

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
16 June 2005 (16.06.2005)

PCT

(10) International Publication Number
WO 2005/054993 A2

- (51) International Patent Classification⁷: **G06F**
- (21) International Application Number:
PCT/US2004/039411
- (22) International Filing Date:
23 November 2004 (23.11.2004)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/525,523 25 November 2003 (25.11.2003) US
- (71) Applicant (for all designated States except US): **COMPUTER ASSOCIATES THINK, INC.** [US/US]; One Computer Associates Plaza, Islandia, NY 11749-7000 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **PARKER, Leo, F.** [US/US]; 316 Washington Avenue, Ocean Springs, MS

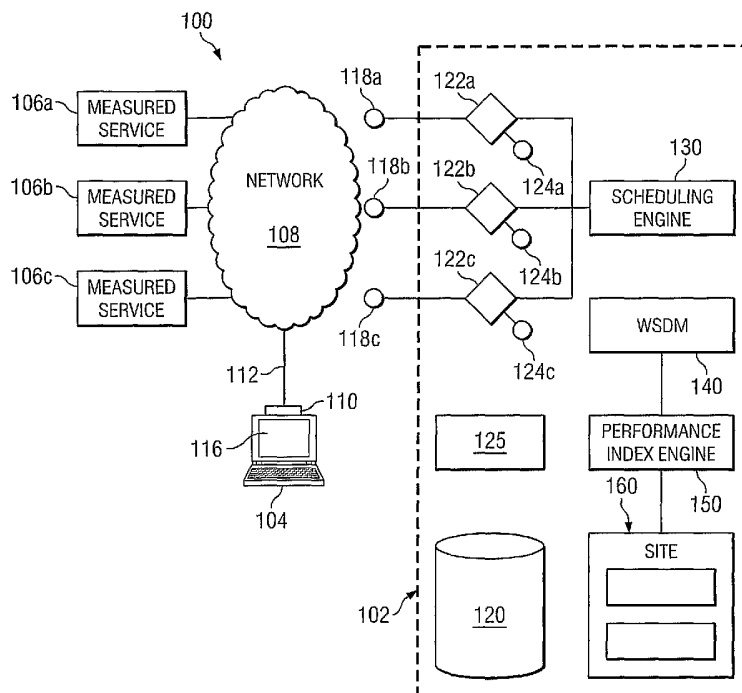
39564 (US). **UMANSKY, Vlad** [US/US]; 340 Parker Street, Newton, MA 02459 (US). **CAMINETSKY, Marc, R.** [CA/US]; 18 Peregrine Road, Newton, MA 02459 (US). **TCHEREVIK, Dmitri** [RU/US]; 22 Maple Road, Setauket, NY 11733 (US). **SRINIVAS, Davanum, M.** [IN/US]; 14 Firebrick Road, Sharon, MA 02067 (US). **KELLY, Dennis, J.** [US/US]; 63 Pleasant Street, Hingham, MA 02043 (US).

(74) Agent: **STALFORD, Terry, J.**; Fish & Richardson P.C., 5000 Bank One Center, 1717 Main Street, Dallas, TX 75201-4605 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, 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, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

[Continued on next page]

(54) Title: WEB SERVICE PERFORMANCE INDEX



(57) Abstract: A system (100) for providing a web service performance index is operable to collect service metric data from each of a plurality of web services (106), with at least one of the web services (106) comprising a remote web service (106). A web service performance index is generated based on the collected service metric data. The web service performance index is then published to a website (160).

WO 2005/054993 A2



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

Published:

— *without international search report and to be republished upon receipt of that report*

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

WEB SERVICE PERFORMANCE INDEX

REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Patent Application No.
5 60/525,523 filed November 25, 2003 entitled "Web Service Performance Index".

TECHNICAL FIELD

This disclosure relates generally to the field of computer networks and, more specifically, to a web service performance index.

BACKGROUND

10 Web services are generally self-contained, self-describing, modular software applications that can be published, located, and invoked across the Internet or other type of network. Current web services perform various computing and processing functions including simple requests and complicated business processes. Once web services are deployed onto the Internet, for example, other applications, including
15 other web services may locate and invoke deployed web services. Typically, web services make enterprise application integration simpler and less expensive and open new opportunities by increasing user reach and extending business partnerships.

