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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 December 2010 (16.12.2010)

PCT

(10) International Publication Number
WO 2010/142786 A1

(51) International Patent Classification:
G06F 9/455 (2006.01) *G06F 9/46* (2006.01)

(21) International Application Number:
PCT/EP2010/058230

(22) International Filing Date:
11 June 2010 (11.06.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/484,085 12 June 2009 (12.06.2009) US

(71) Applicant *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).

(71) Applicant *(for MG only)*: **IBM UNITED KINGDOM LIMITED** [GB/GB]; P.O. Box 41, North Harbour, Portsmouth, Hampshire PO6 3AU (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BOCKUS, Michael Andrew** [US/US]; IBM Corporation, MD 905-6C-015, 11501 Burnet Road, Austin, Texas 78758-3400 (US). **HENDERSON, Jacob** [US/US]; IBM Corporation, IP Law, 11400, Burnet Road, Austin, Texas 78758 (US). **SAWICKI, Kevin Robert** [US/US]; IBM Corporation, IP Law, 11400, Burnet Road, Austin, Texas 78758 (US). **KEANE, John** [US/US]; IBM Corporation, IP Law, 11400, Burnet Road, Austin, Texas 78758 (US).

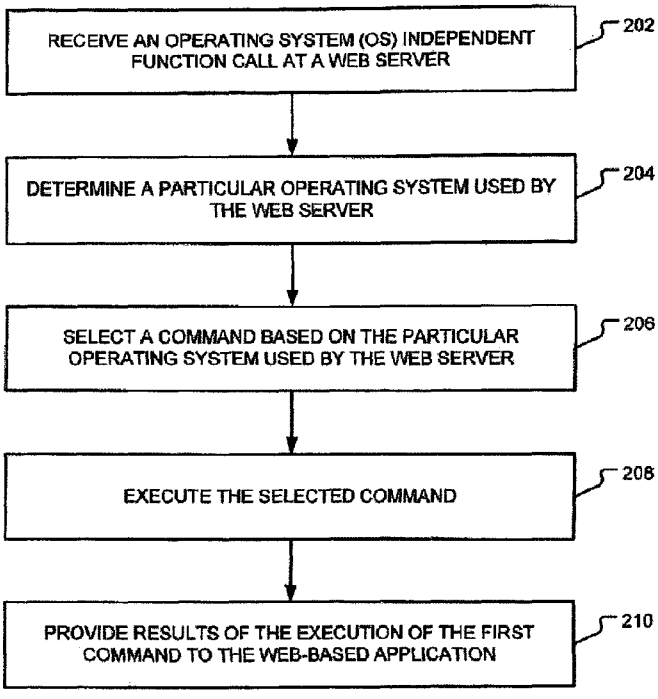
(74) Agent: **ROBERTS, Scott**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester, Hampshire SO21 2JN (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,

[Continued on next page]

(54) Title: SYSTEMS AND METHODS FOR OPERATING SYSTEM IDENTIFICATION AND WEB APPLICATION EXECUTION

FIGURE 2



(57) Abstract: Systems and methods to execute operating system dependencies for web applications are provided. A particular method includes receiving an operating system independent function call at a web server. The operating system independent function call may be initiated at a web-based application. The method may determine a particular operating system used by the web server and select a command based on the particular operating system. The method may further include executing the selected command

WO 2010/142786 A1

NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

SYSTEMS AND METHODS FOR OPERATING SYSTEM IDENTIFICATION AND WEB APPLICATION EXECUTION

Field

The present disclosure is generally related to networked computing systems, and more particularly, to web application execution.

Description of Related Art

Web applications may rely upon functions that use instructions that, in turn, depend upon a particular operating system of a computing system on which the web application is executing. Certain web applications, as a result, include code to execute a series of tests in order to determine the type of operating system on which the web application may execute. The web application may be modified if there is a need or desire to execute the web application on a different computing system having a different operating system. Modifying web applications for different operating system environments adds costs to development.

Brief Summary

Systems and methods to execute web applications are provided. A particular method includes receiving an operating system independent function call at a web server from a web-based application. The method further includes determining a particular operating system used by the web server and selecting a command based on the particular operating system. The method further includes executing the selected command associated with the operating system independent function call at the web server.

