Related U.S. Patent Documents

Reissue of:
(64) Patent No.:  7,010,568
Issued:    Mar. 7, 2006
Appl. No.: 09/643,584
Filed:  Aug. 22, 2000

U.S. Applications:
(63) Continuation-in-part of application No. 09/532,500, filed on Mar. 21, 2000, now Pat. No. 7,136,932, and a continuation-in-part of application No. 09/525,350, filed on Mar. 15, 2000, now Pat. No. 6,338,082.
(60) Provisional application No. 60/152,015, filed on Sep. 1, 1999, provisional application No. 60/153,594, filed on Sep. 13, 1999, provisional application No. 60/143,859, filed on Jul. 15, 1999, provisional application No. 60/135,751, filed on May 25, 1999, provisional application No. 60/125,531, filed on Mar. 22, 1999.

Int. Cl. G06F 15/16 (2006.01)

U.S. Cl. 709/203; 709/217; 709/219; 707/999.003; 707/999.001
FOREIGN PATENT DOCUMENTS

JP 11085492 3/1999
JP 11184667 7/1999
JP 11242682 9/1999
JP 11296428 10/1999
WO 99979726 2/1999
WO 999723188 5/1999
WO 99972979 8/1999
WO 0007133 2/2000

OTHER PUBLICATIONS


Norman, J.; Southern Telecom and Main.net Announce Successful Demonstration of Broadband over Power Lines; Southern Telecom; Dec. 2003, p. 1.

Motegi, N.; Pickett, M.; Web-based Energy Information Systems for Large Commercial Buildings; Ernest Orlando Lawrence Berkeley National Laboratory; May 2002; USA; p. 3, paragraphs 1, 2, 4, Figure 1; p. 4, paragraphs 1, 2, 3, 4, p. 7, paragraphs 2, 3, 4, Table 1; 10, paragraph 2; p. 11, paragraph 1, Table 5.


Perez, Juan Carlos, Mozilla Launches Firefox 1.0, PCWORLD.com, Nov. 9, 2004 <http://www.pcworld.com-news-article-0aid,118537,00.asp>.

Advanced Control Systems, Inc.; Load Management; Advanced Control Systems, Inc.; Jun. 20, 2001 (Aug. 1999); USA; p. 1, paragraph 1; p. 2, paragraphs 1, 2, 3, paragraphs 1 & 2; p. 4, paragraphs 1 & 2.

American City Business Journal, Inc.; Blackout early warnings mandated; Apr. 3, 2001; USA; paragraphs 1, 2, 5, 6.

Bradley, J.; ISO Blackout Notice Plan Follow-Up; Silicon Valley Manufacturers Group & California Independent System Operator (CAISO); May 8, 2001; p. 1, paragraphs 1, 2, 3, 4, p. 2, paragraphs 1, 3, 4.

California, State of; Energy Action Plan; May 20, 2003; USA; p. 5; Section 1; pp. 7 & 8 Section V.

California Independent System Operator; California Independent System Operator Participant’s whom have entered into an Interruptible Service Contract or similar agreement; California Independent System Operator (CAISO); Dec. 10, 2002; p. 1.

Desmond, J.; Customer Centric Alert and Response Program Overview; California Independent System Operator (CAISO); May 24, 2001; USA; pp. 2, 3, 4, 5, 6, 12, 13, 17, 18.

Goldman, C.A.; Kintner-Meyer, M.; Heffler, G.; Do “Enabling Technologies” Affect Customer Performance in Price-Responsive Load Programs?; LBNL-50328; Ernest Orlando Lawrence Berkeley National Laboratory; http://eetd.lbl.gov/E-E-MP/; Aug. 2002; p. 3, paragraph 2; p. 4, paragraphs 2, p. 6, paragraph 3; p. 7, paragraphs 1 & 2, Table 2; p. 8, paragraphs 1, 2, 3, Table 3; p. 9, paragraphs 1, 2, 3, Table 4; p. 10, paragraphs 1, 2, 3, 5, 11, paragraph 1, Table 5; p. 12, paragraphs 1 & 2, Figure 2; p. 15, paragraphs 2, 3, 4, 6; 16, paragraph 2.

Graves, K.; CAISO Summer 2001 Assessment; California Independent System Operator (CAISO); Mar. 22, 2001; USA; p. 29, paragraphs 1 & 2, p. 30, paragraphs 1, 2, Figure III-E.

Graves, K.; CAISO Feb. 2001 Winter Assessment and Summer 2001 Post-season Summary; California Independent System Operator (CAISO); Oct. 8, 2001; USA; pp. 17 & 18, Section IV.

Gross, G.; FCC moves ahead with powerline broadband rules; IDG News Service; Feb. 13, 2004; USA; paragraphs 1, 2, 6, 9.

Guardino; Board of Governors; Notification Plan 2; California Independent System Operator (CAISO); May 21, 2001; USA.

Helman, C.; For years electric companies have dreamed of making their wires the high-speed data pipe to your PC. One tech company may yet make the dream possible; Forbes.com; Jan. 20, 2003; USA; paragraphs 1, 2, 3.

Jonker, R.; Dijk, P.; Enabling Distributed Generation and Demand Response with Enterprise Energy-Management Systems; Darnell. com Inc; May 17, 2001; USA; p. 1, paragraphs 1 & 2; p. 2, paragraphs 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, paragraphs 2, 3, 4, 6, 7, 8, 9, 10, 9 paragraphs 1, 2, 3.

Keoni, A.; California Independent System Operator Load Program Participants; California Independent System Operator (CAISO); Apr. 8, 2002; p. 1, 2, 3.

Keoni, A.; Implementation Plan and Required Information for the Participating Load Program (PLP); California Independent System Operator (CAISO); Aug. 8, 2002; USA; paragraphs 1, 2, paragraphs 1, 2, Table 1.

Labaton, S., F.C.C. Begins Rewriting Rules on Delivery of the Internet; Associated Press; Feb. 13, 2004; USA; paragraphs 1, 2, 6, 7, 8, 9, 10, 11, paragraphs 1, 2, 3, 4, 6, 7.