SUMMARY

This disclosure provides a system and method for providing a web service
20 performance index. For example, the method may include collecting service metric data from each of a plurality of web services, with at least one of the web services comprising a remote web service. A web service performance index is generated based on the collected service metric data. The web service performance index is then published to a website.

DESCRIPTION OF THE DRAWINGS

25 FIGURE 1 illustrates an example system for providing a web service performance index in accordance with one embodiment of the present disclosure;

FIGURE 2 is a more detailed illustration of an example server for providing a web service performance index in accordance with one embodiment of the system of
30 FIGURE 1;

FIGURES 3A-C illustrate an example website and associated web pages in accordance with one embodiment of the system in FIGURE 1;

FIGURES 4A-D are diagrams illustrating example graphical representations of various performance indices and performance metrics in accordance with one embodiment of the website illustrated in FIGURES 3A-C; and

FIGURE 5 is a flowchart illustrating an example method for providing a web service performance index in accordance with one embodiment of the present disclosure.

DESCRIPTION OF EXAMPLE EMBODIMENTS

FIGURE 1 is a block diagram illustrating a system 100 for providing a centralized performance indicator for a plurality of remote web services 106 according to one embodiment of the present disclosure. At a high level, system 100 is a networked environment comprising at least one client 104, a server 102, and one or more web services 106, but may also be a standard computing environment or any other suitable environment. In general, system 100 comprises a local server 102 that compiles web service performance metrics, collectively termed service metric data, from the plurality of web services 106 via, for example, network 108 and provides the appropriate performance metrics and indices to client 104 in an understandable and efficient format. For example, system 100 may comprise an enterprise information technology system that utilizes web services 106 across the Internet, automatically retrieves or receives performance metrics for any number of web services 106, and dynamically provides the appropriate metrics to a system or network administrator via a workstation 102. The term “automatically,” as used herein, generally means that the appropriate processing is substantially performed by at least part of system 100. It should be understood that “automatically” further contemplates any suitable user interaction with system 100 without departing from the scope of this disclosure. The term “dynamically,” as used herein, generally means that certain processing is determined, at least in part, at run-time based on one or more variables.

Server 102 comprises any computer including Web Services Distributed Management (WSDM) application 140 and rendering engine 150 and communicably connected to at least one client 104 and/or one web service 106. For example, server 102 may be a general-purpose personal computer (PC), a Macintosh, a workstation, a Unix-based computer, a server computer, or any other suitable device. FIGURE 1 only provides one example of computers that may be used with the disclosure. For

example, although FIGURE 1 illustrates one server 102 that may be used with the disclosure, system 100 can be implemented using computers other than servers, as well as a server pool. The present disclosure contemplates computers other than general purpose computers as well as computers without conventional operating systems. As used in this document, the term "computer" is intended to encompass a personal computer, workstation, network computer, or any other suitable processing device. Computer server 102 may be adapted to execute any operating system including UNIX, Windows, or any other suitable operating system so long as server 102 remains communicably connected to client 104 and web services 106. Moreover, server 102 may include, alternatively or in combination, a web server or web server functionality. In short, server 102 may comprise any computer with software and/or hardware in any combination suitable to receive or retrieve web service performance metrics from web services 106, generate web pages and performance indices based on the performance metrics, and communicate the web pages to client 104 via network 108.

Web service 106 generally comprises a self-contained, self-describing, modular software application that can be published, located, referenced and/or invoked across network 108. Web service 106 performs various computing and processing functions including simple requests and complicated business processes. Once web service 106 is deployed onto or through network 108, other applications, including other web services 106, may locate and invoke deployed web service 106. In one embodiment, at least some of the web services 106 are software applications normally identified by a URI, as defined by IETF RFC 2396, operable to be defined, described and located by eXtensible Markup Language (XML) artifacts. In this embodiment, web service 106 is further operable to directly interact with other software applications using XML-based messages. Each web service 106 is associated with a web service provider and may include a plurality of web service methods. The web service provider may be a third-party, a remote but related business entity, or any other appropriate supplier or host of one or more web services 106. Often, server 102 will determine a performance index for each web service provider using some or all of the associated web services 106.