A system is described that includes a processor and a memory accessible to the processor. The memory includes instructions executable at the processor, where the instructions include an operating system independent function call. The memory further includes instructions, executable at the processor, to determine an operating system type and to select an operating system dependent command based on the determined operating system type, and to execute

the selected operating system dependent command in response to the operating system independent function call.

A computer readable medium includes instructions that, when executed by a processor in response to an operating system independent function call from a web-based application, cause the processor to determine a particular operating system used by the processor. The computer readable medium may further include instructions that, when executed by the processor, cause the processor to select a command from a plurality of commands associated with the operating system independent function call. Each of the plurality of commands is associated with a different operating system, and the selected command is associated with the particular operating system.

Brief Description of the Drawings

Preferred embodiments of the present invention will now be described, by way of example only, with reference to the following drawings:

FIG. 1 is a block diagram of an illustrative computer system configured to execute a web application in accordance with a preferred embodiment of the present invention;

FIG. 2 is a flowchart showing processes executable by the system of FIG. 1 to execute a web application in accordance with a preferred embodiment of the present invention; and

FIG. 3 is a block diagram of another illustrative computer system configured to execute a web application in accordance with a preferred embodiment of the present invention.

Detailed Description

FIG. 1 illustrates a computing apparatus, such as a client-server based computer system 100 configured to execute web applications. In one aspect, the computer system 100 may provide web developers with an interface to retrieve a path to an executable web application without necessitating extra code to determine the operating system on which the web

application is executing. Instead, a web server 146 may execute an identify command to transparently determine an operating system that is being used. The web server 146 may further provide the correct path to any required executables based on a set of parameters provided by the developers.

The system 100 of FIG. 1 is configured to determine a path to an executable by determining the type of operating system being executed by the system 100. The system 100 includes at least one computer 112. The computer 112 shown in FIG. 1 may represent any type of computer, computer system or other programmable electronic device capable of functioning as a client or a server in a client-server environment. Moreover, the computer 112 may be implemented using one or more networked computers, e.g., in a cluster or other distributed computing system.

The computer 112 includes a central processing unit (CPU) 126 that includes at least one microprocessor coupled to a memory 118. The memory 118 may include an operating system 140. The computer 112 generally operates under the control of the operating system 140. The operating system 140 may generally function as an interface between hardware and applications. The operating system 140 may be responsible for the management and coordination of activities and the sharing of computer resources.

The operating system 140 additionally may act as a host for applications that execute on the computer 112. As a host, the operating system 140 may manage details of the application as related to the operation of the hardware. Applications, such as web applications 148, may access operating system services through application programming interfaces (APIs) 150 or system calls. By invoking an API 150, the application may request a service from the operating system 140, pass parameters, and receive any results of an operation. Users may also interact with the operating system 140 using a software user interface, such as a command line interface or a graphical user interface.

The memory 118 may also include various computer software applications, components, programs, objects, modules, data structures, etc, that are executable by the CPU 126. For example, a web server 146 may include a program that accepts HTTP requests from clients,

e.g., a web browser, and serves responses that typically include web pages, such as HTML documents and linked objects. The web server 146 may identify the operating system 140 being executed by the computer 112. The web server 146 may identify the operating system 140 in response to an operating system independent function call initiated at a web application 148. For example, a web application 148 may initiate an operating system independent call querying sales data. In response to the sales data query, an exemplary web server 146 may execute an identify command, e.g., a Java command including "System.getProperty ("os.name")". The indentify command may be configured to retrieve information identifying the operating system used by the web server 146.

In an embodiment, the web server 146 may include an executable module 156. The executable module 156 may include executables that execute an identify command to determine the operating system type. For example, the executable module 156 may receive the operating system independent function call prompting a display of employee address information. The executable module 156 may execute an identify command to determine the operating system type. The executable module 156 may further execute an operating system dependent command that is based on the operating system type and that initiates the display of employee address information.

The memory 118 may represent random access memory (RAM) devices comprising main storage of the computer 112, as well as any supplemental levels of memory, e.g., cache memories, non-volatile or backup memories (e.g., programmable or flash memories), read-only memories, etc. In addition, the memory 118 may include memory storage physically located elsewhere in the computer 112, e.g., cache memory in CPU 126, as well as any storage capacity used as a virtual memory, e.g., as stored on a mass storage device 130 or on another computer coupled to the computer 112 via a network 136. The computer 112 may receive and output data via a user interface 132.