Oesterreichische Gesellschaft Fur Unwelt Und Technik; teak-experts group on r6d priority setting and evaluation liberalisation of the electricity market; http://www.oeug.at, Mar. 18, 2002; Austria (in English), pp. 4, (Sections 3 & 4), 5, 7 Section 3, 8 (Controllable “Dispatchable” Loads, Section 2, 9) (Distributed Energy Resources).
O’Neil, E.: Cumulative Totals of No-Touch, Restricted Maintenance Operations, Alert, Warning, Emergency and Power Watch Notices Issued from 1998 to Present; California Independent System Operator (CAISO); May 29, 2003; USA; p. 4, Table; p. 5; Table; p. 6; Table; p. 7; Table; p. 8; Table; p. 9; Table; p. 10; Table; p. 11; Table; p. 12; Table; p. 14; Table.


Purdum, T.; Statewide Blackouts Ordered as Heat Strains California Grid; New York Times; May 8, 2001; USA; paragraphs 3, 4, 5, 8, 9. United States of America, Federal Energy Regulation Commission; National Transmission Grid Study, Consolidated List of Recommendations; FERC; May 6, 2002; USA; pp. 2-76 & 3-77 Section 4, 5-79 Section 5.


Weller, G.H.; A Case Study Review of Technical and Technology Issues for Transition of a Utility Load Management Program to Provide System Reliability Resources in Restructured Electricity Markets, LBNL-52408; Ernest Orlando Lawrence Berkeley National Laboratory; http://www.lbl.gov; Jul. 1, 2001; USA; pp. 1 & 2, Section 1; pp. 3 & 4, Section 2; pp. 6-13, Sections 3.1-3.1.8; pp. 16-17, Section 3.3, Table 3.1; pp. 18-40, Section 4; pp. 41-42, Section 5; Appendix A.

Wesolowski, L., Why Rotating Outages? California Independent System Operator (CAISO); Aug. 17, 2001; USA; paragraphs 1, 2, 3, 5, 6, 7.


Fax-Phone Switch for Multi-Ring Telephone Lines; Derwent Abstract; Jul. 1992.

“What does your phone number spell?” Internet print-out of www.phonerecord.com; Wayback Machine.

“What Words are Hiding in Your Phone Number?” Internet print-out of www.dialabe.com; Wayback Machine.


www.phonetic.com; Internet print-out; Wayback Machine.


Network Solutions and VeriSign Launch Combined Internet Name and Certification Registration, Internet World, Mar. 10, 1997.


Schmid, E., McCorkle, S., O’Neil, E.; ISO Electric Emergency Notification Memorandum; California Independent System Operator (CAISO); May 17, 2001; USA; p. 1, paragraphs 1 & 4; p. 2, paragraphs 1, 2, 3; p. 3; p. 4, paragraph 4 & 5; p. 5; p. 6, paragraph 1, p. 7; p. 8.

* cited by examiner
Fig. 1a
Fig. 1b
Start

Receive input

Input a valid URI?

Yes

Can a URI be generated from input?

Yes

Generate valid URI

Attempt to access URI

No

Display message indicating that URI can not be generated or accessed

No

URI accessible?

Yes

End

No

Prior Art

Fig. 2
From 225
Can URI be generated from input? - No

310
Configured to determine if at least one other method may be applied to access a valid URI corresponding to input?

No → To 235 Display Message

Yes →

315
Apply at least one other method in sequence or parallel such as retrieving information from user modifiable configuration settings, template, GO LIST, name translation table, and registry (e.g., processing a FDN, processing a TLDA as a search request, multiple accessibility of a plurality of generated URIs)

End

Fig. 3
From 222

URI accessible? - No

Generate a valid URI to access content by extracting keywords from the non-query component of the inaccessible URI and using such keywords as a search request or as the query portion of the generated valid URI

To 220

Attempt to access URI

Fig. 4
From 410
Generate a valid URI to access content by extracting keywords from the non-query component of the inaccessible URI and using such keywords as a search request or as the query portion of the generated valid URI

510
Create a second URI with at least one frame that accesses the content of the generated valid URI

To 220
Attempt to access URI

Fig. 5
To 410
Generate a valid URI to access content by extracting keywords from the non-query component of the inaccessible URI and using such keywords as a search request or as the query portion of the generated valid URI

From 222
URI accessible?- No

URI in the form of SLD TLD/FLD?

Yes

No

FLD exist?

Yes

Modify file to redirect to second URI

Modify file?

Yes

File exist?

Yes

Create file that redirects to second URI

Create FLD

Generate second URI with FLD as part of search request

To 220
Attempt to access URI

Fig. 6
From 215/230
Valid URI? -
Yes

710

Execute script
to extract SLD
from first URI

715

Generate second URI
with SLD as part of
search request

720

Generate and display web page with two frames
displaying content of first and second URI when accessed

End

Fig. 7
SEARCH ENGINE REQUEST METHOD, PRODUCT, AND APPARATUS

matter enclosed in heavy brackets [ ] appears in the original patent but forms no part of this reissue specification; matter printed in italics indicates the additions made by reissue.

OTHER APPLICATIONS

This application [claims the benefit of the following patent applications, which are hereby incorporated] incorporates the following applications by reference:

[1] this application is a continuation-in-part of U.S. patent application Ser. No. 09/532,500 filed Mar. 21, 2000, by Schneider, entitled "Fictitious domain name method, product, and apparatus", [which claims the benefit of now U.S. Pat. No. 7,136,932; this application is also a continuation-in-part of U.S. patent application Ser. No. 09/523,530 filed Mar. 15, 2000, by Schneider, entitled "Method for integrating domain name registration with domain name resolution" and now U.S. Pat. No. 6,358,082; this application also claims the benefit of U.S. Provisional Application Ser. No. 60/143,859 filed Jul. 15, 1999, by Schneider entitled "Method and apparatus for generation, registration, resolution, and emulation of name space", now abandoned]; this application also claims the benefit of U.S. Provisional Application Ser. No. 60/135,751 filed May 25, 1999, by Schneider entitled "Method and system for name space resolution", now abandoned and; this application also claims the benefit of U.S. Provisional Application Ser. No. 60/125,531 filed Mar. 22, 1999, by Schneider entitled "Method and system for the emulation of name space", now abandoned.

[2] this application also claims the benefit of U.S. Provisional Application Ser. No. 60/153,594 filed Sep. 13, 1999, by Schneider entitled "Method and apparatus for using a portion of a URI to select and display advertising"; this application also claims the benefit of U.S. Provisional Application Ser. No. 60/152,015 filed Sep. 1, 1999, by Schneider et al., entitled "Method and apparatus for using a portion of a URI as a search request."

