

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

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
9 October 2003 (09.10.2003)

PCT

(10) International Publication Number
WO 03/083676 A1

(51) International Patent Classification⁷: **G06F 13/00**

(21) International Application Number: PCT/US03/08434

(22) International Filing Date: 20 March 2003 (20.03.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/104,863 22 March 2002 (22.03.2002) US

(71) Applicant: **CITRIX SYSTEMS, INC.** [US/US]; 851 West Cypress Creek Road, Fort Lauderdale, FL 33309 (US).

(72) Inventor: **KRAMER, Andre**; 55 Victoria Park, Cambridge CB4 3EJ (GB).

(74) Agent: **ABRAMSON, Andrew, F.**; Testa, Hurwitz & Thibault, LLP, High Street Tower, 125 High Street, Boston, MA 02110 (US).

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

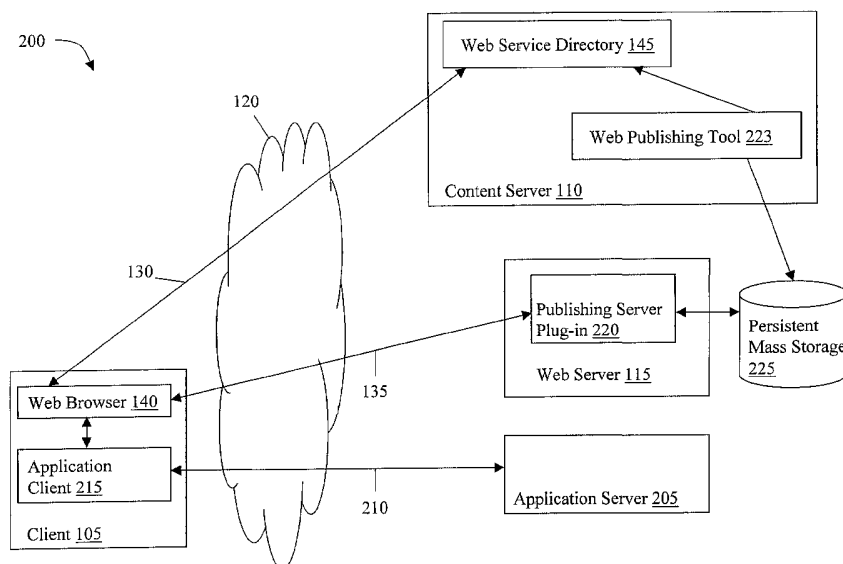
(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ,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

Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

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

(54) Title: METHODS AND SYSTEMS FOR PROVIDING ACCESS TO AN APPLICATION



(57) Abstract: The invention enables the publishing (220) of a graphical user interface application on the web (120) in a manner so that they can be discovered by manual or automatic searches. The invention includes the step of receiving a service access point (SAP) from a web service directory (145). The SAP is associated with a first application and identifies a web server (115). The invention also includes the step of retrieving address information associated with the first application from the web server identified by the SAP and launching a second application. The second application establishes a communication channel with an application server (205) identified by the retrieved address information. The application server (205) subsequently executes the first application and returns information to the second application.



WO 03/083676 A1

METHODS AND SYSTEMS FOR PROVIDING ACCESS TO AN APPLICATION

Field Of The Invention

[0001] The present invention relates generally to publishing web services on the web and, more specifically, to providing access to an application using a web service directory.

Background of the Invention

5 **[0002]** Many companies having web sites on the World Wide Web enrich the experience of visiting their sites by using content from other service providers or web sites. For instance, a web portal can integrate numerous services from other web sites, such as travel information and search capabilities. The supplier of these services, such as the particular travel company providing the travel information, can benefit from greater exposure and/or an
10 increase in income. The integration of these services, also called web services, typically enables the web portal to offer more services to its users.

[0003] Referring to FIG. 1, a computer system 100 known to the prior art typically includes a client computer 105, or client, a content server 110, and a web server 115. The client 105 is typically a personal computer that can download content from the content server
15 110 and/or web server 115 over a network 120, such as the Internet or World Wide Web. The client 105 is typically in communication with the content server 110 over a client-content server communication channel 130. Moreover, the client 105 is typically in communication with the web server 115 over a client-web server communication channel 135.

20 **[0004]** The client 105 may also include a web browser 140, such as INTERNET EXPLORER developed by Microsoft Corporation in Redmond, Washington or NETSCAPE NAVIGATOR developed by Netscape Communications Corporation of Mountain View, California, to connect to the web and/or download content from the servers 110, 115. A conventional content server 110 hosts one or more programs or files that can be accessed by
25 the client 105. The web server 115 typically delivers web pages to the client 105 (e.g., web browser 140).

[0005] To advertise web services, firms or companies can publish their services on the web using the content server 110. As more and more web services are written, a central

- 2 -

repository (i.e., depository) for public web services, or a web service directory 145, is typically stored on the content server 110. The web service directory 145 may include a table listing supported web services and may also include documentation for each listed web service.

5 **[0006]** One example of the web service directory 145 includes the Universal Description, Discovery, and Integration (UDDI) registry, jointly developed by Microsoft Corporation of Redmond, Washington, International Business Machines Corporation of Armonk, New York, and Ariba Inc. of Sunnyvale, California. Web services can alternatively be published on the web service directory 145 using the Lightweight Directory Access Protocol (LDAP),
10 Microsoft's ACTIVE DIRECTORY, developed by Microsoft Corporation of Redmond, Washington, Common Object Request Broker Architecture (CORBA) name service, or ebXML service registry developed by the Organization for the Advancement of Structured Information Standards (OASIS) of Billerica, Massachusetts.

15 **[0007]** A potential user of the conventional web service uses the client 105 to access the web service directory 145. The client 105 typically accesses the web service through a programming interface. For example, the service may be encoded by an interface description language, such as Web Services Description Language (WSDL). Alternatively, the client 105 may access the web service through a web page or a web page address (i.e., a Uniform Resource Locator, or URL).

