORACLE Unveils ORACLE Websystem Internet Line", Computergram International.	11-95
	V	Marshall, Martin "Sybase Sites to Get Web Links", Communciations Week.	11-95
	W	Karpinski, Richard "ORACLE Readies Web Initiative-Businesses Can Tie Their Databases Directly to the Web", Interactive Age, n207, pg.1.	1-30-95
	X	"Hypercard-Based ORACLE Card for Macintosh, Windows 3.0", Computergram International.	1-91

Notice of References Cited

Application No.	Applicant(s)		
08/594,686	BOOKMAN et al.		
Examiner		Group Art Unit	Page 3 of 5
Charles Rones		2307	

U.S. PATENT DOCUMENTS

	DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
A					
B					
C					
D					
E					
F					
G					
H					
I					
J					
K					
L					
M					

FOREIGN PATENT DOCUMENTS

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

NON-PATENT DOCUMENTS

	DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
U	Karpinski, Richard "Web-Based Help Desks Debut", Communications Week.	10-95
V	"Next Leaps Aboard the Internet with Webobjects", Computergram International.	8-15-05
W	Clinton et al. "Intranet Tools: Coprorations Seek Internal Web Applications - and Major Vendors are Happy to Help", Informationweek, n552, p14(2).	11-6-95
X	Nash, Kim S. "Web Browser Makers Follow Modular Trend", News.	4-29-96

Notice of References Cited

Application No.	Applicant(s)		
08/594,686	BOOKMAN et al.		
Examiner		Group Art Unit	Page 4 of 5
Charles Rones		2307	

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
	A					
	B					
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

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

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	U	"Netscape & Sun Announce Javascript", Newsbytes.	12-4-95
	V	"ORACLE Challlenges Netscape with Free Powerbrowser", Computergram International.	12-95
	W	"ORACLE Set Powerbrowser", Computergram International.	12-5-95
	X	"ORACLE Begins Internet Initiative:Multimedia, Hypertext Development Tools", Seybold Report on Desktop Publishing, v9, n7, p27(1).	3-6-95

Notice of References Cited	Application No. 08/594,686	Applicant(s) BOOKMAN et al.	
	Examiner Charles Rones	Group Art Unit 2307	Page 5 of 5

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
	A					
	B					
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

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

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	U	Menefee, Craig "ORACLE Intros Database Products", Newsbytes.	1-23-95
	V		
	W		
	X		

Exhibit 5

Notice of Allowability	Application No. 08/594,686	Applicant(s) BOOKMAN et al.
	Examiner Charles Rones	Group Art Unit 2771

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance and Issue Fee Due or other appropriate communication will be mailed in due course.

☒ This communication is responsive to *interview on January 26, 1997*.

☒ The allowed claim(s) is/are *1-7*.

☒ The drawings filed on *Jan 31, 1996* are acceptable.

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

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

☐ received.

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

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

*Certified copies not received: _____.

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

A SHORTENED STATUTORY PERIOD FOR RESPONSE to comply with the requirements noted below is set to EXPIRE **THREE MONTHS** FROM THE "DATE MAILED" of this Office action. Failure to timely comply will result in ABANDONMENT of this application. Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

☐ Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL APPLICATION, PTO-152, which discloses that the oath or declaration is deficient. A SUBSTITUTE OATH OR DECLARATION IS REQUIRED.

☐ Applicant MUST submit NEW FORMAL DRAWINGS

☐ because the originally filed drawings were declared by applicant to be informal.

☐ including changes required by the Notice of Draftsperson's Patent Drawing Review, PTO-948, attached hereto or to Paper No. _____.

☐ including changes required by the proposed drawing correction filed on _____, which has been approved by the examiner.

☒ including changes required by the attached Examiner's Amendment/Comment.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the reverse side of the drawings. The drawings should be filed as a separate paper with a transmittal lettter addressed to the Official Draftsperson.

☐ Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Any response to this letter should include, in the upper right hand corner, the APPLICATION NUMBER (SERIES CODE/SERIAL NUMBER). If applicant has received a Notice of Allowance and Issue Fee Due, the ISSUE BATCH NUMBER and DATE of the NOTICE OF ALLOWANCE should also be included.

Attachment(s)

☒ Notice of References Cited, PTO-892

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

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

☐ Notice of Informal Patent Application, PTO-152

☒ Interview Summary, PTO-413

☒ Examiner's Amendment/Comment

☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material

☐ Examiner's Statement of Reasons for Allowance

THOMAS G. BLACK
SUPERVISORY PATENT EXAMINER
GROUP 2700

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Charmani Green on January 27, 1998.

2. The application has been amended as follows:

The title has been deleted and --METHOD AND APPARATUS FOR GENERATING DYNAMIC WEB PAGES BY INVOKING A PREDEFINED PROCEDURAL PACKAGE STORED IN A DATABASE--.

Page 5, line 2 which states "illustrates a typical computer system" has been deleted and --illustrates a typical prior art computer system-- has been inserted thereto.

3. The following changes to the drawings have been approved by the examiner and agreed upon by applicant: Figure 1 should contain the --Prior Art-- label. In order to avoid abandonment of the application, applicant must make these above agreed upon drawing changes.

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Charles Rones whose telephone number is (703) 306-3030. The examiner can normally be reached on Monday through Thursday from 8 a.m. to 4 p.m. EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thomas Black, can be reached on (703) 305-9707.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 308-9051, (for formal communications intended for entry)

Or:

(703) 305-9731 (for informal or draft communications, please label "PROPOSED" or "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA., Sixth Floor (Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 308-9600.

CLR

Rones

January 27, 1998

Thomas G. Black
THOMAS G. BLACK
SUPERVISORY PATENT EXAMINER
GROUP 2700

Notice of References Cited

Application No.	Applicant(s)
08/594,686	BOOKMAN et al.
Examiner	**Group Art Unit**
Charles Rones	2771

Page 1 of 1

U.S. PATENT DOCUMENTS

*		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
x	A	5,701,451	12-23-97	Rogers et al.	707	104
x	B	5,710,918	1-20-98	Lagarde et al.	395	680
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

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

NON-PATENT DOCUMENTS

*		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
x	U	"Sybase: Sybase Introduces Powersite Enterprise Web Development Environment", M2 Communications.	12-97
x	V	"ObjectShare IntroducesjKit/Grid; First Offering in a New Line of Advanced Java Components; Extensible New Software Provides Powerful Java Classes for Grids, Tables and More; Advanced Component Automatically Reacts to Data Changes", Buisness Wire, pp. 101.	10-96
	W		
	X		

CR 1-27-98

*A copy of this reference is not being furnished with this Office action. (See Manual of Patent Examining Procedure, Section 707.05(a).)

Exhibit 6

USPTO Patent Full-Text and Image Database

Home | Quick | Advanced | Pat Num | Help
Bottom | View Cart

Searching US Patent Collection...

Results of Search in US Patent Collection db for:
(AN/Oracle AND LREP/"Blakely, Sokoloff, Taylor"): 42 patents.
Hits 1 through 42 out of 42

Jump To

Refine Search AN/Oracle AND LREP/"Blakely, Sokoloff, Taylor"

	PAT. NO.	Title
1	6,160,549	Method and apparatus for generating reports using declarative tools
2	6,157,941	Architecture for client-server communication over a communication link
3	6,138,253	Method and apparatus for reporting errors in a computer system
4	6,126,328	Controlled execution of partitioned code
5	6,055,577	System for granting bandwidth for real time processes and assigning bandwidth for non-real time processes while being forced to periodically re-arbitrate for new assigned bandwidth
6	6,044,216	Method and apparatus for implementing cursor variables for accessing data from database
7	6,041,423	Method and apparatus for using undo/redo logging to perform asynchronous updates of parity and data pages in a redundant array data storage environment
8	6,041,384	Method for managing shared resources in a multiprocessing computer system
9	5,987,422	Method for executing a procedure that requires input from a role
10	5,915,086	Hierarchical protection of seed data
11	5,900,024	Method for processing type-ahead input and operation-abort input
12	5,890,127	Method and apparatus for performing a loop reset in a process modeling system
13	5,878,220	Method and apparatus for storing and transferring data on a network
14	5,872,981	Method for managing termination of a lock-holding process using a waiting lock
15	5,857,180	Method and apparatus for implementing parallel operations in a database management system
16	5,850,547	Method and apparatus for parallel processing aggregates using intermediate aggregate values
17	5,850,517	Communication link for client-server having agent which sends plurality of requests independent of client and receives information from the server independent of the

server

18 5,845,296 T Method and apparatus for implementing segmented arrays in a database
19 5,842,197 T Selecting a qualified data repository to create an index
20 5,835,762 T Method and apparatus for processing electronic mail in parallel
21 5,826,085 T Object oriented computer interface supporting interactive networked applications
22 5,809,495 T Method for obtaining information regarding the current activity of a database management system from a viritual table in a memory of the database management system
23 5,809,266 T Method and apparatus for generating reports using declarative tools
24 5,806,075 T Method and apparatus for peer-to-peer data replication
25 5,806,074 T Configurable conflict resolution in a computer implemented distributed database
26 5,805,804 T Method and apparatus for scalable, high bandwidth storage retrieval and transportation of multimedia data on a network
27 5,802,523 T Method and apparatus for reducing the memory required to store bind variable descriptors in a database
28 5,794,241 T Method and apparatus for dynamically disabling and enabling table locking for a database
29 5,787,410 T Method and apparatus for storing and retrieving data in multiple languages simultaneously using a fully-populated sub-table
30 5,768,589 T Method and apparatus for executing stored procedures in a foreign database management system
31 5,761,673 T Method and apparatus for generating dynamic web pages by invoking a predefined procedural package stored in a database
32 5,742,706 T Method and apparatus for comparison of data strings
33 5,737,601 T Method and apparatus for peer-to-peer data replication including handling exceptional occurrences
34 5,724,556 T Method and apparatus for defining and configuring modules of data objects and programs in a distributed computer system
35 5,721,848 T Method and apparatus for building efficient and flexible geometry management widget classes
36 5,655,117 T Method and apparatus for indexing multimedia information streams
37 5,623,595 T Method and apparatus for transparent, real time reconstruction of corrupted data in a redundant array data storage system
38 5,524,205 T Methods and apparatus for optimizing undo log usage
39 5,504,897 T Method and apparatus for processing electronic mail in parallel
40 5,499,367 T System for database integrity with multiple logs assigned to client subsets
41 5,485,608 T Methods and apparatus for updating information in a computer system using logs and state identifiers
42 5,412,804 T Extending the semantics of the outer join operator for un-nesting queries to a data base

Exhibit 7

From: Willette, Timothy M **On Behalf Of** Bosy, George S
Sent: Thursday, September 13, 2007 12:06 PM
To: Artuz, Robert J.; Herhold, Ted T.
Cc: tfokas@parallelnetworks.com; Roper, Harry J; Trybus, Steven R; Barlow, Aaron A; Patras, Patrick L; Bennett, David R; Margolis, Paul D; Bradford, Benjamin J; Terry, Angela M; Johnson, Emily C.; Erickson, Amy H
Subject: Draft amended answer and counterclaim

Transmitted is a draft amended answer and counterclaim. Please let me know whether Oracle will consent.

George S. Bosy
Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611-7603
(312) 923-8305
(312) 923-8405 (fax)

9/13/07 DRAFT

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

ORACLE CORPORATION, and ORACLE U.S.A., INC.,	:	
Plaintiff,	:	
v.	:	C. A. No. 1:06-cv-00414 (SLR)
PARALLEL NETWORKS, LLC.	:	
Defendant.	:	

DEFENDANT'S FIRST AMENDED ANSWER AND COUNTERCLAIM

Defendant epicRealm Licensing, LP ("epicRealm") amends its answer and counterclaim to Plaintiffs Oracle Corporation and Oracle U.S.A., Inc.'s (collectively "Oracle's") Complaint for Declaratory Judgment ("Complaint") to substitute defendant and counterclaimant Parallel Networks, LLC ("Parallel Networks") for epicRealm. In August 2007, epicRealm assigned to Parallel Networks all right, title and interest in United States Patent No. 5,894,554 (the "'554 patent") and United States Patent No. 6,415,335 (the "'335 patent") (collectively, the "Parallel Networks Patents"), including the right to enforce the Parallel Networks Patents and to recover for past, current and future infringement. Parallel Networks is a limited liability company organized under the laws of the State of Texas. Defendant and counterclaimant Parallel Networks now respectfully files this First Amended Answer and Counterclaim ("Amended Answer").

THE PARTIES

1. Parallel Networks lacks knowledge or information sufficient to form a belief as to the truth of the allegations of paragraph 1 of Oracle's Complaint.

2. Parallel Networks lacks knowledge or information sufficient to form a belief as to the truth of the allegations of paragraph 1 of Oracle's Complaint.

3. Parallel Networks admits that epicRealm Licensing, LP is a Delaware limited partnership. Parallel Networks denies that epicRealm Licensing, LP's principal place of business is at 558 S. Central Expressway, Richardson, Texas 75080-6126. epicRealm Licensing, LP's principal place of business is in 1700 Pacific Avenue, Suite 2320, Dallas, Texas 75201. Parallel Networks, LLC is a limited liability company organized under the laws of the State of Texas with its principal place of business at 1700 Pacific Avenue, Suite 2320, Dallas, Texas 75201.

JURISDICTION AND VENUE

4. Parallel Networks admits the allegations of paragraph 4 of Oracle's Complaint.

5. Parallel Networks admits the allegations of paragraph 5 of Oracle's Complaint.

6. Parallel Networks admits the allegations of paragraph 6 of Oracle's Complaint.

ALLEGATIONS RELATED TO ALL COUNTS

7. Parallel Networks admits the allegations of paragraph 7 of Oracle's Complaint.

8. Parallel Networks admits the allegations of paragraph 8 of Oracle's Complaint.

9. Parallel Networks denies that epicRealm Licensing, LP is the present owner and assignee of the '554 and '335 patents. Parallel Networks states that it is the current owner and assignee of the '554 and '335 patents and has the right to enforce the Parallel Networks Patents against Oracle and to recover for past, current and future infringement. An assignment reflecting this fact has been recorded in the United States Patent and Trademark Office and is attached as Exhibit C.

10. Parallel Networks admits the allegations of paragraph 10 of Oracle's Complaint.

11. Parallel Networks admits the allegations of paragraph 11 of Oracle's Complaint.

12. Parallel Networks admits the allegations of paragraph 12 of Oracle's Complaint.

13. Parallel Networks admits the allegations of paragraph 13 of Oracle's Complaint.

14. Parallel Networks admits the allegations of paragraph 14 of Oracle's Complaint.

15. Parallel Networks admits the allegations of paragraph 15 of Oracle's Complaint.

16. Parallel Networks lacks knowledge or information sufficient to form a belief as to the truth of the allegations of paragraph 16 of Oracle's Complaint.

17. Parallel Networks lacks knowledge or information sufficient to form a belief as to the truth of the allegations of paragraph 17 of Oracle's Complaint.

18. Parallel Networks admits the allegations of paragraph 18 of Oracle's Complaint.

19. Parallel Networks lacks knowledge or information sufficient to form a belief as to the truth of the allegations of paragraph 19 of Oracle's Complaint.

20. Parallel Networks lacks knowledge or information sufficient to form a belief as to the truth of the allegations of paragraph 20 of Oracle's Complaint.

21. Parallel Networks lacks knowledge or information sufficient to form a belief as to the truth of the allegations of paragraph 21 of Oracle's Complaint.

22. Parallel Networks admits the allegations of paragraph 22 of Oracle's Complaint.

23. Parallel Networks admits the allegations of paragraph 23 of Oracle's Complaint.

24. Parallel Networks admits the allegations of paragraph 24 of Oracle's Complaint.

25. Parallel Networks denies that discovery was requested specifically relating to "Safelite's use of the Oracle Software."

26. Parallel Networks admits the allegations of paragraph 26 of Oracle's Complaint.

27. Parallel Networks admits the allegations of paragraph 27 of Oracle's Complaint.

28. Parallel Networks lacks knowledge or information sufficient to form a belief as to the truth of the allegations of paragraph 28 of Oracle's Complaint.

FIRST COUNT: NONINFRINGEMENT

29. Parallel Networks reasserts its answers to paragraphs 1-28 of Oracle's Complaint.

30. Parallel Networks denies that discovery was requested specifically relating to Safelite's use of Oracle Software. Parallel Networks admits that an actual controversy exists between Oracle and Parallel Networks.

31. Parallel Networks denies the allegations of paragraph 31 of Oracle's Complaint.

32. Parallel Networks admits that an actual controversy exists between Oracle and Parallel Networks. Parallel Networks denies Oracle's claims of noninfringement.

SECOND COUNT: INVALIDITY

33. Parallel Networks reasserts its answers to paragraphs 1-32 of Oracle's Complaint.

34. Parallel Networks denies that discovery was requested specifically relating to Safelite's use of Oracle Software. Parallel Networks admits that an actual controversy exists between Oracle and Parallel Networks.

35. Parallel Networks denies the allegations of paragraph 35 of Oracle's Complaint.

36. Parallel Networks admits that an actual controversy exists between Oracle and Parallel Networks. Parallel Networks denies Oracle's claims of invalidity.

37. In response to Oracle's Prayer, Parallel Networks denies that Oracle is entitled to relief of any kind.

COUNTERCLAIM

Parallel Networks counterclaims against Oracle for patent infringement and alleges the following:

38. Parallel Networks reasserts its answers to paragraphs 1-36 of Oracle's Complaint.

39. The Court has jurisdiction over this Counterclaim pursuant to 28 U.S.C. §§ 1331 and 1338.

40. The Court has personal jurisdiction over Oracle by virtue of its commencing this action against Parallel Networks in this Court.

41. Venue is proper in this judicial district pursuant to 28 U.S.C. §§ 1391(b), (c) and 1400(b).

PATENT INFRINGEMENT

42. On April 13, 1999, and July 2, 2002, United States Patent Nos. 5,894,554 and 6,415,335 B1, which are collectively referred to as the "Parallel Networks Patents," duly and legally issued. These two patents concern, among other things, systems and methods for managing dynamic Web page generation requests. Copies of the Parallel Networks Patents are attached hereto as Exhibits "A" and "B" and made a part hereof.

43. Parallel Networks is the owner of the Parallel Networks Patents and has the right to enforce those patents with respect to Oracle.

44. On information and belief, Oracle makes, uses, offers for sale, sells, imports, and/or induces the use of systems and methods for managing dynamic Web page generation requests within the scope of one or more of the claims of the Parallel Networks Patents. As a result, Oracle has been and still is infringing one or more of the claims of the Parallel Networks Patents as defined by 35 U.S.C. § 271. Parallel Networks has suffered damage by reason of Oracle's infringement and will continue to suffer additional damage until this Court enjoins the infringing conduct.

45. To the extent that Oracle has continued or does continue its infringing activities after receiving notice of the Parallel Networks Patents, such infringement is willful, entitling Parallel Networks to the recovery of increased damages under 35 U.S.C. § 284.

46. This is an "exceptional case" justifying an award of attorneys' fees and costs to Parallel Networks pursuant to 35 U.S.C. § 285.

47. Parallel Networks believes that Oracle will continue to infringe the Parallel Networks Patents unless enjoined by this Court. Such infringing activity causes Parallel Networks irreparable harm and will continue to cause such harm without the issuance of an injunction.

JURY DEMAND

48. Parallel Networks requests trial by jury pursuant to Federal Rule of Civil Procedure 38.

PRAYER FOR RELIEF

Parallel Networks requests that the Court find in its favor and against Oracle and that the Court grant the following relief:

a. Judgment that one or more of the claims of the Parallel Networks Patents have been infringed, either literally and/or under the doctrine of equivalents, by Oracle;

b. Judgment in favor of Parallel Networks for the full amount of its actual damages caused by Oracle's infringing activities, including an assessment of interest and costs;

c. Judgment for increased damages for willful infringement pursuant to 35 U.S.C. § 284;

d. Judgment that this is an "exceptional case" and awarding Parallel Networks its reasonable attorneys' fees and costs pursuant to 35 U.S.C. § 285;

e. That Oracle be permanently enjoined from further activity or conduct that infringes the claims of the Parallel Networks Patents; and

f. That the Court award Parallel Networks such other and further relief as is just and proper under the circumstances.