The embodiment of FIG. 1 may reduce the number of tests used to determine the operating system. The web application may be modified to enable execution by the determined operating system at any time without rebuilding the source code. The embodiment of FIG. 1 may further provide web developers with a direct path to an executable web application that

includes codes appropriate for the operating system being executed by the web server. The appropriate code, e.g., command string, may be determined and retrieved without necessitating extra software at the web application to determine the operating system on which the web application is executing. Instead, the web server 146 may transparently determine the operating system that is being used in response to an operating system independent function call. The web server 146 may provide the correct path to a version of code appropriate for the executing operating system. For example, an appropriate version of installation code may be selected from among a number of versions of installation programs, each version associated with a different operating system.

Referring to FIG. 2, a method is described that includes receiving an operating system independent function call at a web server from a web-based application, at 202. By way of example, a web-based application may be designed to list the user accounts on the server. According to one embodiment, the web-based application may be programmed to provide an operating system independent function call. For example, in a Java Websphere environment, the web-based application could utilize the provided method signature:

public.java.lang.String getExecutable (java.lang.String command ID)

In another example, an application web.xml file may be provided as follows:

<executable commandId="LISTUSER">
<command ostype="Linux">cat /etc/passwd | cut -d: -f1</command>
<command ostype="AIX">lsuser -c -a id home ALL | sed '/^#.*/d' | tr ':'\011'</command>
</executable>

Responsive to receiving the operating system independent function call, a determination may be made as to the particular operating system used by the web server, at 204. The method used by the web server to determine the operating system being used may vary by system. As an example, a Java 2 Platform, Enterprise Edition (J2EE) server may use a Java System property “os.name”. The J2EE server may use an identify command that include a Java method, such as System.getProperty (“os.name”), to retrieve the operating system identification information.

Thus, in the exemplary system using a Java Websphere application executing on a J2EE platform, the web based application may be programmed to send the command "LISTUSER" to a web server. Responsive to receiving the command "LISTUSER", a determination may be made at the web server of the particular operating system being used, e.g., by executing the System.getProperty ("os.name") command. A method signature "public java.lang.String getExecutable(java.lang.String commandId)" may take the command (e.g., LISTUSER) attribute of the executable element as a parameter.

A command may be selected in response to the operating system independent function call and based on the particular operating system used by the web server 206. For example, the independent function call from the web application may initiate a determination by the web server of the operating system being executed by the web server. The command may include a version of installation code for the web application that is appropriate for the determined operating system. The web server 206 may provide a command string for AIX or Linux, for instance. Where the operating system is unsupported, a null value may alternatively be returned. In this manner, a command may be selected based on the operating system independent function and based on the particular operating system being used.

The command may be executed to access and implement the function associated with the operating system independent function call, at 208. For example, the command string for AIX or Linux may be executed by a respective AIX or Linux operating system to execute a function to, for example, list user accounts on the system.

In a particular embodiment, results of execution of the command may be provided to the web based application 210. In one particular example, only two operating systems (e.g., AIX and Linux) may be supported. Additional operating systems may be supported in other embodiments. For example, the following additional Java-supported operating systems may be used with certain embodiments: Digital Unix; FreeBSD; HP UX; Irix; Mac OS; MPE/iX; Netware 4.11; OS/2; Solaris; Windows 2000; Windows 95; Windows 98; Windows NT; and Windows XP.

The executable module 156 may be separate from and may operate independently with respect to the web-based application, allowing support for additional operating systems (whether currently supported by Java or supported in the future) to be added without the need to alter or rebuild the software for the web-based application. Rather, additional <command> elements may be added to the web.xml configuration file with corresponding operating system identifiers and commands.

The commands may access a function of the particular operating system. Continuing with the earlier example, one function may be to list the user accounts on the server. However, any number of functions may be accessed using the described techniques. In certain cases, a command may provide for including data necessary to execute the function. Data may be provided as part of the operating system independent function call and passed to the web server, such as the example of using the web.xml configuration file.