FIELD OF THE INVENTION

This invention generally relates to search requests, and more specifically relates to a method and apparatus for using a non-query URI component as a search request.

BACKGROUND OF THE INVENTION

The Internet is a vast computer network consisting of many smaller networks that span the world. A network provides a distributed communicating system of computers that are interconnected by various electronic communication links and computer software protocols. Because of the Internet's distributed and open network architecture, it is possible to transfer data from one computer to any other computer worldwide. In 1991, the World-Wide-Web (WWW or Web) revolutionized the way information is managed and distributed.

The Web is based on the concept of hypertext and a transfer method known as Hypertext Transfer Protocol (HTTP) which is designed to run primarily over a Transmission Control Protocol/Internet Protocol (TCP/IP) connection that employs a standard Internet setup. A server-computer may issue the data and a client computer displays or processes it. TCP may then convert messages into streams of packets at the source, then reassemble them back into messages at the destination. Internet Protocol (IP) handles addressing, seeing to it that packets are routed across multiple nodes and even across multiple networks with multiple standards. HTTP protocol permits client systems connected to the Internet to access independent and geographically scattered server systems also connected to the Internet.

Client side browsers, such as Netscape Navigator and/or Microsoft Internet Explorer (MIE) provide graphical user interface (GUI) based client applications that implement the client side portion of the HTTP protocol. One format for information transfer is to create documents using Hypertext Markup Language (HTML). HTML pages are made up of standard text as well as formatting codes that indicate how the page should be displayed. The client side browser reads these codes in order to display the page. A web page may be static and requires no variables to display information or link to other predetermined web pages. A web page is dynamic when arguments are passed which are either hidden in the web page or entered from a client browser to supply the necessary inputs displayed on the web page. Common Gateway Interface (CGI) is a standard for running external programs from a web server. CGI specifies how to pass arguments to the executing program as part of the HTTP server request. Commonly, a CGI script may take the name and value arguments from an input form of a first web page which may be used as a query to access a database server and generate an HTML web page with customized data results as output that is passed back to the client browser for display.

The Web is a means of accessing information on the Internet that allows a user to "surf the web" and navigate the Internet resources intuitively, without technical knowledge. The Web dispenses with command-line utilities, which typically require a user to transmit sets of commands to communicate with an Internet server. Instead, the Web is made up of millions of interconnected web pages, or documents, which may be displayed on a computer monitor. Hosts running special servers provide the Web pages. Software that runs these Web servers is relatively simple and is available on a wide range of computer platforms including PC's.

A Uniform Resource Identifier (URI) is a compact string of characters for identifying an abstract or physical resource. URIs, is the generic set of all names and addresses that refer to objects on the Internet. URIs that refer to objects accessed with existing protocols are known as URIs. A URL is the address of a file accessible on the Internet. The URI contains the name of the protocol required to access the resource, a domain name, Fully Qualified Domain Name (FQDN), or IP address that identifies a specific computer on the Internet, and a hierarchical description of a file location on the computer. In addition, the last (optional) part of the URI may be a "query" followed by a query string having name/value pairs for parameters (e.g. "?size=m&quantity=3") or a "fragment" followed by a fragment identifier indicating a particular position within the specified document.

The URI "http://www.example.com:80/index.html# appendix" is the concatenation of several components where "http:" is the scheme or protocol, "//www.example.com" is the FQDN having "www" as the host of the domain name "example.com", ":80" is the port connection for the HTTP server request, "index.html" is the filename located on the server, "/#appendix" is the identifier to display a fragment of the HTML file called "index.html". The URL "http://www.example.com" also retrieves an HTML file called "index" on
a HTTP server called "example.com". By default, when either a port or filename is omitted upon accessing a HTTP
server via a URL, the client browser interprets the request by connecting via port 80, and retrieving the HTML file
called "index".

A domain name consists of two parts: a host and a domain. Technically, the letters to the right of the "dot" (e.g., tut.net)
are referred to as Top Level Domains (TLDs), while hosts, computers with assigned IP addresses that are listed in
specific TLD registries are known as second-level domains (SLDs). For the domain name "tut.net", "net" is the TLD, and
"tut" is the SLD. Domain name space is the ordered hierarchical set of all possible domain names either in use or to be
used for locating an IP address on the Internet. TLDs are known as top-level domains because they comprise the high-
est-order name space available on the Internet. Second-level domains, as well as third-level domains (3LDs) such as "king,
tut.net", are subsidiary to TLDs in the hierarchy of the Internet's DNS.

The main use of a web browser location field is for locating URLs to access resources. Entering a URL in the location
field of a web browser serves as a means to access that URL. Because the function of the location field is so critical for
accessing resources, the design of such location fields has rivaled much competition and innovation between existing
web browser products. Improvements to better track and organize sites of URLs that users have visited such as Book-
mark folders, URL history, and the personal toolbar are all examples of functionality designed to help users navigate.

A more recent feature called Smart Browsing is integrated into Netscape Navigator that uses Internet Keywords so users
can streamline the use of URLs and get fast sites using the browser's location field. Any single or multiword
strings typed into the browser's location field that does not include a "", are sent via HTTP to a server at "netscape.com".
The keyword server pulls the string and compares it to several separate lists of keyword-URL pairs. If the keyword system
finds a match, it redirects the user's browser to the URL of the keyword-URL pair. Failing a match against the lists, the
user's browser is redirected to a Netscape Search page with the typed string as the search query.

U.S. Provisional Application Ser. No. 60/143,859 filed Jul.
15, 1999, by Schneider entitled "Method and apparatus for
generation, registration, resolution, and emulation of name
space", now abandoned, uses a domain name having a top
level domain alias (TLD) to simultaneously access and search a given resource demonstrating the combination of
both resolution and search services. For instance, when input
is received and processed such as "http://example.44106",
steps are performed to determine that "0.44106" is not a
resolvable TLD and may be processed instead as a search
term. Steps may then be performed to translate input into a
valid URI such as "http://example.com/weather/
gi?zip=44106". When the URI is accessed, a CGI script
called "weather" is executed passing the value "44106" for
the name "zip", which in this case represents a zip code. By so
doing a fictitious name is used to simultaneously access both
a resource and search request.