20 **[0008]** There are several drawbacks to using the conventional techniques of accessing a web service. Programmatic service interfaces, for instance, typically require client-side applications (having client-side application logic) to be developed by the supplier of the web service to ensure that the client 105 can properly execute the web service through its interface. Moreover, web pages typically require an application to have a web user interface
25 (UI) to access the web page. A web UI can limit the amount of interactivity between a user of the client 105 and the web page. For example, the amount of interactivity can be limited by delays in loading the next screen or object. Additionally, using a web UI for the execution of an application may constrain the complexity of the interface.

30 **[0009]** Therefore, there is a need to access web services in a more user-friendly manner, such as by providing greater interactivity between the client 105 and the web service and less dependence on suppliers of web services for applications to correctly execute with the web service.

Summary of the Invention

[0010] The invention relates to methods and systems for providing access to a graphical user interface (GUI) application using web services. Providing access to, or publishing, a GUI rather than a web UI or a programming service interface has several benefits to the user of a client. For example, using a GUI to access a web service can provide a more powerful interface than a web UI and a more user-friendly environment. Additionally, using a GUI can provide a general UI to a user of the client compared to different web UIs for different web pages. Further, an application executing with the GUI does not need client-side application logic to properly execute on the client.

[0011] In one aspect, a method for remoting an application to an application client or end user application includes the step of receiving a service access point (SAP) from a web service directory. The SAP is associated with a first application and identifies a web server. The method also includes retrieving address information associated with the first application from the web server identified by the SAP and launching a second application. The second application establishes a communication channel with an application server identified by the retrieved address information.

[0012] In one embodiment, the method also includes the step of transmitting a query to the web service directory to search for the first application. The method can also include receiving the address of the web server and receiving a key for locating information associated with the first application in a database. In another embodiment, the step of retrieving address information from the web server also includes the step of retrieving a Multipurpose Internet Mail Extensions (MIME) document including the address information.

[0013] In another aspect, a client for use with executing a remote first application provided by a service provider on a web service directory includes a client browser. The client browser receives a SAP associated with the first application from the web service directory. The SAP identifies a web server. The client browser also retrieves address information associated with the first application from the web server identified by the SAP. The client also includes an application client that is launched in response to the retrieved address information. The application client also establishes a communication channel with an application server identified by the retrieved address information to receive data from the first application.

- 4 -

[0014] In another aspect, a method for remoting an application to an application client or end user application includes the steps of sending a query to a web service directory for information associated with a first application and transmitting a SAP associated with the first application to the client from the web service directory. The SAP identifies a web server. The method also includes the steps of retrieving address information associated with the first application from the web server identified by the SAP and launching a second application by the client. The second application establishes a communication channel with an application server identified by the retrieved address information. The method additionally includes the step of launching the first application by the application server and returning information to the second application. In one embodiment, the method additionally includes accessing a local database using a key contained within the SAP and having the local database return address information associated with the first application.

[0015] In yet another aspect, a method for remoting an application to an application client or end user application includes the step of receiving a request to access a SAP from a client. The SAP is associated with a first application. The method also includes identifying that the requested SAP can be used with an application client and retrieving address information associated with the first application identified by the SAP. The method additionally includes the step of converting the retrieved address information into a file where the format of the file is associated with the application client. The method also includes transmitting the file including the converted address information to the client.

[0016] In one embodiment, the SAP is identified in a web service directory. In other embodiments, the method includes transmitting the SAP associated with the first application to the client from the web service directory. In some embodiments, the method includes launching an application client by the client. The application client may also establish a communication channel with an application server identified by the retrieved address information.

Brief Description of the Drawings

[0017] The advantages of the invention described above, together with further advantages, may be better understood by referring to the following description taken in conjunction with the accompanying drawings. In the drawings, like reference characters generally refer to the same parts throughout the different views. Also, the drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the invention.

- 5 -

[0018] FIG. 1 is a block diagram of an embodiment of a prior art computer system.

[0019] FIG. 2 is a block diagram of an embodiment of a computer system having a publishing server plug-in.

[0020] FIG. 3 is a flow diagram illustrating an embodiment of the operation of the computer system of FIG. 2.

Detailed Description of the Invention

[0021] Referring to FIG. 2, an embodiment of a computer system 200 enabling the publishing of GUIs in a web service directory includes the client 105, the content server 110, the web server 115, and an application server 205. The client 105 can download content from the content server 110, the web server 115, and the application server 205 over the network 120. In one embodiment, the client 105 can download content (e.g., an application) from the application server 205 over a client-application server communication channel 210.

[0022] Although illustrated with three servers 110, 115, 205, any number of servers can be used to implement the functions described above and below. In one embodiment, two or more of the servers 110, 115, 205 are aggregated together and provided as a single physical machine. Moreover, any number of clients 105 can be connected to the servers 110, 115, 205 to implement the functions described above and below.

[0023] The client 105 can be any personal computer (e.g., based on a microprocessor from the x86 family, the Pentium family, the 680x0 family, PowerPC, PA-RISC, MIPS families), smart or dumb terminal, network computer, wireless device, information appliance, workstation, minicomputer, mainframe computer or other computing device. Operating systems supported by the client 105 can include any member of the WINDOWS family of operating systems from Microsoft Corporation of Redmond, Washington, MacOS, JavaOS, and various varieties of Unix (e.g., Solaris, SunOS, Linux, HP-UX, A/IX, and BSD-based distributions).