9/13/07 DRAFT

POTTER ANDERSON & CORROON LLP

By: */s/ David E. Moore*
Richard L. Horwitz (#2246)
David E. Moore (#3983)
Hercules Plaza, 6th Floor
1313 N. Market Street
Wilmington, DE 19899
Tel: (302) 984-6000
rhorwitz@potteranderson.com
dmoore@potteranderson.com

Attorneys for Defendant
Parallel Networks, LLC

OF COUNSEL:

Harry J. Roper
George S. Bosy
Aaron A. Barlow
Patrick L. Patras
David R. Bennett
Paul D. Margolis
Benjamin J. Bradford
JENNER & BLOCK
330 N. Wabash Avenue
Chicago, IL 60611-7603
Tel: (312) 923-8305

Dated: September ____, 2007

EXHIBIT C

TO
9/13/07 DRAFT

DEFENDANT'S FIRST AMENDED ANSWER AND COUNTERCLAIM

TO:B. K. DRINKWATER, C/O BAKER BOTTS L.L.P. COMPANY:2001 ROSS AVENUE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNDER SECRETARY OF COMMERCE FOR INTELLECTUAL PROPERTY AND
DIRECTOR OF THE UNITED STATES PATENT AND TRADEMARK OFFICE

SEPTEMBER 10, 2007

PTAS

B. K. DRINKWATER, C/O BAKER BOTTS L.L.P.
2001 ROSS AVENUE
SUITE 600
DALLAS, TX 75201

500350872A

UNITED STATES PATENT AND TRADEMARK OFFICE
NOTICE OF RECORDATION OF ASSIGNMENT DOCUMENT

THE ENCLOSED DOCUMENT HAS BEEN RECORDED BY THE ASSIGNMENT DIVISION OF THE U.S. PATENT AND TRADEMARK OFFICE. A COMPLETE MICROFILM COPY IS AVAILABLE AT THE ASSIGNMENT SEARCH ROOM ON THE REEL AND FRAME NUMBER REFERENCED BELOW.

PLEASE REVIEW ALL INFORMATION CONTAINED ON THIS NOTICE. THE INFORMATION CONTAINED ON THIS RECORDATION NOTICE REFLECTS THE DATA PRESENT IN THE PATENT AND TRADEMARK ASSIGNMENT SYSTEM. IF YOU SHOULD FIND ANY ERRORS OR HAVE QUESTIONS CONCERNING THIS NOTICE, YOU MAY CONTACT THE EMPLOYEE WHOSE NAME APPEARS ON THIS NOTICE AT 571-272-3350. PLEASE SEND REQUEST FOR CORRECTION TO: U.S. PATENT AND TRADEMARK OFFICE, MAIL STOP: ASSIGNMENT SERVICES BRANCH, P.O. BOX 1450, ALEXANDRIA, VA 22313.

RECORDATION DATE: 09/10/2007 REEL/FRAME: 019795/0962
NUMBER OF PAGES: 6

BRIEF: MERGER (SEE DOCUMENT FOR DETAILS).
DOCKET NUMBER: 066241.0146

ASSIGNOR:
EPICREALM LICENSING, LLC DOC DATE: 12/29/2005

ASSIGNEE:
EPICREALM LICENSING, LP
1700 PACIFIC AVENUE
SUITE 2320
DALLAS, TEXAS 75201

SERIAL NUMBER: 09590760 FILING DATE: 06/08/2000
PATENT NUMBER: ISSUE DATE:
TITLE: METHOD AND APPARATUS FOR CONTENT SYNCHRONIZATION

SERIAL NUMBER: 09592486 FILING DATE: 06/08/2000
PATENT NUMBER: ISSUE DATE:

TO:B. K. DRINKWATER, C/O BAKER BOTTS L.L.P. COMPANY:2001 ROSS AVENUE

TITLE: METHOD AND APPARATUS FOR DYNAMIC DATA FLOW CONTROL USING PRIORITIZATION OF DATA REQUESTS

019795/0962 PAGE 2

SERIAL NUMBER: 09731365 FILING DATE: 12/06/2000
PATENT NUMBER: 7113935 ISSUE DATE: 09/26/2006
TITLE: METHOD AND SYSTEM FOR ADAPTIVE PREFETCHING

SERIAL NUMBER: 09640478 FILING DATE: 08/16/2000
PATENT NUMBER: ISSUE DATE:
TITLE: METHOD AND SYSTEM FOR UNIFORM RESOURCE LOCATOR TRANSFORMATION

SERIAL NUMBER: 09759392 FILING DATE: 01/12/2001
PATENT NUMBER: 7035911 ISSUE DATE: 04/25/2006
TITLE: METHOD AND SYSTEM FOR COMMUNITY DATA CACHING

SERIAL NUMBER: 09759406 FILING DATE: 01/12/2001
PATENT NUMBER: 7188145 ISSUE DATE: 03/06/2007
TITLE: METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

SERIAL NUMBER: 08636477 FILING DATE: 04/23/1996
PATENT NUMBER: 5894554 ISSUE DATE: 04/13/1999
TITLE: SYSTEM FOR MANAGING DYNAMIC WEB PAGE GENERATION REQUESTS BY INTERCEPTING REQUEST AT WEB SERVER AND ROUTING TO PAGE SERVER THEREBY RELEASING WEB SERVER TO PROCESS OTHER REQUESTS

SERIAL NUMBER: 09234048 FILING DATE: 01/19/1999
PATENT NUMBER: 6415335 ISSUE DATE: 07/02/2002
TITLE: SYSTEM AND METHOD FOR MANAGING DYNAMIC WEB PAGE GENERATION REQUESTS

SERIAL NUMBER: 09336186 FILING DATE: 06/18/1999
PATENT NUMBER: 6446111 ISSUE DATE: 09/03/2002
TITLE: METHOD AND APPARATUS FOR CLIENT-SERVER COMMUNICATION USING A LIMITED CAPABILITY CLIENT OVER A LOW-SPEED COMMUNICATIONS LINK

ASSIGNMENT SERVICES BRANCH
PUBLIC RECORDS DIVISION

TO:B. K. DRINKWATER, C/O BAKER BOTTS L.L.P. COMPANY:2001 ROSS AVENUE

PATENT ASSIGNMENT

Electronic Version v1.1
Stylesheet Version v1.1

09/10/2007
500350872

SUBMISSION TYPE:	NEW ASSIGNMENT
NATURE OF CONVEYANCE:	MERGER
EFFECTIVE DATE:	12/29/2005

CONVEYING PARTY DATA

Name	Execution Date
EPICREALM LICENSING, LLC	12/29/2005

RECEIVING PARTY DATA

Name:	EPICREALM LICENSING, LP
Street Address:	1700 Pacific Avenue
Internal Address:	Suite 2320
City:	Dallas
State/Country:	TEXAS
Postal Code:	75201

PROPERTY NUMBERS Total: 9

Property Type	Number
Patent Number:	5894554
Patent Number:	6415335
Patent Number:	6446111
Application Number:	09590760
Application Number:	09592486
Application Number:	09731365
Application Number:	09640478
Application Number:	09759392
Application Number:	09759406

CH $360.00 5894554

CORRESPONDENCE DATA

Fax Number: (214)661-4691
Correspondence will be sent via US Mail when the fax attempt is unsuccessful.
Phone: 214.953.6500

Email:	b.k.drinkwater@bakerbotts.com
Correspondent Name:	B. K. Drinkwater, c/o Baker Botts L.L.P.
Address Line 1:	2001 Ross Avenue
Address Line 2:	Suite 600
Address Line 4:	Dallas, TEXAS 75201

ATTORNEY DOCKET NUMBER:	066241.0146
NAME OF SUBMITTER:	B. K. Drinkwater

Total Attachments: 4
source=EpicRealm Licensing LP merger#page1.tif
source=EpicRealm Licensing LP merger#page2.tif
source=EpicRealm Licensing LP merger#page3.tif
source=EpicRealm Licensing LP merger#page4.tif

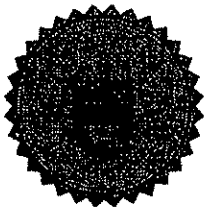
Delaware

The First State

I, HARRIET SMITH WINDSOR, SECRETARY OF STATE OF THE STATE OF DELAWARE, DO HEREBY CERTIFY THE ATTACHED IS A TRUE AND CORRECT COPY OF THE CERTIFICATE OF MERGER, WHICH MERGES:

"EPICREALM LICENSING, LLC", A TEXAS LIMITED LIABILITY COMPANY,

WITH AND INTO "EPICREALM LICENSING, LP" UNDER THE NAME OF "EPICREALM LICENSING, LP", A LIMITED PARTNERSHIP ORGANIZED AND EXISTING UNDER THE LAWS OF THE STATE OF DELAWARE, AS RECEIVED AND FILED IN THIS OFFICE THE TWENTY-NINTH DAY OF DECEMBER, A.D. 2005, AT 1:40 O'CLOCK P.M.



Harriet Smith Windsor

Harriet Smith Windsor, Secretary of State

AUTHENTICATION: 4416658

DATE: 12-30-05

4084262 8100M

051071681

State of Delaware
Secretary of State
Division of Corporations
Delivered 01:42 PM 12/29/2005
FILED 01:40 PM 12/29/2005
SRV 051071681 - 4084262 FILE

CERTIFICATE OF MERGER
OF
EPICREALM LICENSING, LLC,
a Texas limited liability company
With and Into
EPICREALM LICENSING, LP,
a Delaware limited partnership

Pursuant to Section 17-211 of the Delaware Revised Uniform Limited Partnership Act (the "Act"), the undersigned surviving limited partnership submits the following Certificate of Merger for filing and certifies that:

1. The name and jurisdiction of formation or organization of the domestic limited partnership and the foreign limited liability company which are to merge are:

Name	Jurisdiction
EPICREALM LICENSING, LP	Delaware
EPICREALM LICENSING, LLC	Texas

2. An Agreement and Plan of Merger (the "Merger Agreement") has been approved and executed in accordance with the provisions of Section 17-211 of the Act by the foreign limited liability company and the domestic limited partnership which are to merge.

3. The name of the surviving limited partnership is epicRealm Licensing, LP.

4. The Merger Agreement is on file at the place of business of the surviving limited partnership located at 2828 Hood Street # 806, Dallas, Texas 75219.

5. A copy of the Merger Agreement will be furnished by the surviving limited partnership on request and without cost to any member of the foreign limited liability company or any partner of the domestic limited partnership which are to merge.

6. The aforementioned merger shall be effective as of the date of the execution of this Certificate of Merger.

]

IN WITNESS WHEREOF, this Certificate of Merger has been duly executed as of December 29, 2005, and is being filed in accordance with Section 17-211 of the Act by an authorized person of the surviving domestic limited partnership in the merger.

EPICREALM LICENSING, LP,
A Delaware Limited Partnership

By: epicRealm Licensing, LLC,
A Delaware limited liability company
its General Partner

By: _____
Name: Terry Rakas
Title: Manager

[Signature Page to Certificate of Merger – epicRealm Licensing, (TX) into epicRealm Licensing,LP (DE)]

{H04739596; 1}

EXHIBIT A

U.S. Patents

Patent No.	Issue Date	Title
5,894,554	13 Apr 1999	System for Managing Dynamic Web Page Generation Requests by Intercepting Request at Web Server and Routing to Page Server Thereby Releasing Web Server to Process Other Requests.
6,415,335	02 Jul 2002	System and Method for Managing Dynamic Web Page Generation Requests.
6,446,111	03 Sep 2002	Method and Apparatus for Client-Server Communication Using a Limited Capability Client Over a Low-Speed Communications Link.

U.S. Patent Applications

Serial No.	Filing Date	Title
09/590,760	08 June 2000	Method and Apparatus for Content Synchronization.
09/592,486	08 June 2000	Method and Apparatus for Dynamic Data Flow Control.
09/731,365	06 Dec 2000	Method and System for Adaptive Prefetching.
09/640,478	16 Aug 2000	Method and System for Uniform Resource Locator Transformation.
09/759,392	12 Jan 2001	Method and System for Community Data Caching.
09/759,406	12 Jan 2001	Method and System for Dynamic Distributed Data Caching.

TO:B. K. DRINKWATER, C/O BAKER BOTTS L.L.P. COMPANY:2001 ROSS AVENUE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNDER SECRETARY OF COMMERCE FOR INTELLECTUAL PROPERTY AND
DIRECTOR OF THE UNITED STATES PATENT AND TRADEMARK OFFICE

SEPTEMBER 10, 2007

PTAS

500350891A

B. K. DRINKWATER, C/O BAKER BOTTS L.L.P.
2001 ROSS AVENUE
SUITE 600
DALLAS, TX 75201

UNITED STATES PATENT AND TRADEMARK OFFICE
NOTICE OF RECORDATION OF ASSIGNMENT DOCUMENT

THE ENCLOSED DOCUMENT HAS BEEN RECORDED BY THE ASSIGNMENT DIVISION OF THE U.S. PATENT AND TRADEMARK OFFICE. A COMPLETE MICROFILM COPY IS AVAILABLE AT THE ASSIGNMENT SEARCH ROOM ON THE REEL AND FRAME NUMBER REFERENCED BELOW.

PLEASE REVIEW ALL INFORMATION CONTAINED ON THIS NOTICE. THE INFORMATION CONTAINED ON THIS RECORDATION NOTICE REFLECTS THE DATA PRESENT IN THE PATENT AND TRADEMARK ASSIGNMENT SYSTEM. IF YOU SHOULD FIND ANY ERRORS OR HAVE QUESTIONS CONCERNING THIS NOTICE, YOU MAY CONTACT THE EMPLOYEE WHOSE NAME APPEARS ON THIS NOTICE AT 571-272-3350. PLEASE SEND REQUEST FOR CORRECTION TO: U.S. PATENT AND TRADEMARK OFFICE, MAIL STOP: ASSIGNMENT SERVICES BRANCH, P.O. BOX 1450, ALEXANDRIA, VA 22313.

RECORDATION DATE: 09/10/2007 REEL/FRAME: 019795/0968
NUMBER OF PAGES: 5

BRIEF: NUNC PRO TUNC ASSIGNMENT (SEE DOCUMENT FOR DETAILS).
DOCKET NUMBER: 066241.0146

ASSIGNOR:
EPICREALM LICENSING, LP DOC DATE: 08/31/2007

ASSIGNEE:
PARALLEL NETWORKS, LLC
1700 PACIFIC AVENUE
SUITE 2320
DALLAS, TEXAS 75201

SERIAL NUMBER: 08636477 FILING DATE: 04/23/1996
PATENT NUMBER: 5894554 ISSUE DATE: 04/13/1999
TITLE: SYSTEM FOR MANAGING DYNAMIC WEB PAGE GENERATION REQUESTS BY INTERCEPTING REQUEST AT WEB SERVER AND ROUTING TO PAGE SERVER THEREBY RELEASING WEB SERVER TO PROCESS OTHER REQUESTS

TO:B. K. DRINKWATER, C/O BAKER BOTTS L.L.P. COMPANY:2001 ROSS AVENUE

019795/0968 PAGE 2

SERIAL NUMBER: 09234048 FILING DATE: 01/19/1999
PATENT NUMBER: 6415335 ISSUE DATE: 07/02/2002
TITLE: SYSTEM AND METHOD FOR MANAGING DYNAMIC WEB PAGE GENERATION REQUESTS

ASSIGNMENT SERVICES BRANCH
PUBLIC RECORDS DIVISION

PATENT ASSIGNMENT

Electronic Version v1.1
Stylesheet Version v1.1

09/10/2007
500350891

SUBMISSION TYPE:	NEW ASSIGNMENT
NATURE OF CONVEYANCE:	NUNC PRO TUNC ASSIGNMENT
EFFECTIVE DATE:	08/10/2007

CONVEYING PARTY DATA

Name	Execution Date
EPICREALM LICENSING, LP	08/31/2007

RECEIVING PARTY DATA

Name:	PARALLEL NETWORKS, LLC
Street Address:	1700 Pacific Avenue
Internal Address:	Suite 2320
City:	Dallas
State/Country:	TEXAS
Postal Code:	75201

PROPERTY NUMBERS Total: 2

Property Type	Number
Patent Number:	5894554
Patent Number:	6415335

CH $80.00 5894554

CORRESPONDENCE DATA

Fax Number: (214)661-4691
Correspondence will be sent via US Mail when the fax attempt is unsuccessful.
Phone: 214.953.6691
Email: b.k.drinkwater@bakerbotts.com
Correspondent Name: B. K. Drinkwater, c/o Baker Botts L.L.P.
Address Line 1: 2001 Ross Avenue
Address Line 2: Suite 600
Address Line 4: Dallas, TEXAS 75201

ATTORNEY DOCKET NUMBER:	066241.0146
NAME OF SUBMITTER:	B. K. Drinkwater

TO:B. K. DRINKWATER, C/O BAKER BOTTS L.L.P. COMPANY:2001 ROSS AVENUE

Total Attachments: 3
source=Parallel Networks LLC assignment#page1.tif
source=Parallel Networks LLC assignment#page2.tif
source=Parallel Networks LLC assignment#page3.tif

PATENT ASSIGNMENT

WHEREAS, EPICREALM LICENSING, LP (hereafter "Assignor"), a Delaware limited partnership, having an address of 1700 Pacific Avenue, Suite 2320, Dallas, Texas 75201, is the owner of the patents and patent applications set forth on Exhibit A hereto and the inventions described and claimed therein (hereafter the "Patent Property"); and

WHEREAS, PARALLEL NETWORKS, LLC (hereafter "Assignee"), a Texas limited liability company, having an address of 1700 Pacific Avenue, Suite 2320, Dallas, Texas 75201, did acquire all right, title and interest in and to the Patent Property as the result of an Asset Purchase Agreement effective August 10, 2007.

NOW, THEREFORE, for good and valuable consideration, the receipt and sufficiency of which is hereby acknowledged, Assignor does hereby sell, assign, transfer and set over *nunc pro tunc* to Assignee, all its right, title and interest in and to the Patent Property, as well as all provisionals, continuations, divisions, and continuations-in-part of said Patent Property, and all reissues, reexaminations, and extensions thereof, the same to be held and enjoyed by Assignee for its own use and benefit, and for the use and benefit of its successors, assigns, or legal representatives, to the end of the term or terms for which such Patent Property may be granted or reissued, as fully and entirely as the same would have been held and enjoyed by Assignor if this assignment and sale had not been made.

Assignor also assigns to Assignee, all right, title and interest in and to the inventions disclosed in said Patent Property throughout the world, including the right to file applications and obtain patents, utility models, industrial models and designs for said Patent Property in its own name throughout the world, including all rights to publish cautionary notices

reserving ownership of said inventions and all rights to register said Patent Property in appropriate registries; and Assignor further agrees to execute any and all powers of attorney, applications, assignments, declarations, affidavits, and any other papers in connection therewith necessary to perfect such right, title and interest in Assignee.

Assignor also assigns unto Assignee all claims for damages by reason of infringement prior to the assignment date of the Patent Property throughout the world, with the right to sue for and collect the same for its own use and benefit, and for the use and benefit of its successors, assigns and other legal representatives.

Assignor, at the expense of Assignee, will testify in any legal proceedings, sign all lawful papers, execute all provisional, divisional, continuation, continuation-in-part, reissue and substitute applications, make lawful oaths and declarations, and generally do everything possible to vest title in Assignee and to aid Assignee to obtain and enforce proper protection for said Patent Property in all countries.

IN WITNESS WHEREOF, Assignor and its duly authorized representative has caused this Patent Assignment to be executed on the date and in the capacity shown below.

EPICREALM LICENSING, LP

By: EPICREALM LICENSING, LLC,
A Delaware Limited Liability Company
Its: General Partner

By: _____
Terry Pokas
Its: Managing Partner
Date: August 31, 2007

EXHIBIT A

U.S. Patents

Patent No.	Issue Date	Title
5,894,554	13 Apr 1999	System for Managing Dynamic Web Page Generation Requests by Intercepting Request at Web Server and Routing to Page Server Thereby Releasing Web Server to Process Other Requests.
6,415,335	02 Jul 2002	System and Method for Managing Dynamic Web Page Generation Requests.

Exhibit 8

JENNER&BLOCK

October 27, 2007

Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611
Tel 312-222-9350
www.jenner.com

Chicago
Dallas
New York
Washington, DC

VIA EMAIL

Theodore T. Herhold, Esq.
Townsend and Townsend and Crew
379 Lytton Avenue
Palo Alto, California 94301-1431

Paul D. Margolis
Tel 312 923-8323
Fax 312 923-8423
pmargolis@jenner.com

Re: ***Oracle Corp. et al. v. epicRealm Licensing, LP***
CA No. 06-CV-414 (SLR)

Dear Ted:

We are writing to summarize the telephonic meet and confer that you, Eric Hutchins, Pat Patras and I conducted last night. We once again addressed four issues: (1) the stipulation regarding Parallel Networks; (2) Oracle's document production; (3) issues regarding Oracle's source code production; and (4) epicRealm's discovery responses.