This clearly demonstrates the need for methods of encour-
gaging the creative use of URIs to access resources. Accord-
ingly, in light of the above, there is a strong need in the art
for a system and method to improve how resources and content
may be accessed.

SUMMARY OF THE INVENTION

Briefly, the present invention allows URI components to be
used as a search term in a query or search request. The inven-
tion allows for the non-query portion of a first URI to be used
in the query portion of a second URI to generate dynamic
content. The present invention allows for a shorter URI to be
created on the fly in real-time that accesses content of a longer
URI. The invention allows the user to combine Boolean logic
with valid URI notation to generate a search request from the
URI or vice-versa.

In general, in accordance with the present invention method
for locating a network resource from a first identifier
includes the steps of determining whether the first identifier is
accessible, accessing the first identifier in response to deter-
mining that the first identifier is accessible, generating a sec-
ond identifier in response to determining that the first identi-
fer is not accessible, wherein said second identifier is
generated by retrieving information from one or a user modifi-
fiable configuration settings, template, GO LIST, name trans-
lation table, and registry, and accessing said second identifier
in response to generating said second identifier.

In accordance with another aspect of the present invention
method for locating a network resource from a first identifier
having a valid accessible first URI includes the steps of pars-
ing at least one non-query URI component from the first URI,
generating a valid accessible second URI having a query
component that corresponds to the first URI, and simultane-
ously accessing both the first URI and said second URI.

In accordance with yet another aspect of the present invention
method for locating a network resource from a first identifier
having a valid first URI includes the steps of determin-
ing whether the first URI is accessible, accessing the first
URI in response to determining that the first URI is access-
sible, parsing at least one non-query URI component from the
first URI in response to determining that the first URI is not
accessible, generating a valid second URI having a query
component that corresponds to the first URI, and accessing
said second URI.

In accordance with yet additional aspects of the present
invention, an apparatus which implements substantially the
same functionality in substantially the same manner as the
methods described above is provided.

In accordance with other additional aspects of the present
invention, a computer-readable medium that includes com-
puter-executable instructions may be used to perform sub-
stantially the same methods as those described above is pro-
vided.

The foregoing and other features of the invention are here-
after fully described and particularly pointed out in the
claims. The following description and the annexed drawings
set forth in detail one or more illustrative aspects of the
invention, such being indicative, however, of but one or a few
of the various ways in which the principles of the invention
may be employed.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1a is a block diagram of an exemplary distributed
computer system in accordance with the present invention.

FIG. 1b is a block diagram illustrating exemplary informa-
tion records stored in memory in accordance with the present
invention.

FIG. 2 is a flowchart illustrating the steps performed by a
prior art system for accessing a URI.

FIG. 3 is a flowchart illustrating the steps performed for
generating a valid URI in accordance with the present inven-
tion.
FIG. 4 is a flowchart illustrating the steps performed for generating a valid URI in response to determining that an existing valid URI is inaccessible in accordance with the present invention.

FIG. 5 is a flowchart illustrating the steps performed for generating a frame having an accessible URI in accordance with the present invention.

FIG. 6 is a flowchart illustrating the steps performed for creating files or directories as needed to form an accessible URI in accordance with the present invention.

FIG. 7 is a flowchart illustrating the steps performed for extracting a domain to be used as a search request in accordance with the present invention.

FIG. 8 is a diagram depicting how results may be displayed in a web browser in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention will now be described with reference to the drawings, wherein like reference numerals are used to refer to like elements throughout.

Turning first to the nomenclature of the specification, the detailed description that follows represents processes and symbolic representations of operations by conventional computer components, including a local processing unit, memory storage devices for the local processing unit, display devices, and input devices. Furthermore, these processes and operations may utilize conventional computer components in a heterogeneous distributed computing environment, including remote file servers, computer servers, and memory storage devices. These distributed computing components may be accessible to the local processing unit by a communication network.

The processes and operations performed by the computer include the manipulation of data bits by a local processing unit and/or remote server and the maintenance of these bits within data structures resident in one or more of the local or remote memory storage devices. These data structures impose a physical organization upon the collection of data bits stored within a memory storage device and represent electromagnetic spectrum elements.

A process may generally be defined as being a sequence of computer-executed steps leading to a desired result. These steps generally require physical manipulations of physical quantities. Usually, though not necessarily, these quantities may take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, or otherwise manipulated. It is conventional for those skilled in the art to refer to these signals as bits or bytes (when they have binary logic levels), pixel values, works, values, elements, symbols, characters, terms, numbers, points, records, objects, images, files, directories, subdirectories, or the like. It should be kept in mind, however, that these and similar terms should be associated with appropriate physical quantities for computer operations, and that these terms are merely conventional labels applied to physical quantities that exist within and during operation of the computer.

It should also be understood that manipulations within the computer are often referred to in terms such as adding, comparing, moving, positioning, placing, illuminating, removing, altering, etc., which are often associated with manual operations performed by a human operator. The operations described herein are machine operations performed in conjunction with various input provided by a human operator or user that interacts with the computer. The machines used for performing the operation of the present invention include local or remote general-purpose digital computers or other similar computing devices.

In addition, it should be understood that the programs, processes, methods, etc. described herein are not related or limited to any particular computer or apparatus nor are they related or limited to any particular communication network architecture. Rather, various types of general-purpose machines may be used with program modules constructed in accordance with the teachings described herein. Similarly, it may prove advantageous to construct a specialized apparatus to perform the method steps described herein by way of dedicated computer systems in a specific network architecture with hard-wired logic or programs stored in nonvolatile memory, such as read only memory.

FIG. 1a illustrates an exemplary system for providing a distributed computer system 100 in accordance with one aspect of the present invention and may include client computers or any network access apparatus 110 connected to server computers 120 via a network 130. The network 130 may use Internet communication protocols (IP) to allow clients 110 to communicate with servers 120. The network access apparatus 110 may include a modem or like transceiver to communicate with the electronic network 130. The modem may communicate with the electronic network 130 via a line 116 such as a telephone line, an ISDN line, a coaxial line, a cable television line, a fiber optic line, or a computer network line. Alternatively, the modem may wirelessly communicate with the electronic network 130. The electronic network 130 may provide an on-line service, an Internet service provider, a local area network service, a wide area network service, a cable television service, a wireless data service, an intranet, a satellite service, or the like.