The embodiment of FIG. 2 may reduce the number of tests used to determine the operating system. The web server 146 may be used to determine the operating system being executed by the web server 146. The web application may be modified to enable execution by the determined operating system at any time without rebuilding the source code of the web application. The appropriate code, e.g., command string, may be determined and retrieved without necessitating extra web application software to determine the operating system on which the web application is executing. Instead, the web server 146 may transparently determine the operating system that is being used in response to an operating system independent function call. The web server 146 may provide the correct path to a version of code appropriate for executing the requested function at the executing operating system.

FIG. 3 is a block diagram of a computing system configured to execute the method of FIG. 2 to execute a web application. Computing system 300 includes webapp execution program 302, a web server 304, and an executable module 306 for executing a web application in a networked environment.

In the depicted example, the computing system 300 includes a processor 308, a main memory 310, an input device 312, such as a keyboard or mouse, a display device 314, and a network adapter 316. The network adapter 316 is coupled to client computers 318, 320.

The main memory 310 may include computer instructions 322 installed onto a computer readable medium, such as computer memory storage. The main memory 310 may include the computer instructions 322 executable by the processor 308. The computer instructions 322 may include an operating system independent function call. As an example, the operating system independent function call may be the LISTUSER function described above, but may also be any number of other functions as may be useful for implementation with a particular web-based application. The computer instructions 322 may determine the operating system type of the server and may execute an operating system dependent command selected from a plurality of commands 324.

The operating system dependent command may be selected based on the operating system type and in response to the operating system independent function call of the instructions 322. For example, the operating system dependent command may include a version of installation code selected from a number of installation codes where each code is associated with a corresponding operating system. The selection may be automatically made based on the determination of the operating system being executed by the web server. The operating system determination may be made in response to receiving the operating system independent call.

The main memory 310 may also include the executable module 306, such as the executable module 156 described in reference to FIG. 1, to execute a command 324 to determine the operating system type. The executable module 306 may additionally execute an operating system dependent command. In one example, either a first operating system dependent command (e.g., for AIX) or a second operating system dependent command (e.g., for Linux) may be executed depending on the operating system type. Additional operating system dependent commands may be added and made accessible to the main memory 310 for other operating systems.

An operating system (not shown) runs on the processor 308 and coordinates and provides control of various components within the computing system 300. The operating system may be a commercially available operating system such as Microsoft® Windows® XP (Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both). An object oriented programming system, such as the Java® programming system, may run in conjunction with the operating system and provide calls to the operating system from Java programs or applications executing on the computing system 300 (Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both).

The hardware in computing system 300 may vary depending on the implementation. Other internal hardware or peripheral devices, such as flash memory, equivalent non-volatile memory, or optical disk drives and the like, may be used by the computing system 300 but are not specifically illustrated in FIG. 3 to simplify the explanation. Also, the processes of the disclosed illustrative embodiments may be applied to a multiprocessor data processing system.

The depicted examples described with respect to FIG. 3 are not meant to imply architectural limitations. For example, portions of the computing system 300 may be implemented in a personal computer, a server, a server cluster, a tablet computer, a laptop computer, or a communication device.

Particular embodiments of the computing system 300 can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements. In a particular embodiment, the disclosed methods are implemented in software that is embedded in a processor readable medium and executed by a processor (including but not limited to firmware, resident software, microcode, etc).

Further, embodiments of the present disclosure, such as the one or more embodiments in FIGS. 1-3, can take the form of a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a

computer-usable or computer-readable medium can be any apparatus that can tangibly embody a computer program and that can contain, store, or communicate the program, or portions thereof, for use by or in connection with the computer, the instruction execution system, an apparatus, or a computing device.

In various embodiments, the medium can include an electronic, magnetic, optical, electromagnetic, infrared, or a semiconductor system (or apparatus or device). Examples of a computer-readable medium include a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk. Current examples of optical disks include compact disk – read only memory (CD-ROM), compact disk-read/write memory (CD-R/W) and digital versatile disk (DVD).

The previous description of the disclosed embodiments is provided to enable a person skilled in the art to make or use the disclosed embodiments. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles disclosed herein may be applied to other embodiments without departing from the scope of the disclosure. Thus, the present disclosure is not intended to be limited to the embodiments shown herein, but is to be accorded the widest scope possible consistent with the principles and features as defined by the following claims.

CLAIMS

1. A method comprising:

receiving an operating system independent function call at a web server, the call initiated at a web-based application;

determining a particular operating system used by the web server;

selecting a command based on the particular operating system used by the web server; and

executing the selected command.