First, with respect to the stipulation that we sent for your review regarding Parallel Networks, you stated that you may have a few minor changes, but agreed to the substitution of Parallel Networks for epicRealm as the defendant in this action. Additionally, you informed us that Oracle was working on a proposed amended complaint that would add an inequitable conduct claim, but that you would be in a better position to talk about this issue next week.

Second, with respect to Oracle's document production, you indicated that Oracle was not prepared at this time to provide a date certain by which its document production would be complete. In addition, you informed us that you were in the process of reviewing the categories of documents listed in David Bennett's October 23 letter that we believed to be missing from Oracle's production. You also explained that we would be receiving executable code for Oracle current software releases next week, but that it would take longer to produce executable code for older versions of Oracle software.

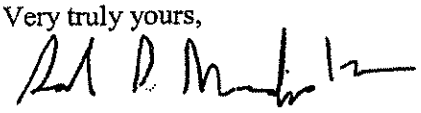
Third, we discussed issues relating to Oracle's production of source code. You informed us that Oracle will provide the manifest of the contents of the Texas computers and MD5 checksums next week. You also stated that Oracle would be willing to provide programs needed to read the contents of the Texas computers such as Microsoft Word and Adobe Acrobat. In addition, we provided you with a folder that we have identified as missing from the source code. With respect to our request to depose an Oracle Rule 30(b)(6) witness pursuant to our August 31, 2007 notice, you were not able to provide us with a date for the deposition yesterday. At the present time, we have an impasse as to when, in addition to where, the deposition will be. We believe the deposition should proceed at the escrow facility where Oracle has produced the source code.

You indicated that you believed the deposition should proceed at Townsend's offices, even though Townsend does not currently have a copy of what Oracle produced at the escrow facility.

Fourth, we discussed Oracle's complaints about epicRealm's discovery responses. We agreed to supplement our response to Interrogatory No. 4 and to identify secondary considerations of nonobviousness in response to Interrogatory No. 12 by November 9. With respect to Interrogatory Nos. 6, 7, 10, 11, 12, 14, 15 and 16, we agreed to provide more specificity with respect to its use of Rule 33(d) in answering those interrogatories by November 30.

You indicated that you no longer want to submit a joint letter to the Court in advance of Monday's hearing. If you would like to speak again in advance of Monday's hearing, please let us know.

Very truly yours,



Paul D. Margolis

PDM

cc: Robert J. Artuz, Esq.
Mary B. Graham, Esq.
George S. Bosy, Esq.
Patrick L. Patras, Esq.
Benjamin J. Bradford, Esq.

Exhibit 9

From: Herhold, Ted T. [mailto:ttherhold@townsend.com]
Sent: Thursday, November 01, 2007 6:15 PM
To: Bosy, George S
Cc: Roper, Harry J; Patras, Patrick L
Subject: RE: Parallel Networks

George, as I mentioned to Pat, the plan is for us to draft an amended complaint replacing ER with PN and setting forth our inequitable conduct defense and forward that to you for review (as discussed with Pat, we believe both issues should be covered at the same time in the same amended complaint). If you agree to the proposed amended complaint, we can enter into the stip (which should be revised to account for our inequitable conduct defense) and we will go ahead and file the amended complaint, you can file your amended answer and counterclaims within, e.g., 5 days, and we can then file our amended answer to your counterclaims 5 days later. That all should be spelled out in the stip.

Let me know if this procedure is acceptable. We should have the draft amended complaint to you by Monday.

Ted

Theodore T. Herhold
Partner
Townsend and Townsend and Crew LLP
379 Lytton Ave.
Palo Alto, CA 94301
(650) 326-2400 (main)
(650) 326-2422 (fax)
(650) 752-2433 (direct)
tth@townsend.com

Exhibit 10

From: Willette, Timothy M on behalf of Bosy, George S
Sent: Friday, November 02, 2007 10:18 AM
To: Herhold, Ted T.
Cc: Roper, Harry J; Patras, Patrick L; 'tfokas@parallelnetworks.com'
Subject: RE: Parallel Networks

Ted,

This is in response to your email of November 1, 2007 on the issue of substituting Parallel Networks for epicRealm. I understand that Oracle has added a new condition, namely, that Parallel Networks agree to Oracle's amendment that would include a new charge of inequitable conduct. For the record, I have never agreed to this. However, I will reserve final judgment on the issue until I have reviewed your proposed amendment.

George

Exhibit 11

From: Herhold, Ted T. [mailto:ttherhold@townsend.com]
Sent: Thursday, November 08, 2007 12:06 AM
To: Bosy, George S; Patras, Patrick L
Cc: rhorwitz@potteranderson.com; mgraham@MNAT.com
Subject: Oracle/epicRealm

George and Pat, attached is a draft amended complaint substituting in Parallel Networks for epicRealm and asserting a claim for inequitable conduct. Substantive changes are highlighted for ease of reference. Please give me your thoughts before we finalize and submit to the court along with the proposed stipulation. With regard to the stipulation, it should be revised to reflect both the substitution of parties and the addition of Oracle's inequitable conduct claim. It should probably also lay out the timing for PN's answer/counterclaims, and Oracle's answer to PN's counterclaims. Let's get local counsel to weigh in on the proper wording and procedure for the stipulation. I've copied Mary and Rich to get their input.

Ted

Theodore T. Herhold
Partner
Townsend and Townsend and Crew LLP
379 Lytton Ave.
Palo Alto, CA 94301
(650) 326-2400 (main)
(650) 326-2422 (fax)
(650) 752-2433 (direct)
tth@townsend.com

This message is from a lawyer and may contain confidential and/or privileged information. If you are not the intended recipient, please notify me immediately by return email and delete this message from your computer system.

DRAFT - 11/07/07

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

ORACLE CORPORATION and ORACLE U.S.A. INC.,	)	
Plaintiffs,	)	
v.	)	C. A. No. 1:06-cv-00414 (SLR)
PARALLEL NETWORKS, LLC,	)	
Defendant.	)	

AMENDED COMPLAINT FOR DECLARATORY JUDGMENT

Plaintiffs Oracle Corporation and Oracle U.S.A. Inc. (together referred to herein as "Oracle") hereby submit this amended complaint against Defendant Parallel Networks, LLC ("Parallel Networks") as follows:

THE PARTIES

1. Parallel Networks alleges that in August 2007 Defendant and Counterclaimant epicRealm Licensing, LP ("epicRealm") assigned to Parallel Networks all right, title and interest in United States Patent No. 5,894,554 (the "'554 patent") and United States Patent No. 6,415,335 (the "'335 patent") (collectively, the "Parallel Networks patents"). By virtue of this alleged assignment, Oracle, epicRealm and Parallel Networks have agreed to substitute Parallel Networks for epicRealm in this action and to dismiss epicRealm without prejudice.

2. Plaintiff Oracle Corporation is a corporation organized and existing under the laws of the State of Delaware, having its principal place of business at 500 Oracle Parkway, Redwood Shores, California 94065.

3. Plaintiff Oracle U.S.A. Inc. is a corporation organized and existing under the laws of the State of Colorado, having its principal place of business at 500 Oracle Parkway, Redwood Shores, California 94065. Oracle U.S.A. Inc. is a wholly owned subsidiary of Oracle Corporation.

4. Defendant Parallel Networks is a limited liability company organized and existing under the laws of the State of Texas, having its principal place of business at 1700 Pacific Avenue, Suite 2320, Dallas, Texas 75201.

JURISDICTION AND VENUE

5. This is a complaint for declaratory relief under the patent laws of the United States. This Court has subject matter jurisdiction over this action pursuant to 28 U.S.C. §§ 1331, 1338(a), 2201(a) and 2202.

6. This Court has personal jurisdiction over Parallel Networks because epicRealm is organized under the laws of the State of Delaware and Parallel Networks seeks to substitute itself in this action for epicRealm. Further, the parties have stipulated that jurisdiction and venue are proper in this judicial district and that Parallel Networks will not seek to transfer this case or any case involving the patents-in-suit, and any patents related thereto, to another judicial district.

7. Venue is proper in this judicial district pursuant to 28 U.S.C. §§ 1391(b), (c) and 1400(b).

THE PARALLEL NETWORKS PATENTS

8. The '554 patent issued on April 13, 1999 and is entitled, System for Managing Dynamic Web Page Generation Requests by Intercepting Request at Web Server and Routing to Page Server Thereby Releasing Web Server to Process Other Requests.

9. The '335 patent issued on July 2, 2002 from a division of U.S. Patent Application No. 08/636,477, now the '554 patent, and is entitled, System and Method for Managing Dynamic Web Page Generation.

10. Parallel Networks asserts that it is the owner and assignee of the Parallel Networks patents and has the right to enforce the Parallel Networks patents against Oracle.

EPICREALM'S PATENT INFRINGEMENT LAWSUITS

11. On May 2, 2005, epicRealm, then known as epicRealm Licensing, LLC, filed a patent infringement complaint in the District Court for the Eastern District of Texas (Marshall Division) accusing defendants Autoflex Leasing, Inc., eHarmony.com, Inc., Friendfinder Network, Inc., Grande Communication Networks, Inc., IJL-NCP, LLC and Transplace Texas, LP of infringing the Parallel Networks patents. That action was assigned Civil Action No. 2:05-CV-163 (the "'163 action") and was originally assigned to United States District Judge T. John Ward. On June 10, 2005, epicRealm filed a First Amended Complaint in the '163 action.

12. On August 5, 2005, epicRealm filed a patent infringement complaint in the District Court for the Eastern District of Texas (Marshall Division) accusing defendants Franklin Covey Co., Clark Consulting, Inc., The Macerich Company, Safelite Group, Inc., Herbalife International of America, Inc. and Pink Sheets, LLC of infringing the Parallel Networks patents. That action was assigned Civil Action Number 2:05-CV-356 (the "'356 action") and was originally assigned to United States District Judge T. John Ward. On November 2, 2005, epicRealm filed a First Amended Complaint in the '356 action.

13. On November 2, 2005, Judge Ward consolidated the '163 and '356 actions (the "Consolidated epicRealm Actions").

14. On November 9, 2005 and November 16, 2005 the Consolidated epicRealm Actions were reassigned to United States District Judge David Folsom.

15. On January 27, 2006, epicRealm filed a Second Amended Complaint in the Consolidated epicRealm Actions accusing defendants Autoflex Leasing, Inc., eHarmony.com, Inc., Friendfinder Network, Inc., Grande Communications Networks, Inc., Transplace Texas, LP, Franklin Covey Co., Clark Consulting, Inc., Macerich Company, Safelite Group, Inc., Herbalife International of America, Inc. and Pink Sheets, LLC of infringing the Parallel Networks patents.

EPICREALM'S PATENT INFRINGEMENT ACCUSATIONS AND ORACLE'S REASONABLE APPREHENSION OF SUIT

16. In its Second Amended Complaint, epicRealm alleges that the named defendants infringe the Parallel Networks patents because they use systems and methods for managing requests for dynamic web pages.

17. Oracle sells and licenses software products to numerous customers for use in conjunction with the delivery of dynamic web pages. Oracle has entered into agreements requiring Oracle to indemnify most of these customers in the event a customer is sued for patent infringement for its use of licensed Oracle software.

18. Safelite Group, Inc. ("Safelite"), a defendant in the '356 action, is a customer of Oracle in that Safelite has purchased and licensed certain software products, including but not limited to Oracle E-Business Suite Release 11i, (collectively, the "Oracle Software") from Oracle.

19. On January 26, 2006, Safelite filed a Third Party Complaint against Oracle and International Business Machines Corp. ("IBM") in the '356 action (the "Safelite Action").

20. In its Third Party Complaint and in a January 26, 2006 letter from Safelite to Oracle (the "Safelite Letter"), Safelite alleged that under an "Oracle License & Services Agreement" dated November 29, 2002, as amended, (the "Oracle Agreement"), it purchased and licensed the Oracle Software from Oracle.

21. In its Third Party Complaint and in the Safelite Letter, Safelite alleged that epicRealm's infringement contentions are broad enough to encompass or implicate "virtually any system/method that generates web pages with dynamic content" including "Oracle products licensed to Safelite."

22. In its Third Party Complaint and in the Safelite Letter, Safelite alleged that under the Oracle Agreement and under common law it is entitled to indemnification from Oracle if a product provided by Oracle to Safelite is accused of infringing a third party's intellectual property rights, and Safelite demanded that Oracle agree to defend and indemnify Safelite from any and all epicRealm claims that any Oracle product infringes the Parallel Networks patents.

23. In its Third Party Complaint, Safelite prayed that the District Court for the Eastern District of Texas hold Safelite entitled to be fully indemnified against any monetary judgment, against Safelite and in favor of epicRealm, as to any claim that any Oracle product provided or licensed to Safelite infringes any claim of the Parallel Networks patents.

24. In a January 25, 2006 letter from epicRealm to Clark Consulting Inc. ("Clark"), a defendant in the '356 action, epicRealm alleged that a system or method that employed proprietary server software "could, if used to generate web pages with dynamic content, be configured in a way that would infringe the claims of the [Parallel Networks] patents." The January 25, 2006 letter from epicRealm to Clark (the "epicRealm Letter") is attached hereto as Exhibit C.

25. In the epicRealm Letter, epicRealm stated that Clark was required to provide discovery disclosures to epicRealm regarding Clark's use of proprietary server software for generating web pages.

26. In the '356 action, epicRealm demanded and received discovery from Safelite regarding Safelite's use of the Oracle Software.

27. In June 2006, epicRealm settled its dispute with Safelite and stipulated to a dismissal with prejudice of its claims against Safelite in the Consolidated epicRealm Actions. The District Court for the Eastern District of Texas entered an order dismissing epicRealm's claims with prejudice on June 29, 2006.

28. In June 2006, Safelite stipulated to dismissing with prejudice its Third Party Complaint against Oracle and IBM, pursuant to an agreement whereby Oracle contributed money to Safelite's settlement with epicRealm. The District Court for the Eastern District of Texas entered an order dismissing Safelite's Third Party Complaint with prejudice on June 29, 2006.

29. On June 30, 2006, Oracle filed a complaint against epicRealm in this Court seeking declaratory judgment of noninfringement and invalidity of the Parallel Networks patents.

30. On April 13, 2007, epicRealm filed an answer and counterclaim against Oracle alleging that Oracle infringes one or more claims of the Parallel Networks patents.

31. Parallel Networks alleges that in August 2007 Defendant and Counterclaimant epicRealm assigned to Parallel Networks all right, title and interest in the Parallel Networks patents.

32. epicRealm's patent infringement allegations in the Consolidated epicRealm Actions against Safelite and others, the epicRealm Letter, epicRealm's demand for discovery of

Oracle information from Safelite, and epicRealm's patent infringement counterclaims asserted against Oracle in this action cause Oracle to have a reasonable apprehension that (1) Parallel Networks will accuse Oracle, Oracle's products and/or Oracle's customers of infringing one or more claims of the Parallel Networks patents, and/or (2) additional Oracle customers will sue Oracle for indemnity as a result of epicRealm's or Parallel Networks' patent infringement claims.

FIRST CLAIM FOR RELIEF

(Declaratory Judgment Of Noninfringement Of The Parallel Networks Patents)

33. Oracle incorporates by reference Paragraphs 1 through 32 above.

34. By virtue of epicRealm's patent infringement allegations in the Consolidated epicRealm Actions against Safelite and others, the epicRealm Letter, epicRealm's demand for discovery of Oracle information from Safelite, and epicRealm's patent infringement counterclaims asserted against Oracle in this action, an actual controversy exists between Oracle and Parallel Networks as to whether Oracle, Oracle's products and/or one or more Oracle customers infringe the Parallel Networks patents.

35. Oracle has not and does not infringe, literally or under the doctrine of equivalents, either directly, indirectly or willfully, any valid and enforceable claim of the Parallel Networks patents.

36. Pursuant to 28 U.S.C. §§ 2201 and 2202, a judicial determination of the respective rights of the parties with respect to Oracle's noninfringement of the Parallel Networks patents is necessary and appropriate under the circumstances.

SECOND CLAIM FOR RELIEF

(Declaratory Judgment Of Invalidity Of The Parallel Networks Patents)

37. Oracle incorporates by reference Paragraphs 1 through 36 above.

38. By virtue of epicRealm's patent infringement allegations in the Consolidated epicRealm Actions against Safelite and others, the epicRealm Letter, epicRealm's demand for discovery of Oracle information from Safelite, and epicRealm's patent infringement counterclaims asserted against Oracle in this action, an actual controversy exists between Oracle and epicRealm as to the validity of the Parallel Networks patents.

39. Each claim of the Parallel Networks patents is invalid for failure to meet one or more of the conditions of patentability specified in 35 U.S.C. §§ 101, 102, 103 and/or 112.

40. Pursuant to 28 U.S.C. §§ 2201 and 2202, a judicial determination of the respective rights of the parties with respect to the validity of the Parallel Networks patents is necessary and appropriate under the circumstances.

THIRD CLAIM FOR RELIEF

(Declaratory Judgment Of Unenforceability Of The Parallel Networks Patents)

41. Oracle incorporates by reference Paragraphs 1 through 40 above.

42. By virtue of epicRealm's patent infringement allegations in the Consolidated epicRealm Actions against Safelite and others, the epicRealm Letter, epicRealm's demand for discovery of Oracle information from Safelite, and epicRealm's patent infringement counterclaims against Oracle in this action, an actual controversy exists between Oracle and Parallel Networks as to the enforceability of the Parallel Networks patents.

43. The Parallel Networks patents are unenforceable because one or more of the Parallel Networks patent inventors, the patentees, and their attorneys engaged in inequitable conduct during the prosecution of the Parallel Networks patents in the USPTO.

44. Particularly, prior to September 1, 1995, inventor Keith Lowery ("Lowery") researched and had knowledge of highly material prior art including but not limited to

publications describing -- and the prior sale, offer for sale or public use of -- Oracle Web Server products (the "Oracle Prior Art") that were known and available to the public prior to the alleged conception and reduction to practice of the claimed inventions of the Parallel Networks patents.

45. Additionally, prior to the issuance of the Parallel Networks patents, the attorneys responsible for prosecuting the Parallel Networks patents had knowledge of additional Oracle Prior Art, including but not limited to the prior art methods and apparatus described in U.S. Patent No. 5,761,763 and the prior art references cited therein.

46. The Oracle Prior Art anticipates or, alone or in combination with other references, renders obvious one or more claims of the Parallel Networks patents and was far more material than any of the prior art cited in connection with the prosecution of the Parallel Networks patents.

47. Lowery, the other named inventors, the patentees, and the prosecuting attorneys all failed to submit the Oracle Prior Art to the USPTO during the prosecution of the Parallel Networks patents in violation of 37 C.F.R. § 1.56. This failure to disclose material prior art was intended to deceive the USPTO.

48. Pursuant to 28 U.S.C. §§ 2201 and 2202, a judicial determination of the respective rights of the parties with respect to the enforceability of the Parallel Networks patents is necessary and appropriate under the circumstances.

PRAYER

WHEREFORE, Oracle requests entry of judgment in its favor and against epicRealm as follows:

a. Declaring that Oracle and its software products that are used in conjunction with the delivery of dynamic web pages do not infringe, literally or under the doctrine

of equivalents, either directly, indirectly or willfully, any valid and enforceable claim of the Parallel Networks patents;

b. Declaring that the claims of the Parallel Networks patents are invalid;

c. Declaring that the Parallel Networks patents are unenforceable due to inequitable conduct;

d. Decreeing this case an "exceptional case" within the meaning of 35 U.S.C. § 285 and awarding reasonable attorneys' fees to Oracle; and

e. Awarding Oracle such other costs and further relief as the Court deems just and proper.

MORRIS, NICHOLS, ARSHT & TUNNELL LLP

Mary B. Graham (#2256)
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
302.658.9200

Attorneys for Plaintiffs
Oracle Corporation and Oracle U.S.A. Inc.

OF COUNSEL:

James G. Gilliland
Theodore T. Herhold
Chad E. King
Robert J. Artuz
Eric M. Hutchins
Eric A. Mercer
TOWNSEND AND TOWNSEND AND CREW LLP
379 Lytton Avenue
Palo Alto, CA 94301
650.326.2400

Dated: November 6, 2007
526725

Exhibit 12

-----Original Message-----
From: Willette, Timothy M [mailto:TWillette@jenner.com] **On Behalf Of** Bosy, George S
Sent: Friday, November 30, 2007 9:23 AM
To: Herhold, Ted T.
Cc: Roper, Harry J; Patras, Patrick L
Subject: Inequitable conduct charge

Ted,

Don't forget that you owe me the further information with respect to your charge of inequitable conduct.