The client computers 110 may be any network access apparatus including hand held devices, palmtop computers, personal digital assistants (PDAs), notebook, laptop, portable computers, desktop PCs, workstations, and/or larger/smaller computer systems. It is noted that the network access apparatus 110 may have a variety of forms, including but not limited to, a general purpose computer, a network computer, an internet television, a set top box, a web-enabled telephone, an internet appliance, a portable wireless device, a game player, a video recorder, and/or an audio component, for example.

Each client 110 typically includes one or more processors 166, memories 168, and input/output devices 170. An input device may be any suitable device for the user to give input to client computer system 110, for example: a keyboard, a 10-key pad, a telephone key pad, a light pen or any pen pointing device, a touchscreen, a button, a dial, a joystick, a steering wheel, a foot pedal, a mouse, a trackball, an optical or magnetic recognition unit such as a bar code or magnetic swipe reader, a voice or speech recognition unit, a remote control attached via cable or wireless link to a game set, television, and/or cable box. A data glove, an eye-tracking device, or any MIDI device may also be used. A display device may be any suitable output device, such as a display screen, text-to-speech converter, printer, plotter, fax, television set, or audio player. Although the input device is typically separate from the display device, they may be combined; for example: a display with an integrated touchscreen, a display with an integrated keyboard, or a speech-recognition unit combined with a text-to-speech converter.

The servers 120 may be similarly configured. However, in many instances server sites 120 include many computers, perhaps connected by a separate private network. In fact, the network 130 may include hundreds of thousands of individual
networks of computers. Although client computers 110 are shown separate from the server computers 120, it is understood that a single computer might perform the client and server roles. Those skilled in the art will appreciate that the computer environment 100 shown in FIG. 1 is intended to be merely illustrative. The present invention may also be practiced in other computing environments. For example, the present invention may be practiced in multiple processor environments wherein the client computer includes multiple processors. Moreover, the client computer need not include all of the input/output devices as discussed above and may also include additional devices. Those skilled in the art will appreciate that the present invention may also be practiced via Intrarowns and more generally in distributed environments in which a client computer requests resources from a server computer.

During operation of the distributed system 100, users of the clients 110 may desire to access information records 122 stored by the servers 120 while utilizing, for example, the Web. Furthermore, such server systems 120 may also include one or more search engines having one or more databases 124. The records of information 122 may be in the form of Web pages 150. The pages 150 may be data records including as content plain text information, or more complex digitally encoded multimedia content, such as software programs, graphics, audio signals, videos, and so forth. It should be understood that although this description focuses on locating information on the World-Wide-Web, the system may also be used for locating information via other wide or local area networks (WANS and LANS), or information stored in a single computer using other communications protocols.

The clients 110 may execute Web browser programs 112, such as Netscape Navigator or MSIE to locate the pages or records 150. The browser programs 112 enable users to enter addresses of specific Web pages 150 to be retrieved. Typically, the address of a Web page is specified as a URI or more specifically as a URL. In addition, when a page has been retrieved, the browser programs 112 may provide access to other pages or records by “clicking” on hyperlinks (or links) to previously retrieved Web pages. Such links may provide an automated way to enter the URL of another page, and to retrieve that page.

FIG. 1b illustrates a block diagram of a processor 166 coupled to a storage device such as memory 168 and to input/output devices 170 in a client 110 and/or server 120 computing system. Stored in memory 168 may be information records 122 having any combination of exemplary content such as lists, files, and databases. Such records may include for example: user modifiable configuration settings 174, identifier generation routines 176, FDN registry 178, TLD cache 180, prefix database 182, Templates 184, GO List 186, name translation table 188, and advertising cache 190. These information records may be further introduced and discussed in more detail throughout the disclosure of this invention.

FIG. 2 is a flowchart illustrating the steps performed by a prior art system for locating a network resource from an identifier by accessing a URI. A device such as a network access apparatus 110, servlet, applet, stand-alone executable program, or user interface element such as a text box object, command line, speech to text interface, location field 814 of a web browser 112, may receive and parse input such as text or voice in step 210. It then may be determined in step 215 whether the input 210 is a URI. If the input is a URI, then an attempt may be made in step 220 to access the URI. The URI may be accessed when it is determined in step 222 that the URI is accessible. If the input is not a URI, then it may be determined in step 225 whether a URI can be generated from the input (e.g., if a scheme is missing the prefix “http://” or another scheme prefix may be concatenated to input). If so, then a URI is generated 230 and an attempt may be made in step 220 to access the URI. If the URI of not be generated or accessed, then a message indicating that the URI cannot be generated or accessed may be displayed in step 235.

FIG. 3 is a flowchart illustrating the steps performed for generating a valid URI. When it is determined in step 225 that a URI may not be generated from the input 210 in accordance with methods known to one of ordinary skill in the art, then it may be determined in step 310 whether it is configured to determine whether other methods may be applied to access a valid URI corresponding to input. If configuration 174 is not enabled, then a message indicating that the URI cannot be generated may be displayed in step 325. However, when configuration is enabled, then at least one other method may be applied in step 315 in sequence or parallel including methods explained in U.S. patent application Ser. No. 09/552,500 filed Mar. 21, 2000, by Schneider, entitled “Fictitious domain name method, product, and apparatus” such as retrieving information from user modifiable configuration settings 174, template 184, GO LIST 186, name translation table 188, and registry 178 (e.g., processing a FDN, processing a TLD as a search request, and multiple accessibility from a plurality of generated URls, etc.).

FIG. 4 is a flowchart illustrating the steps performed for generating a valid URI in response to determining that an existing valid URI is inaccessible. When a valid URI is determined in step 222 to be not accessible, then a valid URI may be generated in step 410 to access content by extracting keywords from the non-query component URI (e.g., directory, domain, port, or fragment, etc.) of the inaccessible URI and using such keywords as a search request or as the query portion of the generated valid URI. After valid URI generation in step 410 an attempt may be made in step 220 to access the URI.