[0024] In one embodiment, the web browser 140 uses Secure Socket Layer (SSL) support for communications to the content server 110 and/or the web server 115. SSL is a secure protocol developed by Netscape Communication Corporation of Mountain View, California, and is now a standard promulgated by the Internet Engineering Task Force (IETF). The web browser 140 can alternatively connect to the content server 110 and/or the web server 115 using other security protocols, such as, but not limited to, Secure Hypertext Transfer Protocol (SHTTP) developed by Terisa Systems of Los Altos, CA, HTTP over SSL (HTTPS), Private

- 6 -

Communication Technology (PCT) developed by Microsoft Corporation of Redmond, Washington, and the Transport Level Security (TLS) standard promulgated by the IETF.

[0025] Additionally, the client 105 includes an application client 215 for establishing and exchanging communications with the application server 205 over the client-application server communication channel 210. In one embodiment, the application client 215 is a GUI application. In some embodiments, the application client 215 is an Independent Computing Architecture (ICA) client, developed by Citrix Systems, Inc. of Fort Lauderdale, Florida, and is also referred to below as ICA client 215. Other embodiments of the application client 215 include a Remote Display Protocol (RDP) client, developed by Microsoft Corporation of Redmond, Washington, a client-side player, interpreter or simulator capable of executing multimedia applications, email, Java, or .NET code. Moreover, in one embodiment the output of an application executing on the application server 205 can be displayed at the client 105 via the ICA client 215.

[0026] The client 105 searches the web service directory 145 for a web service. In one embodiment, the search is a manual search. Alternatively, the search is an automatic search. The web service directory 145 may also provide a service based view, such as white and yellow pages, to search for web services in the web service directory. In another embodiment, the web service directory 145 supports a hierarchical browsing based on a structured service name and service kind for GUI applications. In one embodiment, the web service directory 145 executes on a server independent of the content server 110, such as a directory server. In other embodiments, the web service directory 145 executes on multiple servers.

[0027] In some embodiments, the content server 110 enables the client 105 to select web services based on additional analysis or information by providing this information or analysis in the web service directory 145. Examples of service information that the web service directory 145 can list includes, but is not limited to, the name of the business offering the service, the service type, a textual description of the service, one or more service access points (SAPs), the network type, the path to use (e.g., TCP or HTTPS), and quality of service (QoS) information. Moreover, service information can be client device type or user (e.g., role) specific. Thus, service selection can be based on one or more of the above attributes.

[0028] In one embodiment, the service type denotes a programming interface that the client 105 must use to access the web service. For instance, the service type can state that the

- 7 -

service is encoded by an interface description language, such as Web Services Description Language (WSDL).

[0029] The service access point, or SAP, is a unique address for an application. The SAPs enable the computer system 200 to support multiple applications at the client 105 and each server 110, 115, 205. For example, the application server 205 may support an electronic mail (i.e., e-mail) application, a file transfer application, and/or a GUI application. In one embodiment, these applications would each have a SAP that is unique within the application server 205. In one embodiment, the SAP is a web or Internet address (e.g., Domain Name System (DNS) name, IP/port, or Uniform Resource Locator (URL)). Thus, in one embodiment the SAP identifies the address of the web server 115 as part of the address for an application stored on the web server 115. In some embodiments, the SAP identifies the address of a publishing server plug-in 220 as part of the address for an application stored on the web server 115, as described below. In one embodiment, the SAP is an "accessPoint" from the UDDI registry.

[0030] To prepare an item for publishing in the web service directory 145, the content server 110 includes a web publishing tool 223. In one embodiment, the web publishing tool 223 is a software module. Alternatively, the web publishing tool 223 is another server that may be externally located from or internally located in the content server 110.

[0031] In one embodiment, the web server 115 delivers web pages to the client 105. The web server 115 can be any personal computer (e.g., based on a microprocessor from the x86 family, the 680x0 family, the Pentium family, PowerPC, PA-RISC, MIPS families), Windows-based terminal, network computer, wireless device (e.g., cellular phone), information appliance, X-device, workstation, mini computer, main frame computer, personal digital assistant, or other communications device capable of providing web pages to the client 105.

[0032] In another embodiment, the web server 115 is an Enterprise Information Portal (e.g., corporate Intranet or secured business-to-business extranet). Enterprise portals are company web sites that aggregate, personalize and serve applications, data and content to users, while offering management tools for organizing and using information more efficiently. In some companies, portals have replaced traditional desktop software with browser-based access to a virtual workplace.

- 8 -

[0033] The web server 115 also includes a publishing server plug-in 220 to enable the publishing of graphical user interface (GUI) applications. More specifically, the publishing server plug-in 220 translates a new web service entry URL into a GUI application service so that the GUI can be accessed via the web service directory 145. In one embodiment, the publishing server plug-in 220 is a Common Gateway Interface (CGI) script, which is a program designed to accept and return data that conforms to the CGI specification. The program can be written in any programming language, such as C, Perl, Java, or Visual Basic. In another embodiment, the publishing server plug-in 220 is a Java Server Page (JSP). Using the publishing server plug-in 220 to facilitate the publishing of remote GUI applications, the client 105 can thereby access the web service, not through a programming interface or a web page, but through a full GUI interface, such as with Citrix's ICA or Microsoft's RDP.

[0034] The application server 205 hosts one or more applications that are available for the client 105. Examples of such applications include word processing programs such as MICROSOFT WORD and spreadsheet programs such as MICROSOFT EXCEL, both manufactured by Microsoft Corporation of Redmond, Washington, financial reporting programs, customer registration programs, programs providing technical support information, customer database applications, or application set managers.

[0035] The network 120 can be a local-area network (LAN), a wide area network (WAN), or a network of networks such as the Internet or the World Wide Web (i.e., web). In other embodiments, one or more of the communication channels 130, 135, 210 are established over different networks. For example, the client-content server communication channel 130 can belong to a first network (e.g., the World Wide Web) and the client-web server communication channel 135 can belong to a second network (e.g., a secured extranet or Virtual Private Network (VPN)).