George

George S. Bosy
Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611-7603
(312) 923-8305
(312) 923-8405 (fax)

Exhibit 13

From: Willette, Timothy M **On Behalf Of** Bosy, George S
Sent: Thursday, December 13, 2007 3:04 PM
To: 'Herhold, Ted T.'
Cc: Patras, Patrick L
Subject: RE: Oracle et al. v. epicRealm Licensing, LP, (C.A. No. 06-414-SLR)

Ted,

This is in response to your e-mail of December 11, 2007 with respect to our motion to substitute Parallel Networks as the real party in interest.

Your e-mail contains assertions that are knowingly erroneous and that suggest that our good faith attempt to reach an agreement on the substitution issue was doomed to failure because of Oracle's bad faith from the start. Contrary to your assertions, Oracle never agreed to substitute without the further condition that we agree to your amending your complaint to include the new and meritless charge of inequitable conduct. Indeed, it was not until well after negotiations began that Oracle added this condition in an attempt to gain a substantial advantage for Oracle having nothing to do with what really should have been a straightforward substitution issue. And so, your assertion that we have violated Local Rule 7.1.1 is specious.

As you know, we will not agree to Oracle's request to substantially amend its complaint in exchange for what Oracle should have agreed to in the first instance -- the substitution of Parallel Networks for epicRealm. I await to see what basis Oracle has for opposing our motion, because you have never articulated one.

George

George S. Bosy
Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611-7603
(312) 923-8305
(312) 923-8405 (fax)

Exhibit 14

TOWNSEND
and
TOWNSEND
and
CREW
LLP

Palo Alto

379 Lytton Avenue
Palo Alto, California 94301-1431
Tel 650.326.2400
Fax 650.326.2422

gtorres@townsend.com

December 14, 2007

VIA FEDEX

Benjamin J. Bradford
Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611-7603

Re: *Oracle Corp. and Oracle U.S.A., Inc. v. epicRealm Licensing, LP*
U.S.D.C., District of Delaware, C.A. No. 06- 414-SLR
Our File No.: 015917-03980US

Dear Mr. Bradford:

Enclosed please find one box containing a production of documents by Plaintiffs in the above-referenced matter. Please note that these documents, some of which are marked "Highly Confidential: Outside Counsel's Eyes Only," are identified and referenced for production as follows:

- ORCL01051317-ORCL01051747

This production also includes an original publication and a CD containing source code in native format.

Please contact our office if you have any questions regarding the enclosed production.

Sincerely,

Gloria Torres
Litigation Paralegal

Encls. (1 book, 2 discs)

61235152 v1

Exhibit 15

TOWNSEND
and
TOWNSEND
and
CREW
LLP

Palo Alto

379 Lytton Avenue
Palo Alto, California 94301-1431
Tel 650.326.2400
Fax 650.326.2422

dtrevino@townsend.com

December 19, 2007

VIA FEDEX

Benjamin J. Bradford
Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611-7603

Re: *Oracle Corp. and Oracle U.S.A., Inc. v. epicRealm Licensing, LP*
U.S.D.C., District of Delaware, C.A. No. 06- 414-SLR
Our File No.: 015917-03980US

Dear Mr. Bradford:

Enclosed please find one box containing a production of documents by Plaintiffs in the above-referenced matter. Please note that these documents, some of which are marked "Highly Confidential: Outside Counsel's Eyes Only," are identified and referenced for production as follows:

- ORCL01051748-1078548

Please note that this production includes media packs containing executable software. Please also note that related documents are currently being processed by our vendor and we anticipate producing same later this week.

Please contact our office if you have any questions regarding the enclosed production.

Sincerely,

C. Daniel Treviño
Senior Litigation Paralegal

Encls. (92 discs)

61239501 v1

Exhibit 16

TOWNSEND
and
TOWNSEND
and
CREW
LLP

Palo Alto

379 Lytton Avenue
Palo Alto, California 94301-1431
Tel 650.326.2400
Fax 650.326.2422

dtrevino@townsend.com

December 20, 2007

VIA FEDEX

Benjamin J. Bradford
Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611-7603

Re: *Oracle Corp. and Oracle U.S.A., Inc. v. epicRealm Licensing, LP*
U.S.D.C., District of Delaware, C.A. No. 06- 414-SLR
Our File No.: 015917-03980US

Dear Mr. Bradford:

Enclosed please find a production of documents by Plaintiffs in the above-referenced matter. Please note that these documents, all of which are marked "Highly Confidential: Outside Counsel's Eyes Only," are identified and referenced for production as follows:

- ORCL01078549-1201090

Please also note that related documents are still being processed by our vendor and we anticipate producing same on a rolling basis.

Please contact our office if you have any questions regarding the enclosed production.

Sincerely,

C. Daniel Treviño
Senior Litigation Paralegal

Encls. (8 DVDs)

61240540 v1

Exhibit 17

TOWNSEND
and
TOWNSEND
and
CREW
LLP

Palo Alto

379 Lytton Avenue
Palo Alto, California 94301-1431
Tel 650.326.2400
Fax 650.326.2422

dtrevino@townsend.com

December 21, 2007

VIA FEDEX

Benjamin J. Bradford
Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611-7603

Re: *Oracle Corp. and Oracle U.S.A., Inc. v. epicRealm Licensing, LP*
U.S.D.C., District of Delaware, C.A. No. 06- 414-SLR
Our File No.: 015917-039800US

Dear Mr. Bradford:

Enclosed please find a production of documents by Plaintiffs in the above-referenced matter. Please note that these documents, all of which are designated "Highly Confidential: Outside Counsel's Eyes Only," are identified and referenced for production as follows:

- ORCL01201091-1414792

Please contact our office if you have any questions regarding the enclosed production.

Sincerely,

C. Daniel Treviño
Senior Litigation Paralegal

Encls. (17 discs)

61241985 v1

Exhibit 18

From: Artuz, Robert J. [mailto:rjartuz@townsend.com]
Sent: Wednesday, January 02, 2008 7:57 PM
To: Patras, Patrick L
Cc: Bosy, George S; Herhold, Ted T.; Hutchins, Eric M.
Subject: Oracle's Amended Complaint for Declaratory Judgment.DOC

Pat,

I don't believe we ever got final word from epicRealm as to whether it will stipulate to Oracle amending its complaint to add an inequitable conduct claim.

Attached again is a draft of Oracle's proposed amended complaint. This draft contains additional facts to support its claim. Oracle must move to amend its complaint by January 4. Please state whether or not epicRealm will oppose Oracle's motion to seek the amendment. If we do not hear from you by Friday, we will assume that you intend to oppose Oracle's motion.

Regards,

Rob Artuz

Robert J. Artuz
Associate
Townsend and Townsend and Crew LLP
379 Lytton Ave.
Palo Alto, CA 94301
Phone: 650-326-2400
Fax: 650-326-2422
rjartuz@townsend.com

www.townsend.com

This message may contain confidential information. If you are not the intended recipient and received this in error, any use, or distribution is strictly prohibited. Please also notify us immediately by return e-mail, and delete this message from your computer system.

<<Oracle's Amended Complaint for Declaratory Judgment.DOC>>

DRAFT - 1/02/08

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

ORACLE CORPORATION and ORACLE U.S.A. INC., Plaintiffs, v. PARALLEL NETWORKS, LLC, Defendant.	C. A. No. 1:06-cv-00414 (SLR)

AMENDED COMPLAINT FOR DECLARATORY JUDGMENT

Plaintiffs Oracle Corporation and Oracle U.S.A. Inc. (together referred to herein as "Oracle") hereby submit this amended complaint against Defendant Parallel Networks, LLC ("Parallel Networks") as follows:

THE PARTIES

1. Parallel Networks alleges that in August 2007 Defendant and Counterclaimant epicRealm Licensing, LP ("epicRealm") assigned to Parallel Networks all right, title and interest in United States Patent No. 5,894,554 (the "'554 patent") and United States Patent No. 6,415,335 (the "'335 patent") (collectively, the "Parallel Networks patents"). By virtue of this alleged assignment, Oracle, epicRealm and Parallel Networks have agreed to substitute Parallel Networks for epicRealm in this action and to dismiss epicRealm without prejudice.

2. Plaintiff Oracle Corporation is a corporation organized and existing under the laws of the State of Delaware, having its principal place of business at 500 Oracle Parkway, Redwood Shores, California 94065.

3. Plaintiff Oracle U.S.A. Inc. is a corporation organized and existing under the laws of the State of Colorado, having its principal place of business at 500 Oracle Parkway, Redwood Shores, California 94065. Oracle U.S.A. Inc. is a wholly owned subsidiary of Oracle Corporation.

4. Defendant Parallel Networks is a limited liability company organized and existing under the laws of the State of Texas, having its principal place of business at 1700 Pacific Avenue, Suite 2320, Dallas, Texas 75201.

JURISDICTION AND VENUE

5. This is a complaint for declaratory relief under the patent laws of the United States. This Court has subject matter jurisdiction over this action pursuant to 28 U.S.C. §§ 1331, 1338(a), 2201(a) and 2202.

6. This Court has personal jurisdiction over Parallel Networks because epicRealm is organized under the laws of the State of Delaware and Parallel Networks seeks to substitute itself in this action for epicRealm. Further, the parties have stipulated that jurisdiction and venue are proper in this judicial district and that Parallel Networks will not seek to transfer this case or any case involving the patents-in-suit, and any patents related thereto, to another judicial district.

7. Venue is proper in this judicial district pursuant to 28 U.S.C. §§ 1391(b), (c) and 1400(b).

THE PARALLEL NETWORKS PATENTS

8. The '554 patent issued on April 13, 1999 and is entitled, System for Managing Dynamic Web Page Generation Requests by Intercepting Request at Web Server and Routing to Page Server Thereby Releasing Web Server to Process Other Requests.

9. The '335 patent issued on July 2, 2002 from a division of U.S. Patent Application No. 08/636,477, now the '554 patent, and is entitled, System and Method for Managing Dynamic Web Page Generation.

10. Parallel Networks asserts that it is the owner and assignee of the Parallel Networks patents and has the right to enforce the Parallel Networks patents against Oracle.

EPICREALM'S PATENT INFRINGEMENT LAWSUITS

11. On May 2, 2005, epicRealm, then known as epicRealm Licensing, LLC, filed a patent infringement complaint in the District Court for the Eastern District of Texas (Marshall Division) accusing defendants Autoflex Leasing, Inc., eHarmony.com, Inc., Friendfinder Network, Inc., Grande Communication Networks, Inc., IJL-NCP, LLC and Transplace Texas, LP of infringing the Parallel Networks patents. That action was assigned Civil Action No. 2:05-CV-163 (the "'163 action") and was originally assigned to United States District Judge T. John Ward. On June 10, 2005, epicRealm filed a First Amended Complaint in the '163 action.

12. On August 5, 2005, epicRealm filed a patent infringement complaint in the District Court for the Eastern District of Texas (Marshall Division) accusing defendants Franklin Covey Co., Clark Consulting, Inc., The Macerich Company, Safelite Group, Inc., Herbalife International of America, Inc. and Pink Sheets, LLC of infringing the Parallel Networks patents. That action was assigned Civil Action Number 2:05-CV-356 (the "'356 action") and was originally assigned to United States District Judge T. John Ward. On November 2, 2005, epicRealm filed a First Amended Complaint in the '356 action.

13. On November 2, 2005, Judge Ward consolidated the '163 and '356 actions (the "Consolidated epicRealm Actions").

14. On November 9, 2005 and November 16, 2005 the Consolidated epicRealm Actions were reassigned to United States District Judge David Folsom.

15. On January 27, 2006, epicRealm filed a Second Amended Complaint in the Consolidated epicRealm Actions accusing defendants Autoflex Leasing, Inc., eHarmony.com, Inc., Friendfinder Network, Inc., Grande Communications Networks, Inc., Transplace Texas, LP, Franklin Covey Co., Clark Consulting, Inc., Macerich Company, Safelite Group, Inc., Herbalife International of America, Inc. and Pink Sheets, LLC of infringing the Parallel Networks patents.

EPICREALM'S PATENT INFRINGEMENT ACCUSATIONS AND ORACLE'S REASONABLE APPREHENSION OF SUIT

16. In its Second Amended Complaint, epicRealm alleges that the named defendants infringe the Parallel Networks patents because they use systems and methods for managing requests for dynamic web pages.

17. Oracle sells and licenses software products to numerous customers for use in conjunction with the delivery of dynamic web pages. Oracle has entered into agreements requiring Oracle to indemnify most of these customers in the event a customer is sued for patent infringement for its use of licensed Oracle software.

18. Safelite Group, Inc. ("Safelite"), a defendant in the '356 action, is a customer of Oracle in that Safelite has purchased and licensed certain software products, including but not limited to Oracle E-Business Suite Release 11i, (collectively, the "Oracle Software") from Oracle.

19. On January 26, 2006, Safelite filed a Third Party Complaint against Oracle and International Business Machines Corp. ("IBM") in the '356 action (the "Safelite Action").

20. In its Third Party Complaint and in a January 26, 2006 letter from Safelite to Oracle (the "Safelite Letter"), Safelite alleged that under an "Oracle License & Services Agreement" dated November 29, 2002, as amended, (the "Oracle Agreement"), it purchased and licensed the Oracle Software from Oracle.

21. In its Third Party Complaint and in the Safelite Letter, Safelite alleged that epicRealm's infringement contentions are broad enough to encompass or implicate "virtually any system/method that generates web pages with dynamic content" including "Oracle products licensed to Safelite."

22. In its Third Party Complaint and in the Safelite Letter, Safelite alleged that under the Oracle Agreement and under common law it is entitled to indemnification from Oracle if a product provided by Oracle to Safelite is accused of infringing a third party's intellectual property rights, and Safelite demanded that Oracle agree to defend and indemnify Safelite from any and all epicRealm claims that any Oracle product infringes the Parallel Networks patents.

23. In its Third Party Complaint, Safelite prayed that the District Court for the Eastern District of Texas hold Safelite entitled to be fully indemnified against any monetary judgment, against Safelite and in favor of epicRealm, as to any claim that any Oracle product provided or licensed to Safelite infringes any claim of the Parallel Networks patents.

24. In a January 25, 2006 letter from epicRealm to Clark Consulting Inc. ("Clark"), a defendant in the '356 action, epicRealm alleged that a system or method that employed proprietary server software "could, if used to generate web pages with dynamic content, be configured in a way that would infringe the claims of the [Parallel Networks] patents." The January 25, 2006 letter from epicRealm to Clark (the "epicRealm Letter") is attached hereto as Exhibit C.

25. In the epicRealm Letter, epicRealm stated that Clark was required to provide discovery disclosures to epicRealm regarding Clark's use of proprietary server software for generating web pages.

26. In the '356 action, epicRealm demanded and received discovery from Safelite regarding Safelite's use of the Oracle Software.

27. In June 2006, epicRealm settled its dispute with Safelite and stipulated to a dismissal with prejudice of its claims against Safelite in the Consolidated epicRealm Actions. The District Court for the Eastern District of Texas entered an order dismissing epicRealm's claims with prejudice on June 29, 2006.

28. In June 2006, Safelite stipulated to dismissing with prejudice its Third Party Complaint against Oracle and IBM, pursuant to an agreement whereby Oracle contributed money to Safelite's settlement with epicRealm. The District Court for the Eastern District of Texas entered an order dismissing Safelite's Third Party Complaint with prejudice on June 29, 2006.

29. On June 30, 2006, Oracle filed a complaint against epicRealm in this Court seeking declaratory judgment of noninfringement and invalidity of the Parallel Networks patents.

30. On April 13, 2007, epicRealm filed an answer and counterclaim against Oracle alleging that Oracle infringes one or more claims of the Parallel Networks patents.

31. Parallel Networks alleges that in August 2007 Defendant and Counterclaimant epicRealm assigned to Parallel Networks all right, title and interest in the Parallel Networks patents.

32. epicRealm's patent infringement allegations in the Consolidated epicRealm Actions against Safelite and others, the epicRealm Letter, epicRealm's demand for discovery of

Oracle information from Safelite, and epicRealm's patent infringement counterclaims asserted against Oracle in this action cause Oracle to have a reasonable apprehension that (1) Parallel Networks will accuse Oracle, Oracle's products and/or Oracle's customers of infringing one or more claims of the Parallel Networks patents, and/or (2) additional Oracle customers will sue Oracle for indemnity as a result of epicRealm's or Parallel Networks' patent infringement claims.

FIRST CLAIM FOR RELIEF

(Declaratory Judgment Of Noninfringement Of The Parallel Networks Patents)

33. Oracle incorporates by reference Paragraphs 1 through 32 above.

34. By virtue of epicRealm's patent infringement allegations in the Consolidated epicRealm Actions against Safelite and others, the epicRealm Letter, epicRealm's demand for discovery of Oracle information from Safelite, and epicRealm's patent infringement counterclaims asserted against Oracle in this action, an actual controversy exists between Oracle and Parallel Networks as to whether Oracle, Oracle's products and/or one or more Oracle customers infringe the Parallel Networks patents.

35. Oracle has not and does not infringe, literally or under the doctrine of equivalents, either directly, indirectly or willfully, any valid and enforceable claim of the Parallel Networks patents.

36. Pursuant to 28 U.S.C. §§ 2201 and 2202, a judicial determination of the respective rights of the parties with respect to Oracle's noninfringement of the Parallel Networks patents is necessary and appropriate under the circumstances.

SECOND CLAIM FOR RELIEF

(Declaratory Judgment Of Invalidity Of The Parallel Networks Patents)

37. Oracle incorporates by reference Paragraphs 1 through 36 above.

38. By virtue of epicRealm's patent infringement allegations in the Consolidated epicRealm Actions against Safelite and others, the epicRealm Letter, epicRealm's demand for discovery of Oracle information from Safelite, and epicRealm's patent infringement counterclaims asserted against Oracle in this action, an actual controversy exists between Oracle and epicRealm as to the validity of the Parallel Networks patents.

39. Each claim of the Parallel Networks patents is invalid for failure to meet one or more of the conditions of patentability specified in 35 U.S.C. §§ 101, 102, 103 and/or 112.

40. Pursuant to 28 U.S.C. §§ 2201 and 2202, a judicial determination of the respective rights of the parties with respect to the validity of the Parallel Networks patents is necessary and appropriate under the circumstances.

THIRD CLAIM FOR RELIEF

(Declaratory Judgment Of Unenforceability Of The Parallel Networks Patents)

41. Oracle incorporates by reference Paragraphs 1 through 40 above.

42. By virtue of epicRealm's patent infringement allegations in the Consolidated epicRealm Actions against Safelite and others, the epicRealm Letter, epicRealm's demand for discovery of Oracle information from Safelite, and epicRealm's patent infringement counterclaims against Oracle in this action, an actual controversy exists between Oracle and Parallel Networks as to the enforceability of the Parallel Networks patents.

43. The Parallel Networks patents are unenforceable because one or more of the Parallel Networks patent inventors, the patentees, and their attorneys engaged in inequitable conduct during the prosecution of the Parallel Networks patents in the USPTO.

44. Particularly, prior to September 1, 1995, inventor Keith Lowery ("Lowery") researched and had knowledge of highly material prior art including but not limited to

publications describing -- and the prior sale, offer for sale or public use of -- Oracle web server products (the "Oracle Prior Art") that were known and available to the public prior to the alleged conception and reduction to practice of the claimed inventions of the Parallel Networks patents.

45. In an email dated September 1, 1995, Lowery communicated his knowledge of the Oracle Prior Art to Earl Stahl, Charles Niemeier and Imang Gupta of Gupta Corporation. Lowery's knowledge of the Oracle Prior Art was also shared with Robert L. West who was the Chief Executive Officer of InfoSpinner, Inc. (the original assignee of the '554 patent).

46. The Oracle Prior Art anticipates or, alone or in combination with other references, renders obvious one or more claims of the Parallel Networks patents and was far more material than any of the prior art that any of the Parallel Networks patent inventors, the patentees, and their attorneys cited in connection with the prosecution of the Parallel Networks patents.

47. Lowery, the other named inventors, the patentees, and the prosecuting attorneys failed to submit the Oracle Prior Art to the USPTO during the prosecution of the Parallel Networks patents in violation of 37 C.F.R. § 1.56. This failure to disclose material prior art was intended to deceive the USPTO.

48. Additionally, prior to the issuance of the Parallel Networks patents, the attorneys responsible for prosecuting the Parallel Networks patents, including Jim Salter Esq. and Sharmini Green Esq. of the law firm of Blakely Sokoloff Taylor and Zafman LLP, had knowledge of additional highly material prior art to the Parallel Networks patents, including but not limited to the prior art methods and apparatus described in U.S. Patent No. 5,761,763 issued to Oracle and the prior art references cited therein (the "'763 Prior Art").