In effect, the non-query portion of a first URI may be used in the query portion of a second URI to generate dynamic content. Validity of URI syntax is explained in T. Berners-Lee, “Informational RFC (Request for Comment) 1630: Universal Resource Identifiers in WWW—A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web”, Internet Engineering Task Force (IETF), June 1994, “http://www.iana.org/assignments/rfc1630.html”, which is herein incorporated by reference. Typically, when a HTTP request is submitted to access a URI that does not exist, an error 404 code is returned from the requested web server, and in turn, a script may be executed to redirect and display a gen-eric web page informing the user of the error code. A modification may be made to the script to generate an accessible URI by creating the appropriate directories or files on the web server in response to an error 404 code (or any other applicable error code from the 300 series, 400 series, or 500 series) such that when the HTTP request is resubmitted the URI exists and is accessible.

FIG. 5 is a flowchart illustrating the steps performed for generating a frame having an accessible URI. After valid URI generation in step 410, a script may be executed in step 510 to create a second accessible URI by adding the appropriate directories and files. The content of the newly created file may include at least one frame that displays the content of the valid generated URI. After the second accessible URI is created, an attempt may be made in step 220 to access the second URI.

FIG. 6 is a flowchart illustrating the steps performed for creating files or directories as needed to form an accessible URI. When it is determined in step 222 that a valid URI is not
whether the URI has the minimum form of “scheme://
SLD.TLD/FLD” where the domain name is “SLD.TLD” and
the path is “/FLD”, which in this case represents a first level
directory (FLD). If the URI is not of the minimum form then
steps (such as 410 and/or 510) may be performed. When it is
determined in step 610 that the URI is of the minimum form,
then it may be determined in step 615 whether the FLD exists.
If the FLD exists, then it may be further determined in step
620 whether a file exists within the FLD. If a file exists, then
it may be determined in step 625 whether the content of the
file requires modification (e.g., META tag for purpose of URI
redirection). If modification is needed then the content of the
file may be modified in step 630 as necessary. If the FLD does
not exist, then a FLD may be created in step 635. When the
FLD is created or when the FLD exists but a file within the
FLD does not exist, then a file within the FLD may be created
in step 640. When it is determined in step 620 that the file
already exists and does not need modification in step 625 or
the file is created in step 640 or the existing file is modified in
step 635 then a second URI may be generated in step 645
including the FLD as part of a search request. After the accessible URI is created, an attempt may be made in step 220
to access the URI.

For example, a device receives the input “http://update/ to/ news”. When it is determined that the input is a valid URL
(URL is a subset of a URI), an HTTP GET request is submitted
to retrieve its content. A “404 error code” is received in
response to the request indicating that no such URL exists. As
practiced by those skilled in the art, the “404 error code” is
commonly redirected to display a gen-eric informative page
alerting the user that no such URL exists. Advantage may be
taken by modifying the redirect script.

The modified script may be executed upon the “404 error code” and determines that a “news” directory does not exist
and is created. Furthermore, an “index.htm” is created in the
“news” directory serving as a default file for the URL listed
above. The content of the file includes a <META> tag which
may be used to redirect the URL to another URL such as
“http://update/cgi-bin/update.cgi?search=news” which
serves as a dynamic output.

For instance, the input “update.to/news” becomes equiva-
 lent to “http://update/to/news/index.htm” and includes within
the “index.htm” file the following <META> tag:

```
<META HTTP-EQUIV=Refresh CONTENT="5; URL=http://update/to/cgi-bin/update.cgi?search=news"/>
```

which communicates to a device that interprets HTML to load the
index.htm” has finished loading. Advertising may be dis-
pplayed by selecting an ad from an advertising cache 190 that
corresponds to any URI components or the like before automatic
URI redirection. Advertisement selection is explained in
U.S. Provisional Application Ser. No. 60/153,594 filed
Sep. 13, 1999, by Schneider entitled “Method and apparatus
for using a portion of a URI to select and display advertising.”

In another example, the content of the file may include a
<FRAME> tag, which may be used to access content from
“http://update/to/cgi-bin/update.cgi?search=news”.

In effect, shorter URLs may be generated in real time or
on-the-fly when necessary and used as a substitute or proxy
for longer URLs. Though any filename may be used, it is a
preferred aspect to use “index.htm” as a frame or redirect so
the proxy URL is even shorter in string length. A shorter URL
makes it easier for a user to remember the URI for future use.
The invention is not limited to using only a FLD as a search request. Any directory or combination of directory levels may be
either generated on the fly or used as a search request. In
addition to using the directory portion of a URI as a search request, a domain (e.g., SLD) or any combination of domain
levels may in turn be used to generate a query as well. For
example, a subdomain may be generated to create the FQDN
“news.update.to” having the minimum form of “scheme://
SLD.SLD.TLD”, which may be used to access content from
“http://update/to/cgi-bin/update.cgi?search=news”.

Analysis from major online search engines show that more
than 90% of all search requests include three keywords or
less. For instance, the search request “coffee and cream and
sugar” can yield “http://example.com/coffee/cream/sugar/in-
dex.htm”. In turn, the input “example.com/coffee/cream/
sugar” can generate search results thereby combining the step
of searching by entering in a web address or URI to search.

Boolean logic can be applied to extend variations for simple
search requests. For example, the search request “coffee or
cream or sugar” yields “http://example.com/coffee/cream.
sugar/index.htm” whereas the search request “coffee and cream
and sugar” yields “http://example.com/coffee/cream/sugar/index.htm”. In effect, URI notation can be used as a sys-
tem to generate a Boolean search request from the URI or
vice versa.

FIG. 7 is a flowchart illustrating the steps performed for
extracting a domain to be used as a search request. When the
input 210 is a valid 215 URI or a valid URI is generated in step
230 and the configuration settings 174 indicate that domain
identifier extraction may be used (e.g., SLD) then a script
may be executed to extract in step 710 the SLD from the valid
URI (215, 230) and a second URI may be generated in step 715 by
using the SLD as part of a search request. The script generates
a web page having two frames. The first frame links to the
valid URI and the second frame links to the second URI
which displays content when accessed relating to the use
of SLD as a keyword in a search request.