[0036] Example embodiments of the communication channels 130, 135, 210 include standard telephone lines, LAN or WAN links (e.g., T1, T3, 56kb, X.25), broadband connections (ISDN, Frame Relay, ATM), and wireless connections. The connections over the communication channels 130, 135, 210 can be established using a variety of communication protocols (e.g., HTTP, HTTPS, TCP/IP, IPX, SPX, NetBIOS, Ethernet, RS232, messaging application programming interface (MAPI) protocol, real-time streaming protocol (RTSP), real-time streaming protocol used for user datagram protocol scheme (RTSPU), the Progressive Networks Multimedia (PNM) protocol developed by

RealNetworks, Inc. of Seattle, WA, manufacturing message specification (MMS) protocol, and direct asynchronous connections).

[0037] In one embodiment, the web publishing tool 223 stores information about an application that the web publishing tool 223 is publishing in the web service directory 145 in a persistent mass storage 225. As discussed below, in one embodiment the information is a URL for the dynamic publishing server plug-in 220. The persistent mass storage 225 may be a magnetic disk or magneto-optical drive. In one embodiment, the persistent mass storage 225 is a database server, which stores data related to the published application in one or more local service databases. The persistent mass storage 225 may be a component internally located in or externally located from any or all of the servers 110, 115, 205.

[0038] Referring to FIG. 3, the web publishing tool 223 receives a web service description and access information for an application (e.g., GUI application) for publishing (step 300). In one embodiment, the web service description includes the service information described above (e.g., the name of the business offering the web service, the service type, a textual description of the service, and a SAP). The access information may include, for example, a published application name, a Transmission Control Protocol (TCP) browsing server farm address, and a MetaFrame server IP address. In some embodiments, the access information specifies the address to use and a ticket to use to traverse network or security gateways or bridge devices.

[0039] The web publishing tool 223 then constructs a service-publishing request to request the publication of the web service (i.e., GUI application) (step 305). In one embodiment, the service-publishing request includes a SAP. In some embodiments, the SAP is a URL including the web address of the web server 115 and the publishing server plug-in 220. Further, the web address can be a Uniform Resource Identifier (URI), which is the generic term for the types of names and addresses that refer to objects on the web. A URL is one kind of URI. An example of the URI is the name of the web server 115 (e.g., "web-server") and the CGI script name (e.g., "dynamic-component") for the publishing server plug-in 220.

[0040] The web publishing tool 223 stores a SAP entry associated with the SAP in the persistent mass storage 225 (step 310). In some embodiments, the web publishing tool 223 also associates published application information (e.g., ICA-published-app-info) with the GUI application. In further embodiments, the web publishing tool 223 also includes a key in the service-publishing request to identify the SAP entry that the content server 110 stores in

- 10 -

the persistent mass storage 225. For instance, the key can have the value of "123456677." An example of a SAP identifying the web server 115, the CGI script name of the publishing server plug-in 220, and the key described above is "http://web-server/dynamic-component/?app=123456677."

5 **[0041]** An example of the SAP entry associated with the SAP described above is "key=123456677, value=ICA-published-app-info." The key can be any length (e.g., 56 bit key, 128 bit key). In one embodiment, the key is a cryptographic random number. The key may also provides an access right to the key holder. Although illustrated with a key, any means can be used to provide a form of security to the SAP entry stored in the persistent
10 mass storage 225.

[0042] The web publishing tool 223 then provides the service-publishing request to the content server 110 for publishing in the web service directory 145 (steps 315 and 320). Moreover, in one embodiment, the content server 110 transmits the key of the SAP to the client 105 requesting the particular web service for subsequent use in locating the SAP entry.
15 In one embodiment, the publishing of the service-publishing request enables users of the client 105 to access the service. In one embodiment, GUI applications are published on the web service directory 145 using NFUSE developed by Citrix Systems, Inc. of Fort Lauderdale, Florida. In some embodiments, a publisher of a GUI application customizes the publication of the GUI application on the web service directory 145 using Application
20 Launching And Embedding (ALE), also developed by Citrix Systems, Inc. ALE enables the launching of a GUI application from or the embedding of the application into an HTML page.

[0043] The client 105 then queries a service name from the web service directory 145 (step 322). The content server 110 receives the query from the client 105 (step 325) and
25 finds the requested service name in the web service directory 145. In another embodiment, the user of the client 105 navigates the web service directory 145 until locating a particular service name that the user of the client 105 was attempting to find. Although illustrated with the client 105, any web service directory client (e.g., UDDI client or LDAP browser) can query or navigate the web service directory 145 to discover published web services.

30 **[0044]** Upon location of the SAP associated with the received query, the content server 110 transmits the SAP to the client 105 (step 330). The client 105 receives the SAP (step 335) and determines the address of the publishing server plug-in 220 from the SAP. The

- 11 -

client 105 subsequently transmits a request for the GUI application to the web server 115 (step 340). In some embodiments, the request from the client 105 is an HTTP request transmitted from the web browser 140 to the web server 115. In other embodiments, an application (e.g., general directory browser or HTML UI) executing on the client 105 receives the SAP from the content server 110 and provides the SAP as an argument to the web browser 140. The web browser 140 may then automatically transmit an HTTP request (for the GUI application) to the web server 115. Following along the lines of the previous examples, a particular example of the application request to the web server 115 is “http://web-server/dynamic-component/?app=123456677).”

[0045] The web server 115, and, more particularly, the publishing server plug-in 220, receives the application request associated the SAP (step 345) and determines the SAP entry associated with the request (step 350). In one embodiment, the publishing server plug-in 220 receives the request from the client 105 and retrieves the published application information associated with the request that had been stored (as part of the SAP entry) in the persistent mass storage 225. In some embodiments, the publishing server plug-in 220 uses the SAP (or part of the SAP) that the client 105 received from the content server 110 as the key to access the proper service entry (e.g., the published application information) stored in the persistent mass storage 225.