49. The '763 Prior Art anticipates or, alone or in combination with other references, renders obvious one or more claims of the Parallel Networks patents and was far more material

than any of the prior art that any of the Parallel Networks patent inventors, the patentees, and their attorneys cited in connection with the prosecution of the Parallel Networks patents.

50. The prosecuting attorneys failed to submit the '763 Prior Art to the USPTO during the prosecution of the Parallel Networks patents in violation of 37 C.F.R. § 1.56. This failure to disclose material prior art was intended to deceive the USPTO.

51. Pursuant to 28 U.S.C. §§ 2201 and 2202, a judicial determination of the respective rights of the parties with respect to the enforceability of the Parallel Networks patents is necessary and appropriate under the circumstances.

52. Oracle hereby gives notice that discovery is still ongoing as of the date of this Amended Complaint, and Oracle intends to adduce additional inequitable conduct evidence that may be discovered during the remainder of fact discovery, including but not limited to facts that may be discovered from depositions of the named inventors of the Parallel Networks patents, the prosecuting attorneys, and any other person that had substantive involvement in the prosecution of the Parallel Networks patents.

PRAYER

WHEREFORE, Oracle requests entry of judgment in its favor and against epicRealm as follows:

a. Declaring that Oracle and its software products that are used in conjunction with the delivery of dynamic web pages do not infringe, literally or under the doctrine of equivalents, either directly, indirectly or willfully, any valid and enforceable claim of the Parallel Networks patents;

b. Declaring that the claims of the Parallel Networks patents are invalid;

c. Declaring that the Parallel Networks patents are unenforceable due to inequitable conduct;

d. Decreeing this case an "exceptional case" within the meaning of 35 U.S.C. § 285 and awarding reasonable attorneys' fees to Oracle; and

e. Awarding Oracle such other costs and further relief as the Court deems just and proper.

MORRIS, NICHOLS, ARSHT & TUNNELL LLP

Mary B. Graham (#2256)
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
302.658.9200

Attorneys for Plaintiffs
Oracle Corporation and Oracle U.S.A. Inc.

OF COUNSEL:

James G. Gilliland
Theodore T. Herhold
Chad E. King
Robert J. Artuz
Eric M. Hutchins
Eric A. Mercer
TOWNSEND AND TOWNSEND AND CREW LLP
379 Lytton Avenue
Palo Alto, CA 94301
650.326.2400

Dated: January 4, 2008
526725

Exhibit 19

Oracle's Amended Complaint for Declaratory Judgment.DOC

From: Patras, Patrick L
Sent: Thursday, January 03, 2008 3:54 PM
To: Artuz, Robert J.
Cc: Bosy, George S; Herhold, Ted T.; Hutchins, Eric M.
Subject: RE:Oracle's Amended Complaint for Declaratory Judgment.DOC

Rob:
We intend to oppose Oracle's motion.

Pat

Exhibit 20

TOWNSEND
and
TOWNSEND
and
CREW
LLP

Palo Alto

379 Lytton Avenue
Palo Alto, California 94301-1431
Tel 650.326.2400
Fax 650.326.2422

gtorres@townsend.com

January 4, 2008

VIA FEDEX

Benjamin J. Bradford
Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611-7603

Re: *Oracle Corp. and Oracle U.S.A., Inc. v. epicRealm Licensing, LP*
U.S.D.C., District of Delaware, C.A. No. 06- 414-SLR
Our File No.: 015917-039800US

Dear Mr. Bradford:

Enclosed please find one DVD containing a production of documents by Plaintiffs in the above-referenced matter. Please note that these documents, some of which are marked "Highly Confidential: Outside Counsel's Eyes Only," are identified and referenced for production as follows:

- ORCL01414793-ORCL01432885

Please contact our office if you have any questions regarding the enclosed production.

Sincerely,

Gloria Torres

Gloria Torres
Litigation Paralegal

Encls. (1 DVD)

61248042 v1

Exhibit 21

TOWNSEND
and
TOWNSEND
and
CREW
LLP

Palo Alto

379 Lytton Avenue
Palo Alto, California 94301-1431
Tel 650.326.2400
Fax 650.326.2422

direvino@townsend.com

January 10, 2008

VIA FEDEX

Benjamin J. Bradford
Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611-7603

Re: *Oracle Corp. and Oracle U.S.A., Inc. v. epicRealm Licensing, LP*
U.S.D.C., District of Delaware, C.A. No. 06- 414-SLR
Our File No.: 015917-039800US

Dear Mr. Bradford:

Enclosed please find a production of documents by Plaintiffs in the above-referenced matter. Please note that these documents, some of which are designated "Highly Confidential: Outside Counsel's Eyes Only," are identified and referenced for production as follows:

- ORCL01432886-1603855

Please contact our office if you have any questions regarding the enclosed production.

Sincerely,

C. Daniel Trevino
Case Manager

Encls. (4 discs)

61254475 v1

Exhibit 22

TOWNSEND
and
TOWNSEND
and
CREW
LLP

Palo Alto

379 Lytton Avenue
Palo Alto, California 94301-1431
Tel 650.326.2400
Fax 650.326.2422

Robert J. Artuz
650.462.5336
rjartuz@townsend.com

January 11, 2008

VIA E-MAIL

George Bosy
Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611-7603
(312) 923-8305
(312) 923-8405 (fax)

Re: *Oracle Corp. and Oracle U.S.A., Inc. v. epicRealm Licensing, LP*
Civil Action No. 06-CV-414-SLR
Our Reference No. 015917-039800US

Dear George:

I write regarding Oracle's desire to depose epicRealm's current and former employees and patent attorneys in the above-referenced litigation.

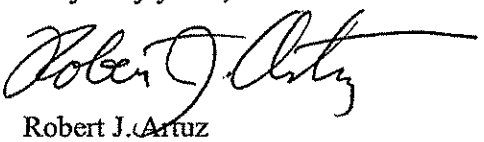
Below is a list of individuals that Oracle would like to depose during the next several weeks. It is our understanding that Quinstreet also intends to depose these individuals.

Keith Lowery
Terry Fokas
B. Kent Hill
John Furguson
Ronald Howell
Yoshi Noguchi
Brad Carl
Jim Salter
Sharmini Green

Because we expect that epicRealm's attorneys will be representing all or most of these individuals, we would like to coordinate with you and Quinstreet's counsel for the service of subpoenas upon them (for non-party witnesses) and for the scheduling of their depositions. Accordingly, for the witnesses that epicRealm's counsel intends to represent for their depositions, please confirm that you will accept service of any subpoenas. If epicRealm's counsel is not representing any of the above individuals, please let us know promptly so that we may contact those individuals separately.

If you have any questions or concerns, feel free to contact me at: 650-462-5336.

Very truly yours,



Robert J. Artuz

rja

cc: Patrick Patras
David R. Bennett
David Doyle
Robert Rigg
Theodore T. Herhold
Eric M. Hutchins

Exhibit 23

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

ORACLE CORPORATION and ORACLE U.S.A. INC.,	)	
Plaintiffs,	)	
v.	)	C.A. No. 06-414 (SLR)
EPICREALM LICENSING, LP,	)	
Defendant.	)	

PLAINTIFFS' SUPPLEMENTAL RESPONSE TO DEFENDANTS' INTERROGATORY NO. 3

In accordance with Rules 26 and 33 of the Federal Rules of Civil Procedure, Plaintiffs Oracle Corporation and Oracle U.S.A., Inc. ("Oracle") serve the following objections and responses to Defendant and Counterclaimant epicRealm Licensing, LP's ("epicRealm") First Set of Interrogatories (1 - 6). Oracle reserves the right to modify or supplement these responses in accordance with Federal Rules of Civil Procedure 26 and 33, as necessary.

OBJECTIONS AND RESPONSES

Oracle incorporates by reference herein the General and Specific Objections set forth in its initial Responses to Defendant's First Set of Interrogatories (1-6).

INTERROGATORY NO. 3:

For each claim of the epicRealm Patents that Oracle contends is not valid, identify every prior art reference or other information that Oracle contends will render any claim of the epicRealm Patents invalid and explain using claim charts how that reference and other information allegedly incorporates, performs, discloses and/or teaches each element of or the combination of elements of each such claim of the epicRealm Patents. The answer to this interrogatory should include but is not limited to providing the complete basis for Oracle's contention that "each claim of the epicRealm

Patents is invalid for failure to meet one or more conditions of patentability specified in 35 U.S.C. §§101, 102, 103 and/or 112."

FIRST SUPPLEMENTAL RESPONSE NO. 3:

Subject to and without waiving its objections, Oracle responds that its current positions are based on the information available to it to date. epicRealm has yet to provide detailed infringement contentions or proposed claim constructions to Oracle. Based on the positions taken by epicRealm in the ongoing Texas Actions and other information available to Oracle, Oracle has a good faith belief that the patents-in-suit are invalid. Oracle's discovery and investigation in connection with this lawsuit are continuing. For example, Oracle's investigation regarding the specific grounds of invalidity including anticipation, public use or on-sale bar under 35 U.S.C. § 102(b), improper inventorship, derivation under 35 U.S.C. § 102(f), obviousness, and failure to comply with 35 U.S.C. § 112, is ongoing. Oracle specifically reserves the right to modify or supplement these responses as additional information becomes available, and as its discovery and investigation proceed. For example, Oracle will provide detailed invalidity contentions in its expert reports pursuant to the Court's scheduling order.

Oracle's present invalidity positions are based in whole or in part on its present understanding of epicRealm's positions concerning the scope and construction of its asserted claims to the extent those positions can be deduced from epicRealm's conclusory infringement contentions in this case and in the Texas Actions. Oracle reserves the right to supplement its invalidity positions if and when epicRealm amends or clarifies its infringement contentions. Oracle's responses herein shall not be deemed adoptions or admissions as to the accuracy of any claim constructions adopted by

epicRealm or any third party to date. Oracle reserves the right to modify or supplement it positions in response to the Court's claim construction.

References cited in Exhibits 3-4 (invalidity charts) disclose the elements of the asserted claims either explicitly or inherently and may be relied upon to show the state of the art in the relevant timeframes. The suggested obviousness combinations are in the alternative to Oracle's anticipation contentions and are not to be construed to suggest that any reference included in the combinations is not anticipatory.

Subject to and without waiving its objections, Oracle further responds as follows:

A. Invalidity Under 35 U.S.C. §§ 102 and 103

All of the claims of epicRealm's patents-in-suit are invalid because each fails to meet one or more of the requirements of patentability under 35 U.S.C. §§ 102 and 103. Because only claims 1-11 of the '554 patent and claims 2, 16 and 29 of the '335 patent are asserted in this lawsuit, Oracle has limited its preliminary invalidity contentions to those asserted claims. Oracle reserves the right to supplement its invalidity positions in the event that epicRealm is permitted to assert and asserts additional patent claims. The prior art references that provide the bases for the invalidity of the patents-in-suit are listed in the following paragraphs. Each of these listed prior art documents, the underlying work and/or the underlying apparatus described in each document qualifies as prior art under one or more sections of 35 U.S.C. § 102. Further, any document indicated as anticipating a claim also renders the claim obvious when viewed either alone or in combination with other prior art references.

The following chart lists prior art references including patents and publications that render one or more claims of the patents-in-suit invalid as anticipated and/or obvious.

Prior Art Reference	***Claims Invalidated – '554 Patent***	***Claims Invalidated – '335 Patent***
U.S. Patent No. 5,701,451	1-6, 9-11 (§ 102) 7, 8 (§ 103)	2, 16, 29 (§ 102)
U.S. Patent No. 5,742,762	1-6, 9-11 (§102) 7, 8 (§ 103)	2, 16, 29 (§ 102)
U.S. Patent No. 6,249,291	1-5, 7-11 (§ 102) 4, 6 (§ 103)	2, 16, 29 (§ 102)
K. Antchev et. al, *A WWW Learning Environment for Mathematics*, Proceedings of the Fourth International World Wide Wed Conference (December 12, 1995)	1-4, 6-11 (§ 102) 5 (§ 103)	2, 16, 29 (§ 102)
C. Derler, *The World-Wide Web Gateway to Hyper-G: Using a Connectionless Protocol to Access Session-Oriented Services,* Institut for Informationsverarbeitung and Computergestutzte nue Medien (March 1995)	1-11 (§ 102)	2, 16, 29 (§ 102)
J. Gaffney, *Illustra's Web DataBlade Module: An Illustra Technical White Paper*, SIGMOD Record, Vol. 25, No. 1 (March 1996)	1-5, 7-11 (§ 102) 6 (§ 103)	2, 16, 29 (§ 102)
M. Garland et al., *Implementing Distributed Server Groups for the World Wide Web* (January 25, 1995)	1-11 (§ 103)	2, 16, 29 (§ 103)
E. Hastings & K. Dumar, *Providing Customers Information Using the Web and CORBA*, Proceedings of the Second International World Wide Web Conference (1994)	1-3, 5, 7-11 (§ 102) 4, 6 (§ 103)	2, 16, 29 (§ 102)
C. Lagoze et. al, *Dienst: Implementation and Reference Manual*, Cornell University Technical Report TR95-1514 (May 5, 1995)	1-11 (§ 102) 6 (§ 103)	2, 16, 29 (§ 102) 29 (§ 103)
Oracle Corporation, *Oracle WebServer User's Guide* (Release 1.0) (1995)	1-11 (§ 102) 7, 8 (§ 103)	2, 16, 29 (§ 102)
Oracle Corporation, *Oracle and the Internet* (July 1995)	1-11 (§ 103)	2, 16, 29 (§ 103)

Prior Art Reference	***Claims Invalidated – '554 Patent***	***Claims Invalidated – '335 Patent***
Oracle Corporation, *Oracle Internet Release 1.0, Bringing the power of Oracle7 to the World Wide Web* (July 1995)	1-11 (§ 102) 6-8 (§ 103)	2, 16, 29 (§ 102)
Oracle Corporation, *Design Specification for Oracle Web Agent and OWS Developer's Toolkit, Oracle Web Server V2.0* (December 15, 1995)	1-11 (§ 103)	2, 16, 29 (§ 103)
B.F. Rasmussen, *WDB – A Web Interface to Sybase*, Astronomical Data Analysis Software and Systems IV, ASP Conference Series, Vol. 77 (1995)	1-3, 7-11 (§ 102) 4-8 (§ 103)	2, 16 (§ 102) 29 (§ 103)
L. Shklar *et. al*, *The "InfoHarness" Information Integration Platform*, Proceedings of the Second International World Wide Web Conference (1994)	1-3, 9-11 (§ 102) 4-8 (§ 103)	2, 16, 29 (§ 102)
M. Sjolin, *A WWW Front End to an OODBMS*, Proceedings of the Second International World Wide Web Conference (1994)	1-5, 7-10 (§ 102) 2, 6 (§ 103)	2, 16, 29 (§ 102)
C. Varela *et. al*, *DB: Browsing Object Oriented Databases over the Web*, Proceedings of the Fourth International World Wide Web Conference (December 12, 1995)	1-6, 9-11 (§ 102) 5, 7, 8 (§ 103)	2, 16, 29 (§ 102)
Japanese Patent Pub. No. 04-271454	One or more claims[1]	One or more claims
Japanese Patent Pub. No. 07-064893	One or more claims	One or more claims
CERN Web Server as represented in at least: Susana Fernández Vega and Jean-Yves Le Meur, "*Collecting and Serving Information within a Large Organization*," CERN, European Laboratory for Particle Physics (April 22, 1996)	1-3, 5-6, 9-11 (§ 102) 4, 7 (§ 103)	2, 16, 29 (§ 102)

[1] Oracle is in the process of obtaining translations of Japanese Patent Pub. Nos. 04-271454 and 07-064893 and will timely supplement its response to Interrogatory No. 3 as necessary after the translations have been obtained an analyzed.

Prior Art Reference	***Claims Invalidated – '554 Patent***	***Claims Invalidated – '335 Patent***
The Harvest Discovery and Access System as represented in at least: C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, *Harvest: A Scalable, Customizable Discovery and Access System*, University of Colorado - Boulder Technical report CU-CS-732-94 (March 1995)	1-3, 6, 9-11 (§ 102) 4-5, 7 (§ 103)	2, 16, 29 (§ 102)
Brian Baker, "*Doing business with the Web: The Informix/Illustra Approach*," 5th International conference on extending database technology (March, 1996)	1-4, 7-11 (§ 102) 5 (§ 103)	2, 16, 29 (§ 102)
Dan Mosedale, William Foss and Rob McCool, "*Administering Very High Volume Internet Services*," Ninth System Administration Conference (LISA '95) (September 18-22, 1995)	1-3, 6, 9-11 (§ 102) 4, 7 (§ 103)	2, 16, 29 (§ 102)
The SWEB System as represented in at least: Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "*SWEB: Towards a Scalable World Wide Web Server on Multicomputers*," Department of Computer Science, University of California (1996)	1, 3, 9-11 (§ 102) 2, 4-8 (§ 103)	2, 16, 29 (§ 102)

Table 1

Exhibits 3-4 identify specific examples of where each limitation of the asserted claims is found in these references. These examples are intended to be illustrative only and do not comprise a complete list.

To the extent that any of the asserted claims are not invalid under § 102 as anticipated by one or more of the references above, several different combinations of prior art references render each asserted claim obvious and unpatentable under § 103. As Exhibits 3-4 illustrate, each of the asserted claims merely combine concepts and features

that were well-known in the prior art, and all of the prior art references listed in Table 1 all are directed to a common need and problem known in the field of the asserted claims, the efficient management of, and response to, requests for dynamically-generated web pages. The asserted claims recite solutions to this common problem; to one of ordinary skill in the art, these solutions would have been obvious to try, with predictable results. Hence, the interrelated teachings of the references in Table 1, along with other prior art references provide several reasons for combining prior art references in the manner recited by the asserted claims. Such other prior art references include, but are not limited to, prior art references disclosed in the Texas Actions and all pending reexamination proceedings of the '554 and '335 patents, including without limitation, U.S. Patent and Trademark Office Reexamination Control Nos. 90/008,342, 90/008,343, 90/008,562, 90/008,568, 90/008,574 and 90/008,584 (collectively, the "Reexaminations").[2]

Merely by way of example, to the extent that any of the references in Table 1 might be interpreted as failing to disclose "releasing [a] Web server to process other requests," as recited by claim 1 of the '335 patent and claim 1 of the '554 patent, "releasing [a] second computer system to process other requests," as recited by claim 9 of the '554 patent, or "releasing [a] HTTP-compliant device to process other requests," as recited by claims 15 and 29 of the '335 patent; L. Stein, *How To Set Up And Maintain A*

[2] Oracle incorporates by reference into its response to Interrogatory No. 3 all of the prior art references disclosed in the Texas Actions and the Reexaminations, and asserts that some or all of those prior art references further invalidate (or may be combined with the prior art references described in Table 1 to invalidate) the patents-in-suit under 35 U.S.C. §§ 102 and/or 103. Other than as expressly stated herein, however, Oracle does not adopt or admit to any assertions or arguments made in the Texas Actions. Similarly, by referencing the Reexaminations, Oracle is not necessarily adopting all of the arguments contained therein.

World Wide Web Site (1995) (disclosed in the Texas Actions) teaches such releasing of a web server through a technique of closing CGI process file handles, which frees the HTTP server to process other requests by closing connections to the web server. It would be obvious to use this technique to release an HTTP server to process other requests, because it was well known in the prior art (as Stein discloses) that CGI page generation techniques (including without limitation techniques that require database access) could be associated with lengthy delays. To avoid these delays, one of ordinary skill in the art would have been motivated to combine Stein's releasing techniques with any of the systems and methods disclosed by the references in Table 1.

As another example, to the extent that any of the references in Table 1 fail to disclose "maintaining a connection cache to . . . one or more data sources," as recited by claim 4 of the '554 patent, "logging into . . . one or more data sources," as recited by claim 5 of the '554 patent, or "maintaining a page cache containing [a] Web page," as recited by claim 6 of the '554 patent, such features were well-known in the software arts for many years prior to the purported invention of the '554 and '335 patents, as evidenced by the multiple references (including without limitation, Web DataBlade, U.S. Patent 6,249,291, Derler and Lagoze) that disclose such features. It would be obvious to add such features to the systems and methods of any of the above references to solve well-known problems in the field of responding to requests for dynamically-generated web pages.

With regard to page caching, for example, Derler notes that: "Caching is a technique often used to improve performance of an information system. This is why in the Hyper-G system, requests from clients rather than being directly sent to document

servers, are routed through a document cache." *Derler*, § 3.3.3. It would have been obvious to one of skill in the art to implement a page caching technique in any of the references listed above, in order to improve performance of the web server in responding to requests.