2. The method of claim 1, wherein the selected command is executed at the web server.

3. The method of claim 1, wherein the selected command initiates a web function associated with the operating system independent function call.

4. The method of claim 1, wherein the selected command initiates a function of the operating system used by the web server.

5. The method of claim 1, wherein the selected command is further selected based on the operating system independent function call.

6. The method of claim 1, wherein determining the particular operating system further comprises initiating an identification command at the web server to determine the particular operating system used by the web server.

7. The method of claim 1, wherein the operating system independent function call includes data to execute the function.

8. The method of claim 1, wherein the selected command includes data to execute a function of the particular operating system, wherein the function is selected based on the operating system independent function call.

9. The method of claim 1, wherein selecting the command further comprises selecting a version of an installation program for the web application based on the particular operating system.

10. The method of claim 1, further comprising providing results of execution of the command at the web server to the web-based application.

11. The method of claim 1, further comprising executing an executable module separate from the web-based application and in communication with the web server, wherein the executable module determines the particular operating system and selects the command based on the particular operating system.

12. The method of claim 1, wherein the web server identifies the particular operating system by executing an identify command.

13. The method of claim 12, wherein the web server is a Java server and the identify command is a Java system command.

14. A system comprising:

a processor; and

a memory accessible to the processor, the memory including:

instructions executable at the processor, the instructions including an operating system independent function call; and

instructions executable at the processor to determine an operating system type and to select an operating system dependent command based on the determined operating system type, and to execute the selected operating system dependent command in response to the operating system independent function call.

15. The system of claim 14, wherein the operating system dependent command is further selected based on the operating system independent function call.

16. The system of claim 14, wherein the memory further comprises an executable module to receive the operating system independent function call from a web application.

17. The system of claim 14, wherein the selected command is executed at a web server.

18. The system of claim 14, wherein the command comprises an installation program for a web application based on the determined operating system.

19. A computer readable medium comprising:

instructions that, when executed by a processor in response to an operating system independent function call from a web-based application, cause the processor to determine a particular operating system used by the processor; and

instructions that, when executed by the processor, cause the processor to select a command from a plurality of commands associated with the operating system independent function call, wherein each of the plurality of commands is associated with a different operating system, and wherein the command is associated with the particular operating system.

20. The computer readable medium of claim 19, further comprising instructions that, when executed by the processor, cause the processor to send results of execution of the command to the web-based application.