Network 108 facilitates wireless or wireline communication between computer server 102 and any other computer. Network 108 may communicate, for example, Internet Protocol (IP) packets, Frame Relay frames, Asynchronous Transfer Mode (ATM) cells, voice, video, data, and other suitable information between network
5 addresses. Network 108 may include one or more local area networks (LANs), radio access networks (RANs), metropolitan area networks (MANs), wide area networks (WANs), all or a portion of the global computer network known as the Internet, and/or any other communication system or systems at one or more locations.

Server 102 further includes service client interfaces 118, service clients 122,
10 WSDM management interfaces 124, memory 120, and processor 125. In one embodiment, a web service provider may register an associated web service 106 with server 102. Once registered, server 102 may automatically generate an associated service client 122 or receive a previously generated or manually entered service client 122 via network 108. Memory 120 may include any memory or database module and
15 may take the form of volatile or non-volatile memory including, without limitation, magnetic media, optical media, random access memory (RAM), read-only memory (ROM), removable media, or any other suitable local or remote memory component. Server 102 also includes processor 125. Processor 125 executes instructions and manipulates data to perform the operations of client 104 such as, for example, a
20 central processing unit (CPU), an application specific integrated circuit (ASIC) or a field-programmable gate array (FPGA). Although FIGURE 1 illustrates a single processor 125 in server 102, multiple processors 125 may be used according to particular needs, and reference to processor 125 is meant to include multiple processors 125 where applicable. In certain embodiments, processor 125 executes
25 one or more processes associated with scheduling engine 130, WSDM engine 140, and/or performance index engine 150.

Scheduling engine 130 could include any hardware, software, firmware, or combination thereof operable to schedule the collection of service metric data from one or more web services 106. Scheduling engine 130 may dynamically schedule the
30 collection of service metric data based on time, web service 106, web service provider, or any other appropriate variable. It will be understood that while scheduling engine 130 is illustrated as a single multi-tasked module, the features and

functionality performed by these engine may be performed by multiple modules such as, for example, a scheduler and a Java interface. WSDM engine 140 could include any hardware, software, firmware, or combination thereof operable to receive and process service metric data and present it to performance index engine 150 for further processing and rendering. Further, WSDM engine 140 may store the performance metrics in memory 120. It will be understood that while WSDM engine 140 is illustrated as a single multi-tasked module, the features and functionality performed by these engine may be performed by multiple modules.

Performance index engine 150 could include any hardware, software, firmware, or combination thereof operable to process collected service metric data, determine one or more performance indices, and present it to client 104 through GUI 116. Further, performance index engine 150 may render graphical representations of the performance indices and generate website 160 based on the collected service metric data. It will be understood that while performance index engine 150 is illustrated as a single multi-tasked module, the features and functionality performed by these engine may be performed by multiple modules such as, for example, an index calculation module and a rendering engine. Moreover, scheduling engine 130, WSDM engine 140, and/or performance index engine 150 may each comprise a child or sub-module of another software module without departing from the scope of this disclosure. In short, scheduling engine 130, WSDM engine 140, and/or performance index engine 150 comprise one or more software modules, individually or collectively, operable to create and present at least one web service performance index for multiple web services 106 via GUI 116.

Each client 104 is any computing device operable to present the user with web service performance data via a graphical user interface 116 (GUI). At a high level, illustrated client 104 includes at least GUI 116 and comprises an electronic computing device operable to receive, transmit, process and store any appropriate data associated with system 100. It will be understood that there may be any number of clients 104 communicably coupled to server 102. Further, "client 104" and "user of client 104" may be used interchangeably without departing from the scope of this disclosure. As used in this document, client 104 is intended to encompass a personal computer, workstation, network computer, kiosk, wireless data port, personal data assistant

(PDA), server, one or more processors within these or other devices, or any other suitable processing device. For example, client 104 may comprise a computer that includes an input device, such as a keypad, touch screen, mouse, or other device that can accept information, and an output device that conveys information associated with the operation of server 102 or clients 102, including digital data or visual information, via GUI 116. Both the input device and output device may include fixed or removable storage media such as a magnetic computer disk, CD-ROM, or other suitable media to both receive input from and provide output to users of clients 102 through GUI 116.