[0046] The publishing server plug-in 220 then constructs a file or document having the published application information (e.g., HTTP address of the application server 205) (step 352) and transmits this document to the client 105 (step 355). The publishing server plug-in 220 constructs the file so that the file has a format compatible with the application client 215. In one embodiment, the document is a Multipurpose Internet Mail Extensions (MIME) or a secure MIME (S/MIME) document. In another embodiment, the document is an HTML document containing an ICA web client embedded object HTML tag.

[0047] The web browser 140 subsequently receives the document and attempts to open the document. In one embodiment, if the application client 215 is not installed on the client 105, the client 105 communicates with the application server 205 to download and install the application client 215. Upon installation of the application client 215 or, alternatively, if the application client 215 has already been installed on the client 105, the client 105 launches the application client 215 to view the document received from the web server 115 (step 360).

- 12 -

[0048] Once the application client 215 is installed and executing on the client 105, the application server 120 then executes the application and displays the application on the application client 215 (step 365). In another embodiment, the client 105 views the document (even before launching the application client 215) and uses the information in the document to obtain the GUI application from the application server 205. In this embodiment, the display of the GUI application includes the installation and execution of the application client 215. Moreover, the viewing of the document may be transparent to the user of the client 105. For example, the client 105 may receive the document from the web server 115 and interpret the document before automatically requesting the GUI application from the application server 205.

[0049] Authentication to the application server 205 can be done in any number of ways. For example, the user can be required to provide a password that the application server 205 verifies is correct. In other embodiments, a “ticket authority” generates one or more tickets for use by the client 105 to authenticate the client 105 to the application server 205.

[0050] In one embodiment, an email is sent to potential service users directing the user to view the web service directory 145. For instance, if an organization updates the web service directory 145 and consequently wants to direct users to the company’s update, the organization may send an email to potential service users to advertise the update.

[0051] Thus, the application client 215 provides service-based access to published applications, desktops, desktop documents, and any other application that is supported by the application client 215. Examples of applications that the application client 215 can provide access to include, but are not limited to, the WINDOWS desktops, WINDOWS documents such as MICROSOFT EXCEL, WORD, and POWERPOINT, all of which were developed by Microsoft Corporation of Redmond, Washington, Unix desktops such as SUN SOLARIS developed by Sun Microsystems of Palo Alto, California, and GNU/Linux developed by Red Hat, Inc. of Durham, North Carolina.

[0052] An organization or firm using a corporate portal, such as XPS developed by Citrix Systems, Inc. of Fort Lauderdale, Florida, to provide users (e.g., employees, suppliers, partners, and customers) with a collection of information associated with the organization or firm may benefit from the publishing of GUI applications. In more detail and in some embodiments, the computer system 200 supports role-based access control for additional security. A role describes a particular group of users who can access certain content and

- 13 -

applications within the portal. As users log onto the portal, the presentation they see is generated depending upon their roles.

[0053] By defining roles, a portal administrator can control and present information that is most relevant to the group's job function. Thus, the computer system 200 can provide users having a certain role access to particular GUI applications (and/or application information). Moreover, the computer system 200 may support subscription-based access to GUI applications (and/or application information) where, for example, a customer pays a subscription fee each month to access published GUI applications.

[0054] For example, an employee of an organization creates a report of results of a test conducted over a six month period in a spreadsheet program, such as "Spreadsheet X" (e.g., MICROSOFT EXCEL), using a desktop computer. The desktop has the most current version of Spreadsheet X available, which was the version used to create the report. The employee then departs on a business trip. Before leaving, the employee knows that, upon return from the business trip, the employee has to present the report to the organization's customer (e.g., another organization) at the customer's site. Thus, the employee saves the report onto a disk before his departure. Upon return, the employee travels to the customer's site. The employee then realizes that the customer's machines only support a previous version of Spreadsheet X. Because the employee used some of the new features of the latest version of Spreadsheet X in the report, the employee would prefer not to use an earlier version of the software.

[0055] The developer of Spreadsheet X recently released the latest version of Spreadsheet X. To attract customers and increase the user-base of Spreadsheet X, the developer publishes Spreadsheet X on the web service directory 145. For instance, the developer's application server 205 transmits Spreadsheet X and web service publishing information about Spreadsheet X (e.g., developer's name, web address of developer's application server 205) to the content server 110. The content server 110 uses this information to create a service publishing request for the Spreadsheet X, such as "http://web-server/dynamic-component/SpreadsheetX=987654321," stores an associated SAP entry in a local database (i.e., persistent mass storage 225), and publishes the service publishing request in the web service directory 145.

[0056] Instead of traveling back to his office and wasting valuable time, the employee can use the customer's computer (i.e., the client 105) to display the latest version of Spreadsheet

- 14 -

X on the customer's computer for his presentation. First, the employee connects the customer's computer to the web. Once connected, the employee queries the content server 110 for the current version of Spreadsheet X. The customer's computer then receives the associated SAP, which is the address of the publishing server plug-in 220, from the content server 110. The customer's computer then communicates with the publishing server plug-in 220 (using the SAP received from the content server 110) to obtain a document containing information to retrieve the published application. To open the document, the customer's computer launches the ICA client 215 and subsequently interprets the document. Using the information in the document, the customer's computer establishes a communication session with the application server 205 to view the latest version of Spreadsheet X using the ICA client 215. Thus, the employee can present the test data using the most current version of Spreadsheet X. Alternatively, the employee can only access the Spreadsheet X if the organization subscribes to the published application service.

[0057] Having described certain embodiments of the invention, it will now become apparent to one of skill in the art that other embodiments incorporating the concepts of the invention may be used. Therefore, the invention should not be limited to certain embodiments, but rather should be limited only by the spirit and scope of the following claims.