FIGURE 1

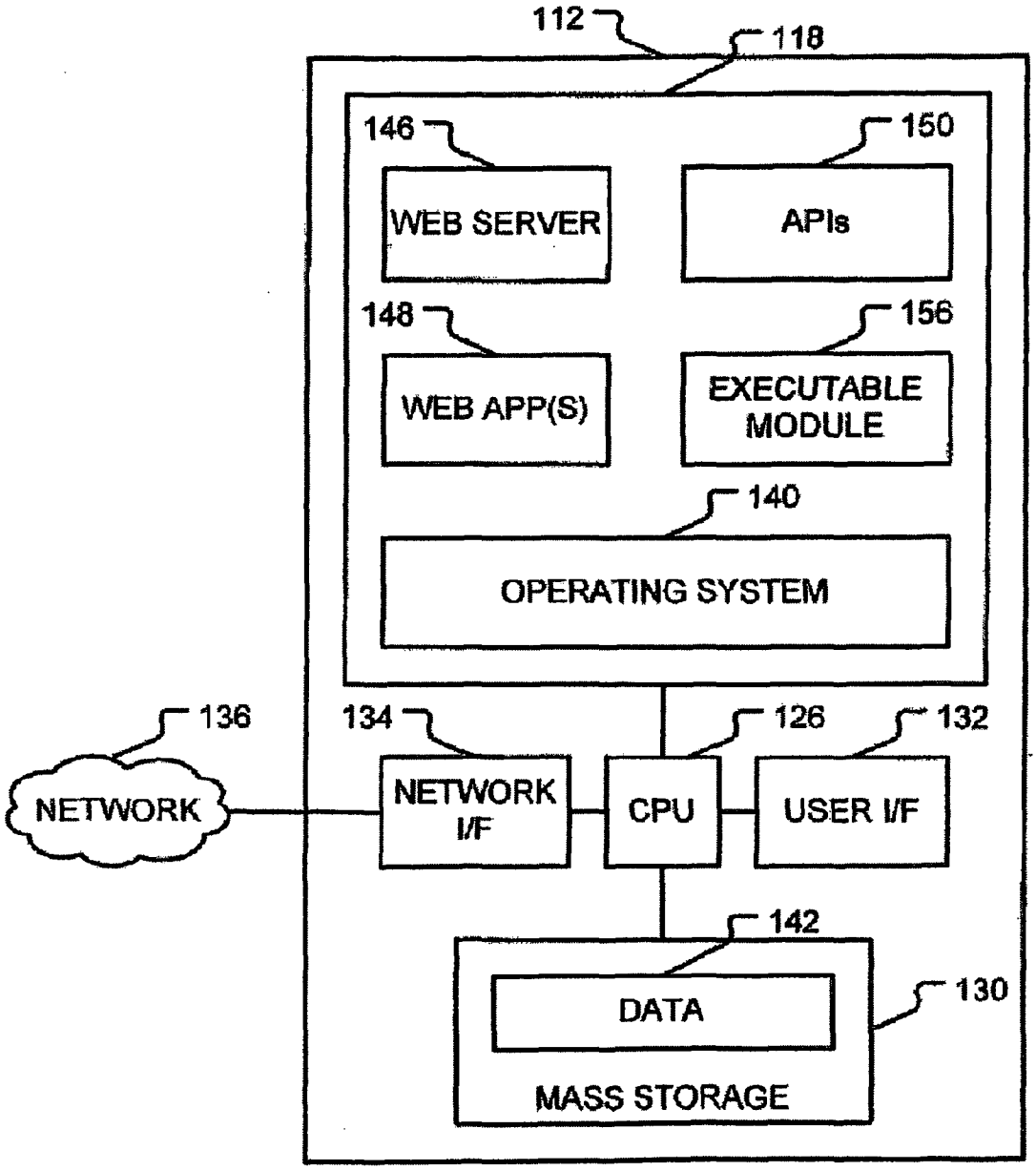

112
118
146
WEB SERVER
150
APIs
148
WEB APP(S)
156
EXECUTABLE MODULE
140
OPERATING SYSTEM
136
NETWORK
134
NETWORK I/F
126
CPU
132
USER I/F
142
DATA
130
MASS STORAGE