Additionally, a motivation to combine the art is explicit or inherent in the references cited and also would have been known to one of ordinary skill in the art. For example, many of the references cite one another or are related. One of ordinary skill in the art would also have been motivated to combine the references because they are all directed to the same concept of efficient managing of, and responding to, requests for dynamically-generated web pages. Finally, each of the anticipatory references cited in Exhibits 3-4 serves as a suggestion to combine.

The prior art references identified herein provide evidence that the claimed subject matter was known or used by others in this country prior to the '554 patent's earliest priority date, or was publicly used, sold or offered for sale in this country before the alleged invention date or more than one year prior to the filing date. In particular, the actual products described in these prior art references are themselves prior art against the asserted patent claims. At least claims 1-11 of the '554 patent and claims 2, 16 and 29 of the '335 patent are invalid in view of the prior invention, use, sale and/or offer for sale of these products under 35 U.S.C. §§ 102(a), (b) and/or (g).

Additionally, Oracle believes that the invention, use, sale and/or offer for sale of the following products and activities may also constitute prior art and render one or more asserted claims invalid under 35 U.S.C. §§ 102(a), (b), (g) and/or 103:

- Oracle Web Server v.2.0-2.1

- Apache v.0.1-1.0.5
- NCSA httpd
- Netscape Netsite
- Squid
- Plexis
- XEROX PARC Map Server
- Next WebObjects
- Microsoft Internet Information Server (IIS)
- Virtual Frog Dissection Kit/Whole Frog Project
- Zelig
- LINCKS

Oracle is currently in the process of discovering more information about these products and activities and will amend or supplement its response to this interrogatory accordingly after it has had sufficient time to do an adequate invalidity analysis.

B. Invalidity Under 35 U.S.C. § 112 - Enablement, Written Description and Definiteness

All of the asserted claims are invalid under 35 U.S.C. § 112(1) for failing to comply with the enablement and written description requirements imposed by that statute. All of the asserted claims use one or more of the terms, "release," "intercept," "dispatch" (or variations thereof), yet the specification and figures of the '554 and '335 patents fail to provide sufficient disclosure either to indicate that the inventor was in possession of the invention at the time of filing or to enable one of skill in the art to practice the purported invention without undue experimentation.

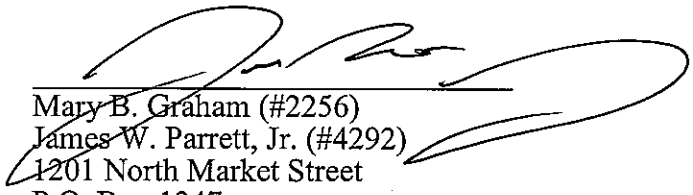
All of the asserted claims are also invalid for indefiniteness under 35 U.S.C. § 112(2). The asserted claims fail to particularly point out and distinctly claim the subject matter that the patent applicant regards as his or her invention because the use of the terms "release," "intercept," "dispatch" (and their variants) render the asserted claims unclear and indefinite when construed in light of the specification. Further, the use of the term "connection cache" renders claim 4 of the '554 patent indefinite when construed in light of the specification.

Claim 9 of the '554 patent is invalid under § 112(2) also because it is directed to an apparatus but it recites method steps (i.e., "wherein said routing further includes intercepting said request at said second computer, routing said request from said second computer to a dispatcher, and dispatching said request"). In addition, claim 9 is invalid under § 112(2),(6) because it recites several means plus function elements (i.e., "a page server having a processing means . . . a first computer system including means for generating said request . . . a second computer system including means for receiving said request . . . [and] said page server processing means processing said request), but the specification of the '554 patent provides no corresponding structure clearly linked to these elements, rendering them indefinite. Claim 10 of the '554 patent, which depends from claim 9, necessarily includes the limitations of claim 9 and therefore is invalid on the same grounds.

Depending on how the claims are construed, additional claims of the patents-in-suit also may be invalid for indefiniteness, lack of enablement, and/or lack of sufficient written description with respect to other claim limitations.

Oracle reserves the right to modify or supplement this response in accordance with Federal Rules of Civil Procedure 26 and 33, as necessary.

MORRIS, NICHOLS, ARSHT & TUNNELL LLP



Mary B. Graham (#2256)
James W. Parrett, Jr. (#4292)
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
302.658.9200

Attorneys for Oracle Corporation and Oracle U.S.A. Inc.

OF COUNSEL:

James G. Gilliland
Theodore T. Herhold
Chad E. King
Robert J. Artuz
Eric M. Hutchins
TOWNSEND AND TOWNSEND AND CREW LLP
379 Lytton Avenue
Palo Alto, CA 94301
650.326.2400

Dorian Daley
Peggy E. Bruggman
Matthew M. Sarboraria
ORACLE CORPORATION
ORACLE U.S.A., INC.
500 Oracle Parkway
Redwood Shores, CA 94065

Dated: January 14, 2008

1383281

CERTIFICATE OF SERVICE

I hereby certify that on January 14, 2008, true and correct copies of the foregoing were caused to be served upon the following individuals in the manner indicated:

BY E-MAIL AND HAND DELIVERY

Richard L. Horwitz
David Ellis Moore
POTTER ANDERSON & CORROON, LLP
1313 N. Market St., Hercules Plaza, 6th Floor
P.O. Box 951
Wilmington, DE 19899-0951

rhorwitz@potteranderson.com
dmoore@potteranderson.com

BY E-MAIL

George S. Bosy, Esquire
JENNER & BLOCK
330 N. Wabash Avenue
Chicago, IL 60611-7603

gbosy@jenner.com

James W. Parrett, Jr. (#4292)

Exhibit 3P – Oracle's Supplemental Response to epicRealm's Interrogatory No. 3
U.S. Patent No. 5,894,554
Prior Art Reference: CERN (1996)

USP 5,894,554 – Claim 1	CERN (Vega) System
A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 1.
routing said request from said Web server to a page server,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, pp. 6-7.
said page server receiving said request and releasing said Web server to process other requests,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
wherein said routing step further includes the steps of	
intercepting said request at said Web server	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
routing said request from said Web server to a dispatcher, and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
dispatching said request to said page server;	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
dynamically generating a Web page in response to said request, said Web page including data	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 9.

USP 5,894,554 – Claim 1	CERN (Vega) System
dynamically retrieved from one or more data sources.	

USP 5,894,554 – Claim 2	CERN (Vega) System
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 10.

USP 5,894,554 – Claim 3	CERN (Vega) System
The computer-implemented method in claim 2 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 9.

USP 5,894,554 – Claim 4	CERN (Vega) System
The computer-implemented method in claim 3 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 5,894,554 – Claim 5	CERN (Vega) System
The computer-implemented method in claim 3 wherein said step of processing said request includes the step of logging into said one or more data sources.	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 10.

USP 5,894,554 – Claim 6	CERN (Vega) System
The computer-implemented method in claim 3 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 11.

USP 5,894,554 – Claim 7	CERN (Vega) System
The computer-implemented method in claim 3 wherein said page server includes custom HTML extension templates for configuring said Web page.	

USP 5,894,554 – Claim 8	CERN (Vega) System
The computer-implemented method in claim 7 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said custom HTML extension	

USP 5,894,554 – Claim 8	CERN (Vega) System
templates.	

USP 5,894,554 – Claim 9	CERN (Vega) System
A networked system for managing a dynamic Web page generation request, said system comprising:	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 1.
one or more data sources;	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 9.
a page server having a processing means;	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
a first computer system including means for generating said request; and	
a second computer system including means for receiving said request from said first computer,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, pp. 6-7.
said second computer system also including a router, said router routing said request from said second computer system to said page server,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
wherein said routing further includes	
intercepting said request at said second computer,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
routing said request from said second computer to a dispatcher, and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.

USP 5,894,554 – Claim 9	CERN (Vega) System
dispatching said request to said page server	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
said page server receiving said request and releasing said second computer system to process other requests,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
said page server processing means processing said request and dynamically generating a Web page in response to said request,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
said Web page including data dynamically retrieved from said one or more data sources.	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 9.

USP 5,894,554 – Claim 10	CERN (Vega) System
The networked system in claim 9 wherein said router in said second computer system includes:	
an interceptor intercepting said request at said second computer system and routing said request; and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, pp. 6-7.
a dispatcher receiving said routed request from said interceptor and dispatching said request to said page server.	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.

USP 5,894,554 – Claim 11	CERN (Vega) System

USP 5,894,554 – Claim 11	CERN (Vega) System
A machine readable medium having stored thereon data representing sequences of instructions, which when executed by a computer system, cause said computer system to perform the steps of:	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 1.
routing a dynamic Web page generation request from a Web server to a page server,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, pp. 6-7.
said page server receiving said request and releasing said Web server to process other requests	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
wherein said routing step further includes the steps of	
intercepting said request at said Web server,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
routing said request from said Web server to a dispatcher,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
and dispatching said request to said page server;	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
dynamically generating a Web page, said Web page including data	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 9.

USP 5,894,554 – Claim 11	CERN (Vega) System
retrieved from one or more data sources.	

61224530 v2

Exhibit 3Q – Oracle's Supplemental Response to epicRealm's Interrogatory No. 3
U.S. Patent No. 5,894,554
Prior Art Referemce: Harvest Ingest System (1994)

USP 5,894,554 – Claim 1	Harvest Ingest System
A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, March 1995 at pp. 1, 6. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 2. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at pp. 62-63.
routing said request from said Web server to a page server,	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10.
said page server receiving said request and releasing said Web server to process other requests,	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 6. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
wherein said routing step further includes the steps of	
intercepting said request at said Web server	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
routing said request from said Web server to a dispatcher, and	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado -

USP 5,894,554 – Claim 1	Harvest Ingest System
	Boulder Technical report CU-CS-732-94, March 1995 at p. 1. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
dispatching said request to said page server;	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at pp. 10 and 13.
dynamically generating a Web page in response to said request, said Web page including data dynamically retrieved from one or more data sources.	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at pp. 2-3 and 5.

USP 5,894,554 – Claim 2	Harvest Ingest System
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at pp. 3 and 35.

USP 5,894,554 – Claim 2	Harvest Ingest System
sources from which to retrieve said data.	

USP 5,894,554 – Claim 3	Harvest Ingest System
The computer-implemented method in claim 2 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 3.

USP 5,894,554 – Claim 4	Harvest Ingest System
The computer-implemented method in claim 3 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 5,894,554 – Claim 5	Harvest Ingest System
The computer-implemented method in claim 3 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 5,894,554 – Claim 6	Harvest Ingest System

USP 5,894,554 – Claim 6	Harvest Ingest System
The computer-implemented method in claim 3 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, March 1995 at pp. 10 and 18.

USP 5,894,554 – Claim 7	Harvest Ingest System
The computer-implemented method in claim 3 wherein said page server includes custom HTML extension templates for configuring said Web page.	

USP 5,894,554 – Claim 8	Harvest Ingest System
The computer-implemented method in claim 7 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said custom HTML extension templates.	

USP 5,894,554 – Claim 9	Harvest Ingest System
A networked system for managing a dynamic Web page generation request, said system comprising:	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, March 1995 at p. 1. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder

USP 5,894,554 – Claim 9	Harvest Ingest System
	Technical report CU-CS-743-94, October 1994 at p. 2. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995 at pp. 62-63. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 6.
one or more data sources;	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at pp. 2, 3 and 5.
a page server having a processing means;	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at pp. 10 and 13.
a first computer system including means for generating said request; and	
a second computer system including means for receiving said request from said first computer,	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10.
said second computer system also including a router, said router routing	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.

USP 5,894,554 – Claim 9	Harvest Ingest System
said request from said second computer system to said page server,	
wherein said routing further includes	
intercepting said request at said second computer,	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
routing said request from said second computer to a dispatcher, and	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, March 1995 at p. 1. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10.
dispatching said request to said page server	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
said page server receiving said request and releasing said second computer system to process other requests,	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
said page server processing means processing said request and dynamically generating a Web page in response to said request,	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at pp. 10 and 13.
said Web page including data dynamically retrieved from said one or	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report

USP 5,894,554 – Claim 9	Harvest Ingest System
more data sources.	CU-CS-732-94, July 1994 at p. 10. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at pp. 2, 3, and 5.

USP 5,894,554 – Claim 10	Harvest Ingest System
The networked system in claim 9 wherein said router in said second computer system includes:	
an interceptor intercepting said request at said second computer system and routing said request; and	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
a dispatcher receiving said routed request from said interceptor and dispatching said request to said page server.	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.

USP 5,894,554 – Claim 11	Harvest Ingest System
A machine readable medium having stored thereon data representing sequences of instructions, which when executed by a computer system, cause said computer system to perform the steps of:	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, March 1995 at p. 1. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 2. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995 at pp. 62-63.

USP 5,894,554 – Claim 11	Harvest Ingest System
	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 6.
routing a dynamic Web page generation request from a Web server to a page server,	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995 at p. 64. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10.
said page server receiving said request and releasing said Web server to process other requests	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 6. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
wherein said routing step further includes the steps of	
intercepting said request at said Web server,	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
routing said request from said Web server to a dispatcher,	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, March 1995 at p. 1. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995 at p. 64. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A

USP 5,894,554 – Claim 11	Harvest Ingest System
	Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10.
and dispatching said request to said page server;	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995 at p. 64.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at pp. 10-13.
dynamically generating a Web page, said Web page including data retrieved from one or more data sources.	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at pp. 2 and 5.

61198508 v1

Exhibit 3R – Oracle's Supplemental Response to epicRealm's Interrogatory No. 3
U.S. Patent No. 5,894,554
Prior Art Reference: Web DataBlade System (1996)

USP 5,894,554 – Claim 1	Web DataBlade System
A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 364, 366-367.
routing said request from said Web server to a page server,	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.
said page server receiving said request and releasing said Web server to process other requests,	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.
wherein said routing step further includes the steps of	
intercepting said request at said Web server	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 367.
routing said request from said Web server to a dispatcher, and	Brian Baker, "Doing Business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 367.
dispatching said request to said page server;	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.
dynamically generating a Web page in response to said request, said Web page including data	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.

USP 5,894,554 – Claim 1	Web DataBlade System
dynamically retrieved from one or more data sources.	

USP 5,894,554 – Claim 2	Web DataBlade System
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 365.

USP 5,894,554 – Claim 3	Web DataBlade System
The computer-implemented method in claim 2 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 367.

USP 5,894,554 – Claim 4	Web DataBlade System
The computer-implemented method in claim 3 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 367.

USP 5,894,554 – Claim 5	Web DataBlade System
The computer-implemented method in claim 3 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 5,894,554 – Claim 6	Web DataBlade System
The computer-implemented method in claim 3 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	

USP 5,894,554 – Claim 7	Web DataBlade System
The computer-implemented method in claim 3 wherein said page server includes custom HTML extension templates for configuring said Web page.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March at 1996, p. 366.

USP 5,894,554 – Claim 8	Web DataBlade System
The computer-implemented method in claim 7 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said custom HTML extension templates.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 366.

USP 5,894,554 – Claim 9	Web DataBlade System
A networked system for managing a dynamic Web page generation request, said system comprising:	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 364, 366-367.
one or more data sources;	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.
a page server having a processing means;	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 364, 366-367.
a first computer system including means for generating said request; and	
a second computer system including means for receiving said request from said first computer,	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 367.
said second computer system also including a router, said router routing said request from said second computer system to said page server,	INFORMIX®, Informix Web DataBlade Module User's Guide, Version 3.3, May 1997, pp. 1-5.
wherein said routing further includes	
intercepting said request at said second computer,	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 367.
routing said request from said second computer to a dispatcher, and	Brian Baker, "Doing Business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 367.
dispatching said request to said page server	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.
said page server receiving said	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference

USP 5,894,554 – Claim 9	Web DataBlade System
request and releasing said second computer system to process other requests,	on extending database technology, March, 1996 at pp. 366-367.
said page server processing means processing said request and dynamically generating a Web page in response to said request,	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.
said Web page including data dynamically retrieved from said one or more data sources.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.

USP 5,894,554 – Claim 10	Web DataBlade System
The networked system in claim 9 wherein said router in said second computer system includes:	
an interceptor intercepting said request at said second computer system and routing said request; and	Brian Baker, "Doing Business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 367.
a dispatcher receiving said routed request from said interceptor and dispatching said request to said page server.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.

USP 5,894,554 – Claim 11	Web DataBlade System
A machine readable medium having stored thereon data representing sequences of instructions, which when executed by a computer system, cause said computer system to perform the	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 364, 366-367.

USP 5,894,554 – Claim 11	Web DataBlade System	
steps of:		
routing a dynamic Web page generation request from a Web server to a page server,	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 366-367.	
said page server receiving said request and releasing said Web server to process other requests	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 366-367.	
wherein said routing step further includes the steps of		
intercepting said request at said Web server,	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996, at p. 367.	
routing said request from said Web server to a dispatcher,	Brian Baker, "Doing Business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at p. 367.	
and dispatching said request to said page server;	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.	
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.	
dynamically generating a Web page, said Web page including data retrieved from one or more data sources.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at pp. 366-367.	

Exhibit 3S – Oracle's Supplemental Response to epicRealm's Interrogatory No. 3
U.S. Patent No. 5,894,554
Prior Art Reference: Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.

USP 5,894,554 – Claim 1	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
routing said request from said Web server to a page server,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
said page server receiving said request and releasing said Web server to process other requests,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
wherein said routing step further includes the steps of	
intercepting said request at said Web server	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
routing said request from said Web server to a dispatcher, and	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
dispatching said request to said page server;	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
dynamically generating a Web page in response to said request, said	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 100.

USP 5,894,554 – Claim 1	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
Web page including data dynamically retrieved from one or more data sources.	

USP 5,894,554 – Claim 2	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 100.

USP 5,894,554 – Claim 3	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
The computer-implemented method in claim 2 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 100.

USP 5,894,554 – Claim 4	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
The computer-implemented method in claim 3 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 5,894,554 – Claim 5	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
The computer-implemented method in claim 3 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 5,894,554 – Claim 6	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
The computer-implemented method in claim 3 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 99.

USP 5,894,554 – Claim 7	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
The computer-implemented method in claim 3 wherein said page server includes custom HTML extension templates for configuring said Web page.	

USP 5,894,554 – Claim 8	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
The computer-implemented method in claim 7 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources	

USP 5,894,554 – Claim 8	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
into said custom HTML extension templates.	

USP 5,894,554 – Claim 9	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
A networked system for managing a dynamic Web page generation request, said system comprising:	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
one or more data sources;	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
a page server having a processing means;	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
a first computer system including means for generating said request; and	
a second computer system including means for receiving said request from said first computer,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
said second computer system also including a router, said router routing said request from said second computer system to said page server,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
wherein said routing further includes	
intercepting said request at said second computer,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
routing said request from said second computer to a dispatcher, and	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.

USP 5,894,554 – Claim 9	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
dispatching said request to said page server	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
said page server receiving said request and releasing said second computer system to process other requests,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
said page server processing means processing said request and dynamically generating a Web page in response to said request,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
said Web page including data dynamically retrieved from said one or more data sources.	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 100.

USP 5,894,554 – Claim 10	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
The networked system in claim 9 wherein said router in said second computer system includes:	
an interceptor intercepting said request at said second computer system and routing said request; and	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
a dispatcher receiving said routed request from said interceptor and dispatching said request to said page	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.

USP 5,894,554 – Claim 10	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
server.	

USP 5,894,554 – Claim 11	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
A machine readable medium having stored thereon data representing sequences of instructions, which when executed by a computer system, cause said computer system to perform the steps of:	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
routing a dynamic Web page generation request from a Web server to a page server,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
said page server receiving said request and releasing said Web server to process other requests	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
wherein said routing step further includes the steps of	
intercepting said request at said Web server,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
routing said request from said Web server to a dispatcher,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
and dispatching said request to said page server;	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
processing said request, said processing being performed by said page server while said Web server	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.

USP 5,894,554 – Claim 11	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95
concurrently processes said other requests; and	
dynamically generating a Web page, said Web page including data retrieved from one or more data sources.	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 100.

61213456 v3

Exhibit 3T – Oracle's Supplemental Response to epicRealm's Interrogatory No. 3
U.S. Patent No. 5,894,554
Prior Art Reference: SWEB System (1996)

USP 5,894,554 – Claim 1	SWEB System
A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 850.
routing said request from said Web server to a page server,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 852.
said page server receiving said request and releasing said Web server to process other requests,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, pp. 850-851.
wherein said routing step further includes the steps of	
intercepting said request at said Web server	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
routing said request from said Web server to a dispatcher, and	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
dispatching said request to said page server;	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
dynamically generating a Web page in response to said request, said	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.