- 15 -

CLAIMS

What is claimed is:

1 1. A method of remotng an application to an application client or end user application
2 comprising:

3 (a) receiving from a web service directory a service access point associated with a first
4 application, the service access point identifying a web server;

5 (b) retrieving address information associated with the first application from the web
6 server identified by the service access point;

7 (c) launching a second application, the second application establishing a
8 communication channel with an application server identified by the retrieved address
9 information.

1 2. The method of claim 1 further comprising transmitting a query to the web service
2 directory to search for the first application prior to step (a).

1 3. The method of claim 1 wherein step (a) comprises receiving the address of a web server
2 and receiving a key for locating information associated with the first application in a local
3 database.

1 4. The method of claim 1 wherein step (b) comprises retrieving a MIME document including
2 the address information associated with a first application.

1 5. The method of claim 1 wherein the second application comprises an application client.

1 6. The method of claim 5 further comprising installing the application client prior to step (c).

1 7. The method of claim 1 wherein the step (c) further comprises establishing the
2 communication channel with the application server and transmitting the address information
3 to the application server.

1 8. The method of claim 1 wherein the application server launches the first application.

1 9. The method of claim 8 wherein the first application comprises a GUI application.

1 10. A client for use with executing a remote first application provided by a service provider
2 on a web service directory, the client comprising:

3 a client browser, the client browser (a) receiving from the web service directory a
4 service access point associated with the first application, the service access point identifying
5 a web server and (b) retrieving address information associated with the first application from
6 the web server identified by the service access point; and

7 an application client, the application client (a) being launched in response to the
8 retrieved address information and (b) establishing a communication channel with an

- 16 -

9 application server identified by the retrieved address information to receive data from the
10 first application.

1 11. The client of claim 10 wherein the retrieved address information is included in a MIME
2 document.

1 12. The client of claim 11 wherein the MIME document is an ICA file type.

1 13. A method of remoting an application to an application client or end user application
2 comprising:

3 a) sending, by a client, a query to a web service directory for information associated
4 with a first application;

5 b) transmitting to the client from the web service directory a service access point
6 associated with the first application, the service access point identifying a web server;

7 c) retrieving, by the client, address information associated with the first application
8 from the web server identified by the service access point;

9 d) launching by the client a second application, the second application establishing a
10 communication channel with an application server identified by the retrieved address
11 information; and

12 e) launching by the application server the first application and returning information to
13 the second application.

1 14. The method of claim 13 wherein step (b) further comprises receiving an address of the
2 web server and receiving a key for locating information associated with the first application
3 in a local database.

1 15. The method of claim 13 wherein step (c) further comprises accessing a local database
2 using a key contained within the service access point and having the local database return
3 address information associated with the first application.

1 16. The method of claim 13 wherein step (c) comprises retrieving a MIME document
2 including the address information associated with a first application.

1 17. The method of claim 13 wherein the second application is an application client.

1 18. The method of claim 17 further comprising installing the application client prior to step
2 (d).

1 19. The method of claim 13 wherein the first application is a GUI application.

1 20. A method of remoting an application to an application client or end user application
2 comprising:

3 a) receiving from a client a request to access a service access point associated with a
4 first application;

- 17 -

5 b) identifying that the requested service access point can be used with an application
6 client;

7 c) retrieving address information associated with the first application identified by the
8 service access point;

9 d) converting the retrieved address information into a file, the format of the file
10 associated with the application client; and

11 e) transmitting the file including the converted address information to the client.

1 21. The method of claim 20 wherein the service access point is identified in a web service
2 directory.

1 22. The method of claim 20 further comprising transmitting the application client to the
2 client.

1 23. The method of claim 20 further comprising sending, by the client, a query to the web
2 service directory for information associated with a first application.

1 24. The method of claim 20 further comprising transmitting to the client from the web
2 service directory the service access point associated with the first application, the service
3 access point identifying a web server;

1 25. The method of claim 20 further comprising launching by the client an application client,
2 the application client establishing a communication channel with an application server
3 identified by the retrieved address information.

1 26. The method of claim 20 further comprising receiving an address of the web server and
2 receiving a key for locating information associated with the first application in a local
3 database.