FIGURE 2

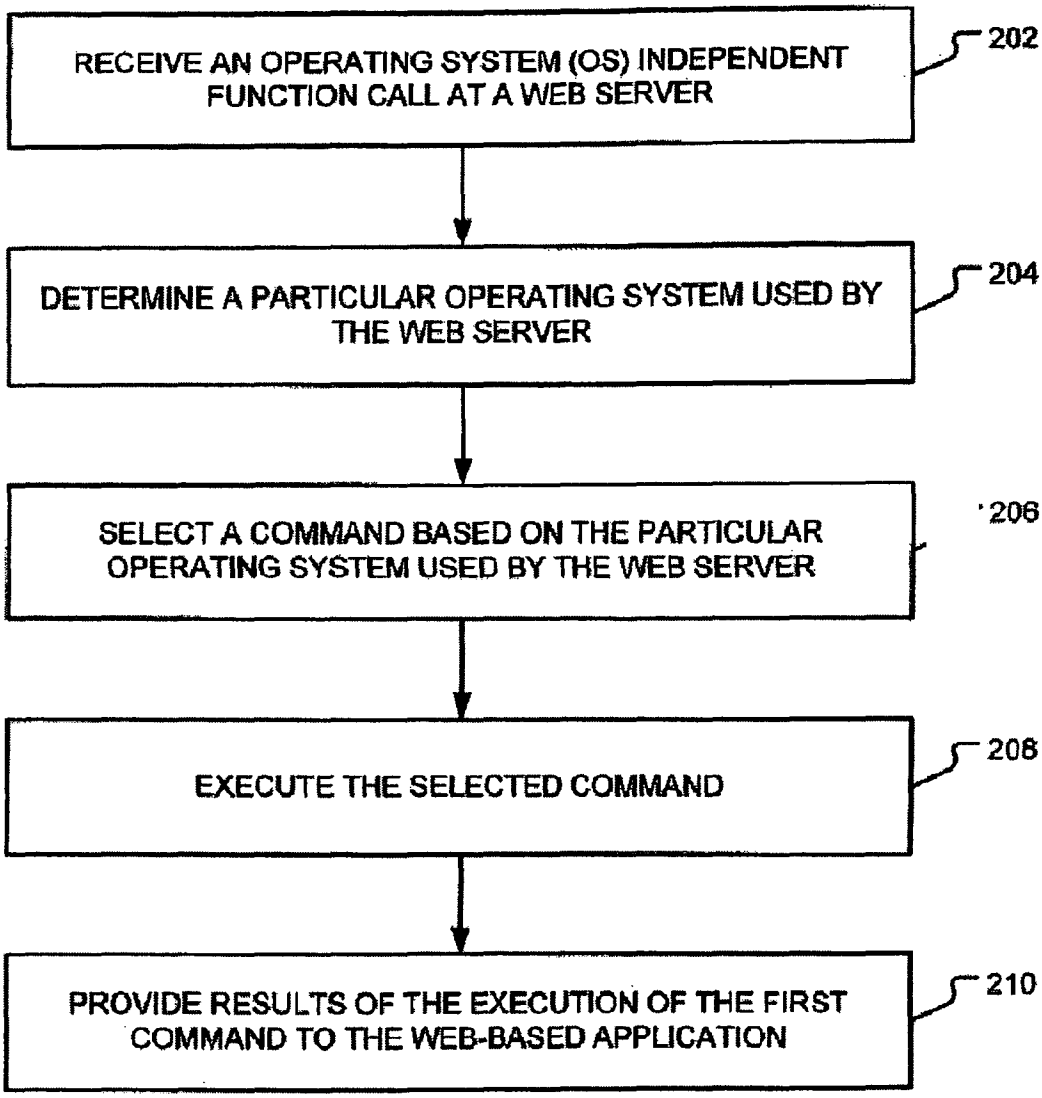
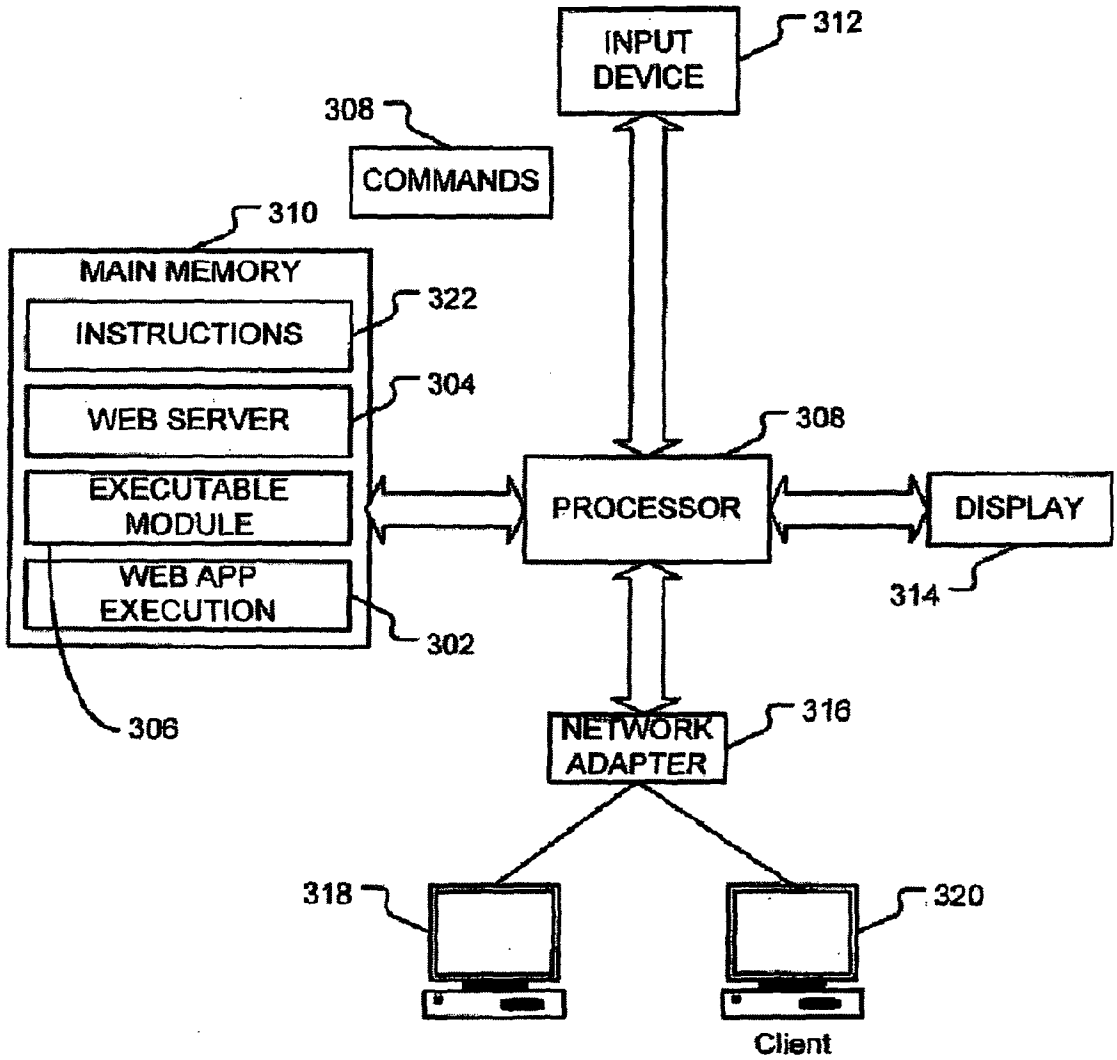