USP 5,894,554 – Claim 1	SWEB System
Web page including data dynamically retrieved from one or more data sources.	

USP 5,894,554 – Claim 2	SWEB System
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	

USP 5,894,554 – Claim 3	SWEB System
The computer-implemented method in claim 2 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.

USP 5,894,554 – Claim 4	SWEB System
The computer-implemented method in claim 3 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 5,894,554 – Claim 5	SWEB System
The computer-implemented method in claim 3 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 5,894,554 – Claim 6	SWEB System
The computer-implemented method in claim 3 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	

USP 5,894,554 – Claim 7	SWEB System
The computer-implemented method in claim 3 wherein said page server includes custom HTML extension templates for configuring said Web page.	

USP 5,894,554 – Claim 8	SWEB System
The computer-implemented method in claim 7 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said custom HTML extension templates.	

USP 5,894,554 – Claim 9	SWEB System
A networked system for managing a dynamic Web page generation request, said system comprising:	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 850.
one or more data sources;	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 851.
a page server having a processing means;	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
a first computer system including means for generating said request; and	
a second computer system including means for receiving said request from said first computer,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
said second computer system also including a router, said router routing said request from said second computer system to said page server,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
wherein said routing further includes	
intercepting said request at said second computer,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
routing said request from said second computer to a dispatcher, and	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
dispatching said request to said page server	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
said page server receiving said	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide

USP 5,894,554 – Claim 9	SWEB System
request and releasing said second computer system to process other requests,	Web Server on Multicomputers," Department of Computer Science, University of California, 1996, pp. 850-851.
said page server processing means processing said request and dynamically generating a Web page in response to said request,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
said Web page including data dynamically retrieved from said one or more data sources.	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.

USP 5,894,554 – Claim 10	SWEB System
The networked system in claim 9 wherein said router in said second computer system includes:	
an interceptor intercepting said request at said second computer system and routing said request; and	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 852.
a dispatcher receiving said routed request from said interceptor and dispatching said request to said page server.	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.

USP 5,894,554 – Claim 11	SWEB System
A machine readable medium having stored thereon data representing sequences of instructions, which when executed by a computer system, cause said computer system to perform the	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 850.

USP 5,894,554 – Claim 11	SWEB System
steps of:	
routing a dynamic Web page generation request from a Web server to a page server,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 852.
said page server receiving said request and releasing said Web server to process other requests	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, pp. 850-851.
wherein said routing step further includes the steps of	
intercepting said request at said Web server,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
routing said request from said Web server to a dispatcher,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
and dispatching said request to said page server;	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
dynamically generating a Web page, said Web page including data retrieved from one or more data sources.	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.

Exhibit 4P – Oracle's Supplemental Response to epicRealm's Interrogatory No. 3
U.S. Patent No. 6,415,335
Prior Art Reference: *CERN (1996)*

USP 6,415,335 – Claim 1	CERN (Vega) System
A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 1.
routing a request from a Web server to a page server,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, pp. 6-7. Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, Fig. 4.
said page server receiving said request and releasing said Web server to process other requests	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
wherein said routing step further includes the steps of:	
intercepting said request at said Web server and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
routing said request to said page server;	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
dynamically generating a Web page in response to said request, said Web page including data dynamically retrieved from one or more data sources.	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 9. Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, Fig. 6.

USP 6,415,335 – Claim 1	CERN (Vega) System

USP 6,415,335 – Claim 2	CERN (Vega) System
The computer-implemented method in claim 1 wherein said step of routing said request includes the steps of:	
routing said request from said Web server to a dispatcher; and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
dispatching said request to said page server.	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.

USP 6,415,335 – Claim 3	CERN (Vega) System
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	

USP 6,415,335 – Claim 4	CERN (Vega) System
The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or	

USP 6,415,335 – Claim 4	CERN (Vega) System
more data sources.	

USP 6,415,335 – Claim 5	CERN (Vega) System
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 6,415,335 – Claim 6	CERN (Vega) System
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 6,415,335 – Claim 7	CERN (Vega) System
The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	

USP 6,415,335 – Claim 8	CERN (Vega) System
The computer-implemented method in claim 1 wherein said page server	

USP 6,415,335 – Claim 8	CERN (Vega) System
includes tag-based text templates for configuring said Web page.	

USP 6,415,335 – Claim 9	CERN (Vega) System
The computer-implemented method in claim 8 wherein said step of processing said request further includes the step of inserting said-dynamically retrieved data from said one or more data sources into said tag-based text templates.	

USP 6,415,335 – Claim 10	CERN (Vega) System
The computer-implemented method in claim 8 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request.	

USP 6,415,335 – Claim 11	CERN (Vega) System
The computer-implemented method in claim 8 wherein said tag-based text templates include HTML templates.	

USP 6,415,335 – Claim 12	CERN (Vega) System
The computer-implemented method in	

USP 6,415,335 – Claim 12	CERN (Vega) System
claim 1 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources.	

USP 6,415,335 – Claim 13	CERN (Vega) System
The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of processing an object handling extension.	

USP 6,415,335 – Claim 14	CERN (Vega) System
The computer-implemented method in claim 13 wherein said object handling extension is an OLE extension.	

USP 6,415,335 – Claim 15	v
A computer-implemented method comprising the steps of:	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 1.
transferring a request from an HTTP-compliant device to a page server,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, pp. 6-7. Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, Fig. 4.

USP 6,415,335 – Claim 15	v
said page server receiving said request and releasing said HTTP-compliant device to process other requests	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
wherein said transferring step further includes the steps of:	
intercepting said request at said HTTP-compliant device and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
transferring said request to said page server;	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
dynamically generating a page in response to said request, said page including data dynamically retrieved from one or more data sources.	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 9. Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, Fig. 6.

USP 6,415,335 – Claim 16	CERN (Vega) System
The computer-implemented method in claim 15 wherein said step of transferring said request includes the steps of:	

USP 6,415,335 – Claim 16	CERN (Vega) System
transferring said request from said HTTP-compliant device to a dispatcher; and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
dispatching said request to said page server.	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.

USP 6,415,335 – Claim 17	CERN (Vega) System
The computer-implemented method in claim 15 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	

USP 6,415,335 – Claim 18	CERN (Vega) System
The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of dynamically retrieving said data from said one or more data sources.	

USP 6,415,335 – Claim 19	CERN (Vega) System
The computer-implemented method in claim 15 wherein said step of processing said request includes the	

USP 6,415,335 – Claim 19	CERN (Vega) System
step of said page server maintaining a connection cache to said one or more data sources.	

USP 6,415,335 – Claim 20	CERN (Vega) System
The computer-implemented method in claim 15 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 6,415,335 – Claim 21	CERN (Vega) System
The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of maintaining a page cache containing said page.	

USP 6,415,335 – Claim 22	CERN (Vega) System
The computer-implemented method in claim 15 wherein said page server includes tag-based text templates for configuring said page.	

USP 6,415,335 – Claim 23	CERN (Vega) System
The computer-implemented method in	

USP 6,415,335 – Claim 23	CERN (Vega) System
claim 22 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said tag-based text templates.	

USP 6,415,335 – Claim 24	CERN (Vega) System
The computer-implemented method in claim 22 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request.	

USP 6,415,335 – Claim 25	CERN (Vega) System
The computer-implemented method in claim 22 wherein said tag-based text templates include HTML templates.	

USP 6,415,335 – Claim 26	CERN (Vega) System
The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources.	

USP 6,415,335 – Claim 27	CERN (Vega) System
The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of processing an object handling extension.	

USP 6,415,335 – Claim 28	CERN (Vega) System
The computer-implemented method in claim 27 wherein said object handling extension is an OLE extension.	

USP 6,415,335 – Claim 29	CERN (Vega) System
A computer-implemented method comprising the steps of:	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 1.
transferring a request from an HTTP-compliant device to a dispatcher;	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
maintaining dynamic information regarding data sources a given page server may access;	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 10.
dispatching said request to an appropriate page server based on said request and based on said dynamic information,	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
said page server receiving said request and releasing said HTTP-compliant device to process other	The server receives the request. Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large

USP 6,415,335 – Claim 29	CERN (Vega) System
requests;	Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 7.
dynamically generating a page in response to said request, said page including data dynamically retrieved from one or more data sources.	Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, p. 9. Susana Fernández Vega and Jean-Yves Le Meur, "Collecting and Serving Information within a Large Organization," *CERN, European Laboratory for Particle Physics*, April 22, 1996, Fig. 6.

61256009 v2

Exhibit 4Q – Oracle's Supplemental Response to epicRealm's Interrogatory No. 3
U.S. Patent No. 6,415,335
Prior Art Reference: Harvest System (1994)

USP 6,415,335 – Claim 1	HARVEST SYSTEM
A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, March 1995 at pp. 1. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 6. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 2. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 62-63. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at Fig. 4.
routing a request from a Web server to a page server,	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10.
said page server receiving said request and releasing said Web server to process other requests	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 6. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.

USP 6,415,335 – Claim 1	HARVEST SYSTEM
wherein said routing step further includes the steps of:	
intercepting said request at said Web server and	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
routing said request to said page server;	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 13.
dynamically generating a Web page in response to said request, said Web page including data dynamically retrieved from one or more data sources.	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 2. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 3, Fig. 2. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 5.

USP 6,415,335 – Claim 2	HARVEST SYSTEM
The computer-implemented method in claim 1 wherein said step of routing said request includes the steps of:	
routing said request from said Web server to a dispatcher; and	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
dispatching said request to said page server.	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.

USP 6,415,335 – Claim 3	HARVEST SYSTEM
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	

USP 6,415,335 – Claim 4	HARVEST SYSTEM
The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	

USP 6,415,335 – Claim 5	HARVEST SYSTEM
The computer-implemented method in	

USP 6,415,335 – Claim 5	HARVEST SYSTEM
claim 1 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 6,415,335 – Claim 6	HARVEST SYSTEM
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 6,415,335 – Claim 7	HARVEST SYSTEM
The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	

USP 6,415,335 – Claim 8	HARVEST SYSTEM
The computer-implemented method in claim 1 wherein said page server includes tag-based text templates for configuring said Web page.	

USP 6,415,335 – Claim 9	HARVEST SYSTEM
The computer-implemented method in	

USP 6,415,335 – Claim 9	HARVEST SYSTEM
claim 8 wherein said step of processing said request further includes the step of inserting said-dynamically retrieved data from said one or more data sources into said tag-based text templates.	

USP 6,415,335 – Claim 10	HARVEST SYSTEM
The computer-implemented method in claim 8 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request.	

USP 6,415,335 – Claim 11	HARVEST SYSTEM
The computer-implemented method in claim 8 wherein said tag-based text templates include HTML templates.	

USP 6,415,335 – Claim 12	HARVEST SYSTEM
The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources.	

USP 6,415,335 – Claim 13	HARVEST SYSTEM

USP 6,415,335 – Claim 13	HARVEST SYSTEM
The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of processing an object handling extension.	

USP 6,415,335 – Claim 14	HARVEST SYSTEM
The computer-implemented method in claim 13 wherein said object handling extension is an OLE extension.	

USP 6,415,335 – Claim 15	HARVEST SYSTEM
A computer-implemented method comprising the steps of:	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, March 1995 at p. 1. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 6. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 2, Fig. 1. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at pp. 62-63, Fig. 4.
transferring a request from an HTTP-compliant device to a page server,	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, March 1995 at p. 1.

USP 6,415,335 – Claim 15	HARVEST SYSTEM
	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
said page server receiving said request and releasing said HTTP-compliant device to process other requests	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 6. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
wherein said transferring step further includes the steps of:	
intercepting said request at said HTTP-compliant device and	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
transferring said request to said page server;	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 13.
dynamically generating a page in	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A

USP 6,415,335 – Claim 15	HARVEST SYSTEM
response to said request, said page including data dynamically retrieved from one or more data sources.	Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 2. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 3, Fig. 2. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 5.

USP 6,415,335 – Claim 16	HARVEST SYSTEM
The computer-implemented method in claim 15 wherein said step of transferring said request includes the steps of:	
transferring said request from said HTTP-compliant device to a dispatcher; and	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
dispatching said request to said page server.	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.

USP 6,415,335 – Claim 17	HARVEST SYSTEM
The computer-implemented method in claim 15 wherein said step of processing said request includes the step of identifying said one or more	

USP 6,415,335 – Claim 17	HARVEST SYSTEM
data sources from which to retrieve said data.	

USP 6,415,335 – Claim 18	HARVEST SYSTEM
The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of dynamically retrieving said data from said one or more data sources.	

USP 6,415,335 – Claim 19	HARVEST SYSTEM
The computer-implemented method in claim 15 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 6,415,335 – Claim 20	HARVEST SYSTEM
The computer-implemented method in claim 15 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 6,415,335 – Claim 21	HARVEST SYSTEM

USP 6,415,335 – Claim 21	HARVEST SYSTEM
The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of maintaining a page cache containing said page.	

USP 6,415,335 – Claim 22	HARVEST SYSTEM
The computer-implemented method in claim 15 wherein said page server includes tag-based text templates for configuring said page.	

USP 6,415,335 – Claim 23	HARVEST SYSTEM
The computer-implemented method in claim 22 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said tag-based text templates.	

USP 6,415,335 – Claim 24	HARVEST SYSTEM
The computer-implemented method in claim 22 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request.	

USP 6,415,335 – Claim 25	HARVEST SYSTEM
The computer-implemented method in claim 22 wherein said tag-based text templates include HTML templates.	

USP 6,415,335 – Claim 26	HARVEST SYSTEM
The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources.	

USP 6,415,335 – Claim 27	HARVEST SYSTEM
The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of processing an object handling extension.	

USP 6,415,335 – Claim 28	HARVEST SYSTEM
The computer-implemented method in claim 27 wherein said object handling extension is an OLE extension.	

USP 6,415,335 – Claim 29	HARVEST SYSTEM
A computer-implemented method comprising the steps of:	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado -

USP 6,415,335 – Claim 29	HARVEST SYSTEM
	Boulder Technical report CU-CS-732-94, March 1995 at p. 1. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 6. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 2, Fig. 1. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at pp. 62-63, Fig. 4.
transferring a request from an HTTP-compliant device to a dispatcher;	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz, and Duane P. Wessels, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, March 1995 at p. 1. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
maintaining dynamic information regarding data sources a given page server may access;	The replicator can add new replicas as needed. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 62, Fig. 4. Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.

USP 6,415,335 – Claim 29	HARVEST SYSTEM
dispatching said request to an appropriate page server based on said request and based on said dynamic information,	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
said page server receiving said request and releasing said HTTP-compliant device to process other requests;	Darren R. Hardy, Michael F. Schwartz and Duane Wessels, Harvest User's Manual, Version 1.3, University of Colorado - Boulder Technical report CU-CS-743-94, September 7, 1995, at p. 64.
processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 13.
dynamically generating a page in response to said request, said page including data dynamically retrieved from one or more data sources.	C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz, Harvest: A Scalable, Customizable Discovery and Access System, University of Colorado - Boulder Technical report CU-CS-732-94, July 1994 at p. 10. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 2. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 3, Fig. 2. Darren R. Hardy and Michael F. Schwartz, Harvest User's Manual, University of Colorado - Boulder Technical report CU-CS-743-94, October 1994 at p. 5.

Exhibit 4R – Oracle's Supplemental Response to epicRealm's Interrogatory No. 3
U.S. Patent No. 6,415,335
Prior Art Reference: Web Datablade System (1996)

USP 6,415,335 – Claim 1	WEB DATABLADE SYSTEM
A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 364. Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
routing a request from a Web server to a page server,	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
said page server receiving said request and releasing said Web server to process other requests	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
wherein said routing step further includes the steps of:	
intercepting said request at said Web server and	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 367.
routing said request to said page server;	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
dynamically generating a Web page in response to said request, said Web page including data dynamically retrieved from one or more data sources.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.

USP 6,415,335 – Claim 2	WEB DATABLADE SYSTEM

USP 6,415,335 – Claim 2	WEB DATABLADE SYSTEM
The computer-implemented method in claim 1 wherein said step of routing said request includes the steps of:	
routing said request from said Web server to a dispatcher; and	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 367.
dispatching said request to said page server.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.

USP 6,415,335 – Claim 3	WEB DATABLADE SYSTEM
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	

USP 6,415,335 – Claim 4	WEB DATABLADE SYSTEM
The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	

USP 6,415,335 – Claim 5	WEB DATABLADE SYSTEM
The computer-implemented method in	

USP 6,415,335 – Claim 5	WEB DATABLADE SYSTEM
claim 1 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 6,415,335 – Claim 6	WEB DATABLADE SYSTEM
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 6,415,335 – Claim 7	WEB DATABLADE SYSTEM
The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	

USP 6,415,335 – Claim 8	WEB DATABLADE SYSTEM
The computer-implemented method in claim 1 wherein said page server includes tag-based text templates for configuring said Web page.	

USP 6,415,335 – Claim 9	WEB DATABLADE SYSTEM
The computer-implemented method in	

USP 6,415,335 – Claim 9	WEB DATABLADE SYSTEM
claim 8 wherein said step of processing said request further includes the step of inserting said-dynamically retrieved data from said one or more data sources into said tag-based text templates.	

USP 6,415,335 – Claim 10	WEB DATABLADE SYSTEM
The computer-implemented method in claim 8 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request.	

USP 6,415,335 – Claim 11	WEB DATABLADE SYSTEM
The computer-implemented method in claim 8 wherein said tag-based text templates include HTML templates.	

USP 6,415,335 – Claim 12	WEB DATABLADE SYSTEM
The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources.	

USP 6,415,335 – Claim 13	WEB DATABLADE SYSTEM

USP 6,415,335 – Claim 13	WEB DATABLADE SYSTEM
The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of processing an object handling extension.	

USP 6,415,335 – Claim 14	WEB DATABLADE SYSTEM
The computer-implemented method in claim 13 wherein said object handling extension is an OLE extension.	

USP 6,415,335 – Claim 15	WEB DATABLADE SYSTEM
A computer-implemented method comprising the steps of:	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 364. Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
transferring a request from an HTTP-compliant device to a page server,	
said page server receiving said request and releasing said HTTP-compliant device to process other requests	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
wherein said transferring step further includes the steps of:	
intercepting said request at said HTTP-compliant device and	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 367.

USP 6,415,335 – Claim 15	WEB DATABLADE SYSTEM
transferring said request to said page server;	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
dynamically generating a page in response to said request, said page including data dynamically retrieved from one or more data sources.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.

USP 6,415,335 – Claim 16	WEB DATABLADE SYSTEM
The computer-implemented method in claim 15 wherein said step of transferring said request includes the steps of:	
transferring said request from said HTTP-compliant device to a dispatcher; and	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 367.
dispatching said request to said page server.	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.

USP 6,415,335 – Claim 17	WEB DATABLADE SYSTEM
The computer-implemented method in	

USP 6,415,335 – Claim 17	WEB DATABLADE SYSTEM
claim 15 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	

USP 6,415,335 – Claim 18	WEB DATABLADE SYSTEM
The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of dynamically retrieving said data from said one or more data sources.	

USP 6,415,335 – Claim 19	WEB DATABLADE SYSTEM
The computer-implemented method in claim 15 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 6,415,335 – Claim 20	WEB DATABLADE SYSTEM
The computer-implemented method in claim 15 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 6,415,335 – Claim 21	WEB DATABLADE SYSTEM
The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of maintaining a page cache containing said page.	

USP 6,415,335 – Claim 22	WEB DATABLADE SYSTEM
The computer-implemented method in claim 15 wherein said page server includes tag-based text templates for configuring said page.	

USP 6,415,335 – Claim 23	WEB DATABLADE SYSTEM
The computer-implemented method in claim 22 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said tag-based text templates.	

USP 6,415,335 – Claim 24	WEB DATABLADE SYSTEM
The computer-implemented method in claim 22 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in	

USP 6,415,335 – Claim 24	WEB DATABLADE SYSTEM
response to said request.	

USP 6,415,335 – Claim 25	WEB DATABLADE SYSTEM
The computer-implemented method in claim 22 wherein said tag-based text templates include HTML templates.	

USP 6,415,335 – Claim 26	WEB DATABLADE SYSTEM
The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources.	

USP 6,415,335 – Claim 27	WEB DATABLADE SYSTEM
The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of processing an object handling extension.	

USP 6,415,335 – Claim 28	WEB DATABLADE SYSTEM
The computer-implemented method in claim 27 wherein said object handling extension is an OLE extension.	