FIG. 8 is a diagram depicting how results may be displayed
in a web browser. A client 110 web browser 112 having a web
page 810 is used to connect to a server 120 via the Internet 130
that runs a CGI script 812. The location field of the web
browser 112 is suppressed and the web page 810 displays at
least two frames. The first frame is the web based location
field 814 and the second frame 816 is used to display the
contents 150 of a web address. An input device (e.g., key-
board, mouse, pen light, touch screen, or microphone, etc.)
of a client computer or network access apparatus 110 is used
to receive a web address as input either directly from a hyperlink
(not shown) in the web page 810, or from the location field
814 of the web page 810. A URI GET request is generated
from the input and the browser 112 forwards the request to
a server 120, which processes the request by executing a CGI
script 812 to determine accessibility 204. An accessible URI
206 may be generated and the requested page 150 is sent to
the browser 112. The content 150 of the URI may be displayed
(as discussed in FIG. 6 with a specific example) in the
second frame 816 of the web page 810. The location field
814 of the first frame may either persist by displaying the input or
is cleared out for entry of another web address.

Another web page 810 (as discussed in FIG. 7) having a
location field that is not suppressed may be displayed by using
a different CGI script 812. A URI GET request may be
generated from the input and the browser 112 forwards the
request to a server 120, which processes the request by
executing a CGI script 812 to determine accessibility. An
accessible URI may be generated and the requested page 150
is sent to the browser 112. The content 150 of the URI may be
displayed in a first frame 818 of the web page 810 whereas
the content of the second frame 820 may be dynamically
generated by corresponding the extracted SLD from the URI
as a search term in a second URI. By using the steps illustrated
in FIG. 7, when a first URI “http://www.example.com” is
received as input, a script extracts “example” from the URI and generates a second URI \textasciitilde \texttt{http://search.yahoo.com/bin/search?p=example"}. Both the first URI and second URI are generated as frames and displayed as a web page \textasciitilde \texttt{810}. Though the above aspects demonstrate how URIs may be accessed based upon a web-based version of a location field, similar teachings may be applied to those skilled in the art by providing a user interface element such as a text box object as input. The text box object may be located anywhere and on the web page including a text box that may be embedded or displayed as part of an on-line advertisement. The text box object may be used in a stand-alone application or stored on magnetic and/or optical media that may be non-volatile, writable, removable, or portable. The text box object may be incorporated as an applet or servlet and embedded in other applications. The text box object may be integrated in the task bar or any part of the GUI’s OS, or the OS bypassed and a user interface element overlaid as a graphic on a display device based on modifications of a video card and/or it’s associated firmware or software drivers. A command line text box may be further overlaid as an interactive object in other embodiment such as Internet television, cable television, digital television, or interactive television through an Internet appliance or set top box.

Those skilled in the art may make and use software programs that functions as a browser plug-in. Such a program may be downloaded and installed for integration into the command line of a device or location field \texttt{154} of a browser program \texttt{112}. Modifying the source code of the browser program \texttt{112} itself may be more desirable, in effect, enabling tens of millions of users to take advantage of more creative ways to use input as a means to access a valid URI.

Although the invention has been shown and described with respect to a certain preferred aspect or aspects, it is obvious that equivalent alterations and modifications will occur to others skilled in the art upon the reading and understanding of this specification and the annexed drawings. In particular regard to the various functions performed by the above described items referred to by numerals (components, assemblies, devices, compositions, etc.), the terms (including a reference to a “means”) used to describe such items are intended to correspond, unless otherwise indicated, to any item which performs the specified function of the described item (e.g., that is functionally equivalent), even though not structurally equivalent to the disclosed structure which performs the function in the herein illustrated exemplary aspect or aspects of the invention. In addition, while a particular feature of the invention may have been described above with respect to only one of several illustrated aspects, such feature may be combined with one or more other features of the other aspects, as may be desired and advantageous for any given or particular application.

The description herein with reference to the figures will be understood to describe the present invention in sufficient detail to enable one skilled in the art to utilize the present invention in a variety of applications and devices. It will be readily apparent that various changes and modifications could be made therein without departing from the spirit and scope of the invention as defined in the following claims.

What is claimed is:

1. A method for generating newly created information comprising:
   - receiving a user request at a computing device, the user request comprising a Uniform Resource Identifier (URI) comprising at least one of a scheme, one or more domains, port, path, one or more directory names, and one or more file names;
   - determining whether the user request corresponds to a valid existing resource available from the computing device;
   - determining that said file can not be located on said server;
   - extracting at least one keyword from said at least one of a scheme, one or more domains, port, path, one or more directory names, and one or more file names;
   - generating and performing a search request having at least one keyword to at least one Internet search engine provider to obtain at least one web page including said at least one keyword; and,
   - dynamically generating and providing said newly created information to said user wherein said newly created information includes results of said search request from said Internet search engine provider.

2. The method, as set forth in claim 1, wherein said URI can be represented as “scheme://SLD.TLD/FLD/index.htm” whereby SLD.TLD is a domain name, FLD is a first level directory path, and index.htm is a default file.

3. The method, as set forth in claim 2, further including creating said FLD and said default file when it is determined that said file can not be located on said server.

4. The method, as set forth in claim 3, wherein said default file is capable of dynamically providing said newly created information to said user.

5. The method, as set forth in claim 1, wherein said URI can be represented as “scheme://3LD.SLD.TLD/index.htm” whereby SLD.TLD is a domain name, 3LD is a subdomain of said domain name, and index.htm is a default file.

6. The method, as set forth in claim 5, further including creating said 3LD and said default file when it is determined that said file can not be located on said server.

7. The method, as set forth in claim 6, wherein said default file is capable of dynamically generating said newly created information to said user.

8. The method, as set forth in claim 1, wherein said search request includes [Boolean] Boolean logic translated from said path of said URI.

9. The method, as set forth in claim 1, further including providing at least one advertisement corresponding to said at least one keyword before dynamically generating said newly created information to said user.

10. A method for presenting information, the method comprising:
   - receiving a user request at a computing device, the user request comprising a Uniform Resource Identifier (URI) comprising at least one of a scheme, one or more domains, port, path, one or more directory names, or one or more file names;
   - determining whether the user request corresponds to a valid existing resource available from the computing device;
   - the computing device extracting at least one keyword from the at least one of a scheme, one or more domains, port, path, one or more directory names, or one or more file names in response to determining the user request does not correspond to a valid existing resource;
   - the computing device generating a search request based on at least the at least one keyword to obtain an identification of at least one resource corresponding to at least one keyword; and,
   - the computing device communicating the identification of the at least one resource corresponding to the at least one keyword.

11. The method according to claim 10, wherein generating a search request comprises generating a search request to an Internet search engine, and wherein communicating the identification of at least one resource comprises communicating over the Internet.