RECEIVE AN OPERATING SYSTEM (OS) INDEPENDENT FUNCTION CALL AT A WEB SERVER
202
DETERMINE A PARTICULAR OPERATING SYSTEM USED BY THE WEB SERVER
204
SELECT A COMMAND BASED ON THE PARTICULAR OPERATING SYSTEM USED BY THE WEB SERVER
206
EXECUTE THE SELECTED COMMAND
208
PROVIDE RESULTS OF THE EXECUTION OF THE FIRST COMMAND TO THE WEB-BASED APPLICATION
210

FIGURE 3


INPUT DEVICE
312
308
COMMANDS
310
MAIN MEMORY
INSTRUCTIONS
322
WEB SERVER
304
EXECUTABLE MODULE
WEB APP EXECUTION
302
306
PROCESSOR
308
DISPLAY
314
NETWORK ADAPTER
316
318
320
Client

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2010/058230

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/455 G06F9/46
ADD.

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)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, IBM-TDB, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 725 451 B1 (SCHUETZ WOLFGANG [DE] ET AL) 20 April 2004 (2004-04-20) column 1, line 54 - column 2, line 51 column 4, line 66 - column 5, line 12	1-20
X	WO 2004/104827 A2 (HEWLETT PACKARD DEVELOPMENT CO [US]; NOVAES REYNALDO [BR]; SCHEER ROQU) 2 December 2004 (2004-12-02) page 3, line 3 - line 13 page 4, line 9 - line 25 page 6, line 15 - line 28	1-20
X	US 2008/256564 A1 (FATHALLA DIAA [US]) 16 October 2008 (2008-10-16) paragraph [0011] - paragraph [0021] paragraph [0045] - paragraph [0047]	1-20

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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 document but 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	Date of mailing of the international search report
19 August 2010	25/08/2010

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Fax: (+31–70) 340–3016	Milasinovic, Goran

2

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2010/058230

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2002/191018 A1 (BROUSSARD SCOTT J [US]) 19 December 2002 (2002-12-19) paragraph [0014] - paragraph [0018] paragraph [0059] - paragraph [0061]	1-20
X	WO 2006/010263 A1 (RESEARCH IN MOTION LTD [CA]; STOUT CRAIG ALLEN [CA]; EMERY JEFFREY KEN) 2 February 2006 (2006-02-02) paragraph [0008] - paragraph [0013] paragraph [0022] - paragraph [0024]	1-20

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2010/058230

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6725451 B1	20-04-2004	WO 9931584 A1	24-06-1999
		EP 1040414 A1	04-10-2000
		JP 2002508565 T	19-03-2002
WO 2004104827 A2	02-12-2004	US 2004249947 A1	09-12-2004
US 2008256564 A1	16-10-2008	NONE	
US 2002191018 A1	19-12-2002	NONE	
WO 2006010263 A1	02-02-2006	CA 2526846 A1	30-01-2006
		EP 1766513 A1	28-03-2007