USP 6,415,335 – Claim 29	WEB DATABLADE SYSTEM
A computer-implemented method comprising the steps of:	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 364. Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
transferring a request from an HTTP-compliant device to a dispatcher;	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367. Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 369.
maintaining dynamic information regarding data sources a given page server may access;	
dispatching said request to an appropriate page server based on said request and based on said dynamic information,	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
said page server receiving said request and releasing said HTTP-compliant device to process other requests;	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.
dynamically generating a page in response to said request, said page	Brian Baker, "Doing business with the Web: The Informix/Illustra Approach," 5th International conference on extending database technology, March, 1996 at 366-367.

USP 6,415,335 – Claim 29	WEB DATABLADE SYSTEM
including data dynamically retrieved from one or more data sources.	

61205804 v1

Exhibit 4S – Oracle's Supplemental Response to epicRealm's Interrogatory No. 3
U.S. PATENT NO. 6,415,335

REFERENCE: Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.

USP 6,415,335 – Claim 1	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
routing a request from a Web server to a page server,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
said page server receiving said request and releasing said Web server to process other requests	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
wherein said routing step further includes the steps of:	
intercepting said request at said Web server and	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
routing said request to said page server;	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
dynamically generating a Web page in response to said request, said Web page including data dynamically retrieved from one or more data sources.	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 100.

USP 6,415,335 – Claim 2	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.

USP 6,415,335 – Claim 2	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 1 wherein said step of routing said request includes the steps of:	
routing said request from said Web server to a dispatcher; and	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
dispatching said request to said page server.	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.

USP 6,415,335 – Claim 3	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	

USP 6,415,335 – Claim 4	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or more data sources.	

USP 6,415,335 – Claim 5	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.

USP 6,415,335 – Claim 5	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 6,415,335 – Claim 6	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 6,415,335 – Claim 7	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	

USP 6,415,335 – Claim 8	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 1 wherein said page server includes tag-based text templates for configuring said Web page.	

USP 6,415,335 – Claim 9	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 8 wherein said step of processing said request further includes the step of inserting said-dynamically retrieved data from said one or more data sources into said tag-based text templates.	

USP 6,415,335 – Claim 10	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 8 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request.	

USP 6,415,335 – Claim 11	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 8 wherein said tag-based text templates include HTML templates.	

USP 6,415,335 – Claim 12	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources.	

USP 6,415,335 – Claim 13	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of processing an object handling extension.	

USP 6,415,335 – Claim 14	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 13 wherein said object handling extension is an OLE extension.	

USP 6,415,335 – Claim 15	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
A computer-implemented method comprising the steps of:	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
transferring a request from an HTTP-compliant device to a page server,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
said page server receiving said request and releasing said HTTP-compliant device to process other requests	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
wherein said transferring step further includes the steps of:	
intercepting said request at said HTTP-compliant device and	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
transferring said request to said	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services,"

USP 6,415,335 – Claim 15	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
page server;	Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
dynamically generating a page in response to said request, said page including data dynamically retrieved from one or more data sources.	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 100.

USP 6,415,335 – Claim 16	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 15 wherein said step of transferring said request includes the steps of:	
transferring said request from said HTTP-compliant device to a dispatcher; and	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
dispatching said request to said page server.	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.

USP 6,415,335 – Claim 17	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 15 wherein said step of processing said request includes the	The DNS could identify the sources of the data "An additional experiment would be to partition other types of content (e.g., graphics) to their own

USP 6,415,335 – Claim 17	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
step of identifying said one or more data sources from which to retrieve said data.	machines in the same way. If necessary, DNS-based load-balancing could be used to spread out load across multiple machines of the same type (e.g., gif.netscape.com could be used to refer to multiple machines whose only purpose in life is to serve GIF files)." Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 100.

USP 6,415,335 – Claim 18	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of dynamically retrieving said data from said one or more data sources.	

USP 6,415,335 – Claim 19	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 15 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 6,415,335 – Claim 20	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 15 wherein said step of processing said request includes the	

USP 6,415,335 – Claim 20	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
step of logging into said one or more data sources.	

USP 6,415,335 – Claim 21	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of maintaining a page cache containing said page.	

USP 6,415,335 – Claim 22	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 15 wherein said page server includes tag-based text templates for configuring said page.	

USP 6,415,335 – Claim 23	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 22 wherein said step of processing said request further includes the step of inserting said dynamically retrieved data from said one or more data sources into said tag-based text templates.	

USP 6,415,335 – Claim 24	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.

USP 6,415,335 – Claim 24	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 22 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request.	

USP 6,415,335 – Claim 25	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 22 wherein said tag-based text templates include HTML templates.	

USP 6,415,335 – Claim 26	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources.	

USP 6,415,335 – Claim 27	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of processing an object handling extension.	

USP 6,415,335 – Claim 28	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
The computer-implemented method in claim 27 wherein said object handling extension is an OLE extension.	

USP 6,415,335 – Claim 29	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
A computer-implemented method comprising the steps of:	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
transferring a request from an HTTP-compliant device to a dispatcher;	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
maintaining dynamic information regarding data sources a given page server may access;	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 100.
dispatching said request to an appropriate page server based on said request and based on said dynamic information,	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
said page server receiving said request and releasing said HTTP-compliant device to process other requests;	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 95.
dynamically generating a page in response to said request, said page	The Netscape site could use CGI to retrieve dynamic content from data sources.

USP 6,415,335 – Claim 29	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995.
including data dynamically retrieved from one or more data sources.	Dan Mosedale, William Foss and Rob McCool, "Administering Very High Volume Internet Services," Ninth System Administration Conference (LISA '95), September 18-22, 1995, p. 100.

61229484 v1

Exhibit 4T – Oracle's Supplemental Response to epicRealm's Interrogatory No. 3
U.S. PATENT NO. 6,415,335
Prior Art Reference: SWEB System (1996)

USP 6,415,335 – Claim 1	SWEB System
A computer-implemented method for managing a dynamic Web page generation request to a Web server, said computer-implemented method comprising the steps of:	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 850.
routing a request from a Web server to a page server,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 852.
said page server receiving said request and releasing said Web server to process other requests	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 851. Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 850.
wherein said routing step further includes the steps of:	
intercepting said request at said Web server and	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
routing said request to said page server;	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853. Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
processing said request, said processing being performed by said page server while said Web server concurrently processes said other requests; and	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
dynamically generating a Web page in response to said request, said Web page	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.

USP 6,415,335 – Claim 1	SWEB System
including data dynamically retrieved from one or more data sources.	

USP 6,415,335 – Claim 2	SWEB System
The computer-implemented method in claim 1 wherein said step of routing said request includes the steps of:	
routing said request from said Web server to a dispatcher; and	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
dispatching said request to said page server.	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.

USP 6,415,335 – Claim 3	SWEB System
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	

USP 6,415,335 – Claim 4	SWEB System
The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of dynamically retrieving said data from said one or	

USP 6,415,335 – Claim 4	SWEB System
more data sources.	

USP 6,415,335 – Claim 5	SWEB System
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more data sources.	

USP 6,415,335 – Claim 6	SWEB System
The computer-implemented method in claim 1 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 6,415,335 – Claim 7	SWEB System
The computer-implemented method in claim 1 wherein said step of dynamically generating said Web page includes the step of maintaining a page cache containing said Web page.	

USP 6,415,335 – Claim 8	SWEB System
The computer-implemented method in claim 1 wherein said page server includes tag-based text templates for	

USP 6,415,335 – Claim 8	SWEB System
configuring said Web page.	

USP 6,415,335 – Claim 9	SWEB System
The computer-implemented method in claim 8 wherein said step of processing said request further includes the step of inserting said-dynamically retrieved data from said one or more data sources into said tag-based text templates.	

USP 6,415,335 – Claim 10	SWEB System
The computer-implemented method in claim 8 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request.	

USP 6,415,335 – Claim 11	SWEB System
The computer-implemented method in claim 8 wherein said tag-based text templates include HTML templates.	

USP 6,415,335 – Claim 12	SWEB System
The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of	

USP 6,415,335 – Claim 12	SWEB System
dynamically updating data at said one or more data sources.	

USP 6,415,335 – Claim 13	SWEB System
The computer-implemented method in claim 1 wherein said step of processing said request further includes the step of processing an object handling extension.	

USP 6,415,335 – Claim 14	SWEB System
The computer-implemented method in claim 13 wherein said object handling extension is an OLE extension.	

USP 6,415,335 – Claim 15	SWEB System
A computer-implemented method comprising the steps of:	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 850.
transferring a request from an HTTP-compliant device to a page server,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 852.
said page server receiving said request and releasing said HTTP-compliant device to process other requests	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 851. Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 850.

USP 6,415,335 – Claim 15	SWEB System
wherein said transferring step further includes the steps of:	
intercepting said request at said HTTP-compliant device and	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 852. Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
transferring said request to said page server;	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853. Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
dynamically generating a page in response to said request, said page including data dynamically retrieved from one or more data sources.	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.

USP 6,415,335 – Claim 16	SWEB System
The computer-implemented method in claim 15 wherein said step of transferring said request includes the steps of:	
transferring said request from said	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide

USP 6,415,335 – Claim 16	SWEB System
HTTP-compliant device to a dispatcher; and	Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
dispatching said request to said page server.	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853. -

USP 6,415,335 – Claim 17	SWEB System
The computer-implemented method in claim 15 wherein said step of processing said request includes the step of identifying said one or more data sources from which to retrieve said data.	

USP 6,415,335 – Claim 18	SWEB System
The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of dynamically retrieving said data from said one or more data sources.	

USP 6,415,335 – Claim 19	SWEB System
The computer-implemented method in claim 15 wherein said step of processing said request includes the step of said page server maintaining a connection cache to said one or more	

USP 6,415,335 – Claim 19	SWEB System
data sources.	

USP 6,415,335 – Claim 20	SWEB System
The computer-implemented method in claim 15 wherein said step of processing said request includes the step of logging into said one or more data sources.	

USP 6,415,335 – Claim 21	SWEB System
The computer-implemented method in claim 15 wherein said step of dynamically generating said page includes the step of maintaining a page cache containing said page.	

USP 6,415,335 – Claim 22	SWEB System
The computer-implemented method in claim 15 wherein said page server includes tag-based text templates for configuring said page.	

USP 6,415,335 – Claim 23	SWEB System
The computer-implemented method in claim 22 wherein said step of processing said request further includes the step of inserting said dynamically	

USP 6,415,335 – Claim 23	SWEB System
retrieved data from said one or more data sources into said tag-based text templates.	

USP 6,415,335 – Claim 24	SWEB System
The computer-implemented method in claim 22 wherein at least one of said tag-based text templates drives a format of the data dynamically retrieved from said one or more data sources in response to said request.	

USP 6,415,335 – Claim 25	SWEB System
The computer-implemented method in claim 22 wherein said tag-based text templates include HTML templates.	

USP 6,415,335 – Claim 26	SWEB System
The computer-implemented method in claim 15 wherein said step of processing said request further includes the step of dynamically updating data at said one or more data sources.	

USP 6,415,335 – Claim 27	SWEB System
The computer-implemented method in claim 15 wherein said step of	

USP 6,415,335 – Claim 27	SWEB System
processing said request further includes the step of processing an object handling extension.	

USP 6,415,335 – Claim 28	SWEB System
The computer-implemented method in claim 27 wherein said object handling extension is an OLE extension.	

USP 6,415,335 – Claim 29	SWEB System
A computer-implemented method comprising the steps of:	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 850.
transferring a request from an HTTP-compliant device to a dispatcher;	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 852.
maintaining dynamic information regarding data sources a given page server may access;	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
dispatching said request to an appropriate page server based on said request and based on said dynamic information,	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
said page server receiving said request and releasing said HTTP-compliant device to process other requests;	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 851. Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 850.

USP 6,415,335 – Claim 29	SWEB System
processing said request, said processing being performed by said page server while said HTTP-compliant device concurrently processes said other requests; and	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.
dynamically generating a page in response to said request, said page including data dynamically retrieved from one or more data sources.	Daniel Andresen, Tao Yang, Vegard Homedahl, Oscar H. Ibarra, "SWEB: Towards a Scalable World Wide Web Server on Multicomputers," Department of Computer Science, University of California, 1996, p. 853.

61227132 v1

Exhibit 24

JENNER&BLOCK

January 14, 2008

Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611
Tel 312-222-9350
www.jenner.com

Chicago
Dallas
New York
Washington, DC

VIA E-MAIL

Theodore T. Herhold, Esq.
Townsend and Townsend and Crew
379 Lytton Avenue
Palo Alto, CA 94301-1431

David R. Bennett
Tel 312 923-8314
Fax 312 923-8414
dbennett@jenner.com

Re: Oracle Corp. et al. v. epicRealm Licensing, LP
CA No. 06-CV-414 (SLR)

Dear Ted:

The letter is in regard to Oracle's continued, and extremely delinquent, production of documents.

In a letter accompanying Oracle's September 14, 2007 production, Oracle stated that "due to the large volume of documents Oracle has been processing for production this past week, we are still in the midst of processing ***a small percentage of documents*** that we intend to produce by FedEx within the next few days." (9/14/07 letter R. Artuz to B. Bradford (emphasis added)). On September 19, 2007, during a meet and confer, Oracle represented to epicRealm that it only had a ***few hundred documents*** remaining to produce. During the October 29, 2007 status hearing, Oracle represented to the Court that the parties "reached a tentative agreement that the document production ***will be done by the end of November***." (Transcript at 5:11-12 (emphasis added)). On November 30, 2007, Oracle produced only 26,000 additional pages and stated that "we intend to produce ***some additional documents*** that are currently being reviewed and processed by our vendor which we anticipate producing shortly." (Trevino Letter of November 30, 2007 (emphasis added)).

Despite all these representations, Oracle has produced over 550,000 pages since December 19, 2007. Most recently on January 10, 2008, six weeks after the date that Oracle told the Court that it's document production "will be done," Oracle produced over 170,000 pages of documents. Oracle's production continues without any apparent end in sight and indisputably contradicts its representations to the Court and callously disregards the Court's scheduling order.

On what day will Oracle complete its document production and exactly how many additional documents will Oracle produce before completing its production?

Very truly yours,

David R. Bennett

DRB:tam

cc: Eric M. Hutchins, Esq.
Robert J. Artuz, Esq.
Mary B. Graham, Esq.
George S. Bosy, Esq.
Patrick L. Patras, Esq.
Benjamin J. Bradford, Esq.

Exhibit 25

JENNER&BLOCK

January 14, 2008

Jenner & Block LLP
330 N. Wabash Avenue
Chicago, IL 60611
Tel 312-222-9350
www.jenner.com

Chicago
Dallas
New York
Washington, DC

VIA EMAIL

Theodore T. Herhold, Esq.
Townsend and Townsend and Crew
379 Lytton Avenue
Palo Alto, CA 94301-1431

Patrick L. Patras
Tel 312 923-2945
Fax 312 840-7345
ppatras@jenner.com

Re: Oracle Corp. et al. v. epicRealm Licensing, LP
CA No. 06-CV-414 (SLR)

Dear Ted:

Inequitable conduct allegations must be pled with particularity under the strictures of Federal Rule of Civil Procedure 9(b). In particular, inequitable conduct pleadings must specifically name any allegedly withheld information. *See, e.g., EMC Corp. v. Storage Tech. Corp.*, 921 F. Supp. 1261, 1263 (D. Del. 1996). Oracle's draft Amended Complaint does not comply with the pleading requirements of Rule 9(b) because it fails to specifically name the allegedly withheld information.

Please specifically name the allegedly withheld information within Oracle's categories of "Oracle Prior Art" and the "'673 Prior Art" in compliance with Rule 9(b) by 5:00 p.m. C.S.T. on Tuesday, January 15, 2008. If we do not receive such specific information by then, epicRealm intends to rely on Oracle's failure to respond in its opposition to Oracle's motion for leave to amend.

Very truly yours,

Patrick L. Patras

Exhibit 26

From: Herhold, Ted T. [mailto:ttherhold@townsend.com]
Sent: Monday, January 14, 2008 5:15 PM
To: Patras, Patrick L
Cc: Artuz, Robert J.; Hutchins, Eric M.; mgraham@MNAT.com
Subject: RE: Oracle v. epicRealm

Pat, your letter/demand comes a little late in the day. You have known of our intent to add the inequitable conduct claim for months. We provided you with a list of the evidence we intended to rely on in early December. While we believe we have complied with Rule 9 requirements, we will consider your request under the time constraints you created.

Regards,

Ted

Theodore T. Herhold
Partner
Townsend and Townsend and Crew LLP
379 Lytton Ave.
Palo Alto, CA 94301
(650) 326-2400 (main)
(650) 326-2422 (fax)
(650) 752-2433 (direct)
tth@townsend.com

This message is from a lawyer and may contain confidential and/or privileged information. If you are not the intended recipient, please notify me immediately by return email and delete this message from your computer system.

Exhibit 27

From: Artuz, Robert J. [mailto:rjartuz@townsend.com]
Sent: Thursday, January 17, 2008 7:52 PM
To: Patras, Patrick L
Cc: Hutchins, Eric M.; mgraham@MNAT.com; Herhold, Ted T.
Subject: RE: Oracle v. epicRealm

Pat,

In response to your January 14, 2008 letter, Oracle alleges that "inventor Keith Lowery ("Lowery") researched and had knowledge of highly material prior art including but not limited to publications describing -- and the prior sale, offer for sale or public use of -- Oracle web server products (the "Oracle Prior Art") that were known and available to the public prior to the alleged conception and reduction to practice of the claimed inventions of the Parallel Networks patents." As discussed and referenced in our briefing, these Oracle web products include at least Oracle WebServer and Oracle's Web-Oracle-Web interface. Keith Lowery was also aware of publications describing these web server products including compuserve pages as referenced in Lowery's September 1, 1995 (Bates No. EPIC000149-150).

Oracle also alleges that "the attorneys responsible for prosecuting the Parallel Networks patents, including Jim Salter Esq. and Sharmini Green Esq. of the law firm of Blakely Sokoloff Taylor and Zafman LLP, had knowledge of additional highly material prior art to the Parallel Networks patents, including but not limited to the prior art methods and apparatus described in U.S. Patent No. 5,761,673 issued to Oracle and the prior art references cited therein (the "'673 Prior Art")." This art includes at least the '673 patent's disclosure and the references cited in the patent, which are plainly apparent on the patent's face and in the patent's prosecution history:

U.S. Patent Documents

5530852	June 1996	Meske, Jr. et al.
5537586	July 1996	Amram et al.
5572643	November 1996	Judson
5701451	December 1997	Rogers et al.
5710918	January 1998	Lagarde et al.

Other References

"Oracle Version 7", Computergram International, Jun. 17, 1992. .
"Oracle Rides the Internet with Web", Computergram International, Feb. 10, 1995. .
"Sun Java, Netscape Livescript Married as Javascript", Computergram International, Dec. 1995. .
Marshall et al. "Industry Lines Up for Java--Vendors Set Stage for Dynamic", Commications Week, Dec. 1995. .

"Oracle Unveils Oracle Websystem Internet Line", Computergram International, Nov. 1995. .
Marshall, Martin "Sybase Sites to Get Web Links", Communciations Week, Nov. 1995. .
Karpinski, Richard "Oracle Readies Web Initiative-Businesses Can Tie Their Databases Directly to the Web", Interactive Age, n207, p. 1, Jan. 30, 1995. .
"Hypercard-Based Oracle Card for Macintosh, Windows 3.0", Computergram International, Jan. 1991. .
Karpinski, Richard "Web-Based Help Desks Debut", Communications Week, Oct. 1995. .
"Next Leaps Aboard the Internet with Webobjects", Computergram International, Aug. 15, 1905. .
Clinton et al. "Intranet Tools: Coprorations Seek Internal Web Applications--and Major Vendors are Happy to Help", Informationweek, n552, p14(2), Nov. 6, 1992. .
Nash, Kim S. "Web Browser Makers Follw Modular Trend", News, Apr. 29, 1996. .
"Netscape & Sun Announce Javascript", Newsbytes, Dec. 4, 1995. .
"Oracle Challlenges Netscape with Free Powerbrowser", Computergram International, Dec. 1995. .
"Oracle Set Powerbrowser", Computergram International, Dec. 1995. .
"Oracle Begins Internet Initiative:Multimedia, Hypertext Development Tools", Seybold Report on Desktop Publishing, v9, n7, p27(1), Mar. 6, 1995. .
Menefee, Craig "Oracle Intros Database Products", Newsbytes, Jan. 23, 1995. .
"Sybase: Sybase Introduces Powersite Enterprise Web Development Environment", M2 Communications, Dec. 1997. .
"ObjectShare IntroducesjKit/Grid; First Offering in a New Line of Advanced Java Components; Extensible New Software Provides Powerful Java Classes for Grids, Tables and More; Advanced Component Automatically Reacts to Data Changes", Buisness Wire, pp. 101, Oct. 1996..

We assume this answers your question.

Regards,

Rob Artuz