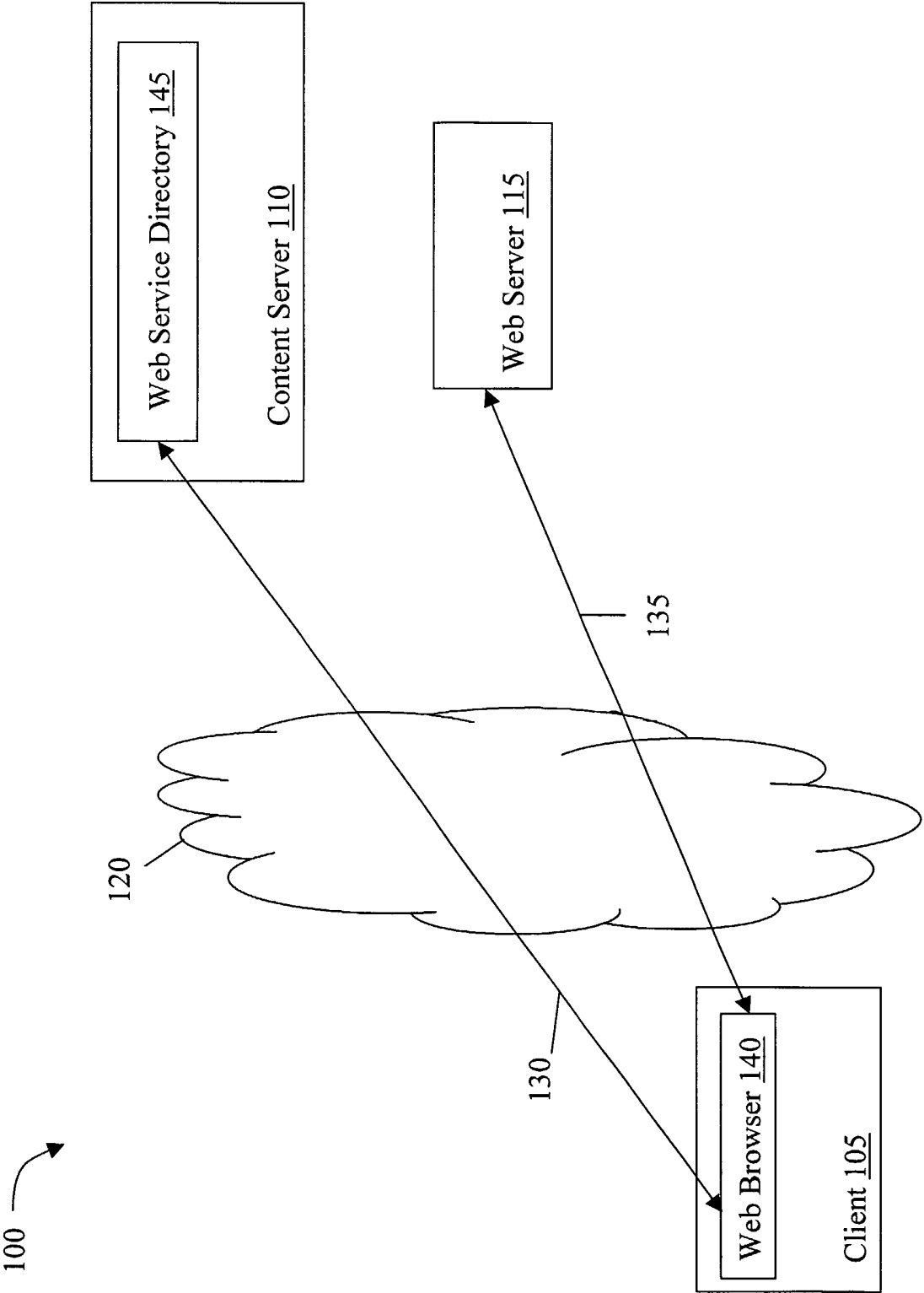
1 27. The method of claim 20 further comprising accessing a local database using a key
2 contained within the service access point and having the local database return address
3 information associated with the first application.

1 28. The method of claim 20 wherein step (c) comprises retrieving a MIME document
2 including the address information associated with a first application.

1 29. The method of claim 20 further comprising installing the application client on the client.

1 30. The method of claim 20 wherein the first application is a GUI application.

1 31. The method of claim 20 wherein the second application is a GUI application.