12. The method according to claim 10, wherein the URI comprises a representation comprising “HLD.TLD/DP” whereby HLD.TLD is a domain name comprising a top level
domain (TLD) and at least one hierarchal second or higher level domain name (HLD), and DP is a directory path.

13. The method according to claim 1, further comprising creating the DP and a resource corresponding to the URI in response to the URI not corresponding to a valid existing resource.

14. The method according to claim 13, wherein the resource is adapted to dynamically provide newly created information.

15. The method according to claim 10, wherein the URI comprises a representation comprising “3LD.SLD.TLD”, whereby SLD.TLD is a domain name comprising a top level domain name (TLD) and a second level domain name (SLD), and 3LD is a subdomain of the second level domain name.

16. The method according to claim 15, further comprising creating the 3LD and a resource in response to the URI not corresponding to a valid existing resource.

17. The method according to claim 16, wherein the resource is capable of dynamically generating newly created information.

18. The method according to claim 10, wherein the query comprises Boolean logic translated from a path of the URI.

19. The method according to claim 10, further comprising providing information relating to at least one advertisement relating to the at least one keyword.

20. A non-transitory tangible computer readable medium having instructions stored thereon, the instructions comprising:

instructions for receiving a user request to locate a file, the user request including a Uniform Resource Identifier (URI) having at least one of a scheme, one or more domains, port, path, one or more directory names, or one or more file names;

instructions for determining that the file cannot be located on a server;

instructions for extracting at least one keyword from the at least one of a scheme, one or more domains, port, path, one or more directory names, or one or more file names;

instructions for generating a search request having the at least one keyword to obtain at least one web page; and,

instructions for dynamically generating and providing newly created information wherein the newly created information comprises, at least in part, information that leads to the at least one web page.

21. A non-transitory tangible computer readable medium having instructions stored thereon that cause a computing device to perform operations comprising:

determining whether a user request comprising a Uniform Resource Identifier (URI) comprising at least one of a scheme, domain, port, path, directory name, or file name corresponds to an existing resource;

extracting at least one keyword from the at least one of a scheme, domain, port, path, directory name, or file name in response to determining that the user request does not correspond to an existing resource;

generating a query based on at least the at least one keyword to obtain an identification of at least one resource corresponding to the at least one keyword; and,

communicating the identification of the at least one resource corresponding to the at least one keyword.

22. The non-transitory tangible computer readable medium according to claim 21, wherein generating a query comprises generating a search request to an Internet search engine and communicating the identification of the at least one resource comprises communicating via the Internet.

23. The non-transitory tangible computer readable medium according to claim 21, wherein the URI comprises a representation comprising “3LD.SLD.TLD”, whereby SLD.TLD is a domain name comprising a top level domain name (TLD) and at least one hierarchal second or higher level domain name (HLD), and DP is a directory path.

24. The non-transitory tangible computer readable medium according to claim 23 having instructions stored thereon that cause the computing device to perform operations further comprising:

creating the DP and a resource corresponding to the URI in response to determining that the URI does not correspond to an existing resource.

25. The non-transitory tangible computer readable medium according to claim 21 having instructions stored thereon that cause the computing device to perform operations further comprising:

providing information relating to at least one advertisement relating to the at least one keyword.

26. A system, comprising:

at least one interface adapted to receive a user request, the user request comprising a Uniform Resource Identifier (URI) comprising at least one of a scheme, domain, port, path, directory name, or file name; and

at least one computing device adapted to:

determine whether the user request corresponds to an available resource;

extract at least one keyword from the at least one of a scheme, domain, port, path, directory name, or file name in response to determining that the request does not correspond to an available resource;

generate a query based on at least the at least one keyword to obtain an identification of at least one resource corresponding to the at least one keyword; and,

communicate through the at least one interface, at least one of the query or the identification of at least one resource corresponding to the at least one keyword.

27. The system according to claim 26, wherein generating a query comprises generating a search request to an Internet search engine, and communicating at least one of the query or the identification of the at least one resource comprises communicating over the Internet.

28. The system according to claim 26, wherein the URI comprises a representation comprising “HLD.TLD/DP”, whereby HLD.TLD is a domain name comprising a top level domain name (TLD) and at least one hierarchal second or higher level domain name (HLD), and DP is a directory path.

29. The system according to claim 26, wherein the computing device is further adapted to create the DP and a new resource corresponding to the URI in response to the computing device determining that the URI does not correspond to an available resource.

30. The system according to claim 29, wherein the computing device is adapted to create the new resource to be adapted to dynamically provide newly created information.

31. The system according to claim 26, wherein the URI comprises a representation comprising “3LD.SLD.TLD”, whereby SLD.TLD is a domain name comprising a top level domain name (TLD) and a second level domain name (SLD), and 3LD is a subdomain of the second level domain name.

32. The system according to claim 31, wherein the computing device is adapted to create the 3LD and a default resource in response to the computing device determining that the URI does not correspond to an available resource.

33. The system according to claim 32, wherein the computing device is adapted to create the default resource to be adapted to dynamically generate newly created information.

34. The system according to claim 26, wherein the computing device is adapted to generate the query comprising Boolean logic translated from a path of the URI.

35. The system according to claim 26, wherein the computing device is further adapted to:

provide information relating to at least one advertisement relating to the at least one keyword.
On the Title Page, item (56), under “OTHER PUBLICATIONS”, in Column 2, Line 1, delete “the Telephone” and insert -- The Telephone --, therefor.

On Page 5, item (56), under “OTHER PUBLICATIONS”, in Column 2, Line 12, delete “al.:” and insert -- al., --, therefor.


On Page 5, item (56), under “OTHER PUBLICATIONS”, in Column 2, Line 25, delete “Schol.” and insert -- School, --, therefor.

On Page 5, item (56), under “OTHER PUBLICATIONS”, in Column 2, Line 38, delete “new_pr_51700.html.” and insert -- new_pr_51700.html. --, therefor.

In Column 2, Line 47, delete “URLs” and insert -- URLs --, therefor.

In Column 12, Line 34, in Claim 8, delete “said” and insert -- [said] a --, therefor.

Signed and Sealed this Twenty-sixth Day of February, 2013

Teresa Stanek Rea
Acting Director of the United States Patent and Trademark Office