PRIOR ART

FIG. 1

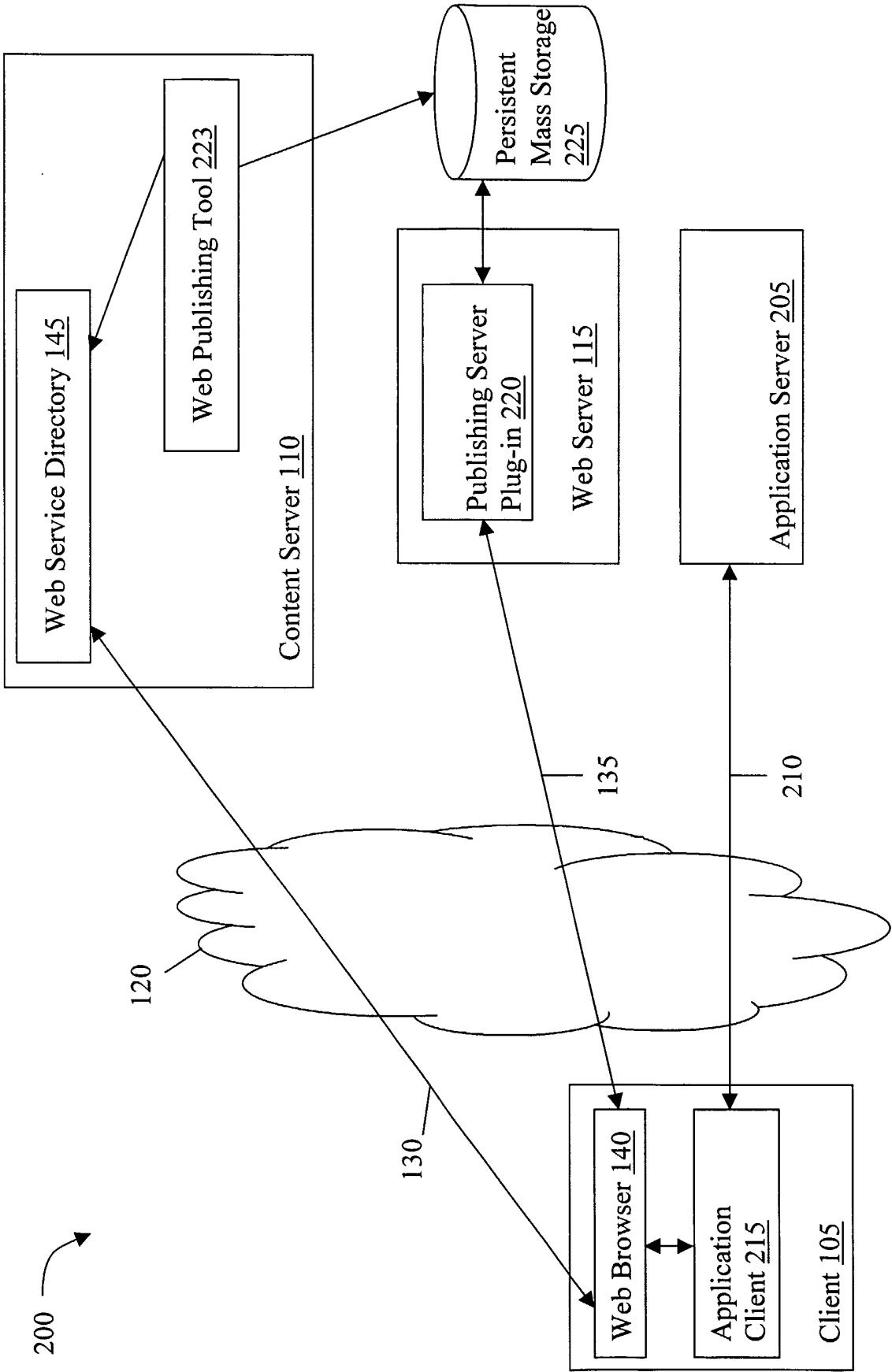


FIG. 2

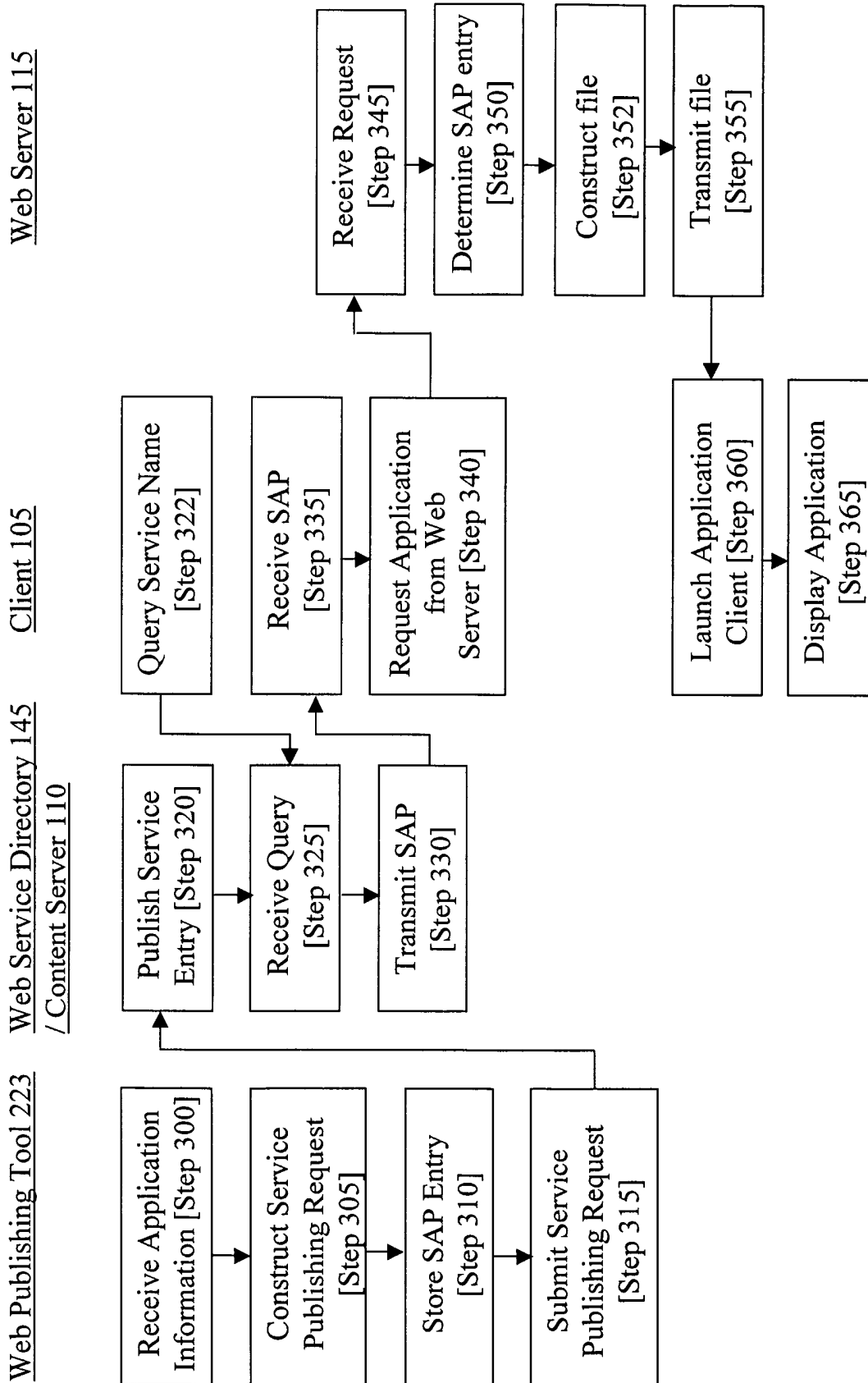


FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US03/08434

A. CLASSIFICATION OF SUBJECT MATTER IPC(7) : G06F 13/00 US CL : 709/328 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) U.S. : 709/200,203,310,328,330 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched NONE Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) WEST: search terms: (web near2 (servic\$ near1 direcot\$)) and (gui or guis)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6,067,568 A (LI et al.) 23 MAY 2000 (23.05.2000), see Abstract, figures 1-14b, and col. 2 (line 37-et seq.).	1-31
A	US 5,960,409 A (WEXLER) 28 September 1999 (28.09.1999), see Abstract, figures 1-3, and col. 2 (line 38-et seq.).	1-31
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 13 July 2003 (13.07.2003)		Date of mailing of the international search report 14 AUG 2003
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450 Alexandria, Virginia 22313-1450 Facsimile No. (703) 746-7239		Authorized officer For Robert B. Harrell Telephone No. (703) 305-9692 <i>James R. Matthews</i>