GUI 116 comprises a graphical user interface operable to allow the user of client 104 to interface with system 100 to view performance indices based on performance metrics, each from one web service 106. Generally, GUI 116 provides the user of client 104 with an efficient and user-friendly presentation of data provided by system 100, namely website 160. GUI 116 may comprise a plurality of frames or views having interactive fields, pull-down lists, and buttons operated by the user. In one embodiment, GUI 116 presents one or more web pages from performance index website 160 and receives commands from the user of client 104 via one of the input devices. As illustrated in more detail in FIGURES 3A-C, website 160 comprises a plurality of web pages operable to present client 104 with one or more performance indices. Often, website 160 is stored on server 102; but it will be understood that website 160 may be local or remote as well as distributed or integrated, without departing from the scope of the disclosure. Returning to GUI 116, it should be understood that the term graphical user interface may be used in the singular or in the plural to describe one or more graphical user interfaces and each of the displays of a particular graphical user interface. Further, GUI 116 contemplates any graphical user interface, such as a generic web browser, that processes information in system 100 and efficiently presents the information to the user. Server 102 can accept data from client 104 via the web browser (e.g., Microsoft Internet Explorer or Netscape Navigator) and return the appropriate HTML or XML responses using network 108.

Illustrated client 104 includes interface 110 for communicating with other computer systems, such as server 102, over network 108 in a client-server or other distributed environment. In certain embodiments, client 104 receives the web server

performance data from network 108, through interface 110, for display in GUI 116 or storage in memory or cache. Generally, interface 110 comprises logic encoded in software and/or hardware in a suitable combination and operable to communicate with network 108 via link 112. More specifically, interface 110 may comprise
 5 software supporting one or more communications protocols associated with communications network 108 or hardware operable to communicate physical signals.

In one aspect of operation, scheduling engine 130 initializes and loads one or more service clients 122 operable to retrieve, receive, or otherwise collect service metric data from web services 106 via service client interfaces 118. Based on this
 10 collected service metric data, performance index engine 150 calculates at least one performance index for each web service provider. For example, performance index engine 150 may calculate a consistency performance index for each provider or any other appropriate performance index based on service response time, transaction
 15 volume, fault frequency, and/or other performance metrics. The example consistency performance index is generally defined as one hundred minus the standard deviation of the average service performance over a determined timeframe, such as the past seven days. Accordingly, in this example, each provider's consistency index is based on the average weekly service response time for each web service 106 associated with the provider. The average weekly response time for a provider is calculated using the
 20 average response time for all web services 106 associated with the provider that are selected for being measured. For example purposes, a first provider includes measured web services 106a, 106b, and 106c, with each associated average weekly response times as follows:

	Week 1	Week 2	Week 3
First measured service 106a	14	12	15
Second measured service 106b	20	30	32
Third measured service 106c	16	16	18

25

Based on this example metric data, performance index engine 150 calculates the average weekly response times for the provider:

$$\text{Week 1} = (14+20+16)/3 = 16.7$$

$$\text{Week 2} = (12+30+16)/3 = 19.3$$

$$\text{Week 3} = (15+32+18)/3 = 21.7$$

Next, performance index engine 150 determines the standard deviation of the
 5 average weekly response times. The standard deviation for this three week period is
 3.559, calculated as follows:

$$(19+17+22)/3 = 19.33 = \text{average response}$$

$$(19.33-19)**2 = .1089$$

$$(19.33-17)**2 = 5.4289$$

$$10 \quad (19.33-22)**2 = 7.1289$$

$$(.1089 + 5.4289 + 7.1289) = 12.6667$$

$$\text{The square root of } 12.6667 = 3.559$$

The above example calculations result in a standard deviation of 3.559.
 Accordingly, the example performance index for the first provider over the three
 15 week period is $(100 - 3.559)$ or 96.441. It will be understood that the above example
 calculations for the performance index are for illustration purposes only and any
 appropriate mathematical or non-mathematical technique, algorithm, or logic,
 resulting in any numeric, non-numeric, graphical, or other suitable performance index
 may be used. Moreover, performance index engine 150 may use any performance
 20 metric of the collected service metric data to generate one or more of the performance
 indices. Once the appropriate performance indices are calculated or otherwise
 determined, performance index engine 150 generates at least one graphical
 representation of each performance index. These graphical representations are then
 embedded in one or more appropriate web pages and published to website 160 for
 25 subsequent access by client 104. Each graphical representation may be a chart, a
 table, a graph, a text log, or any other appropriate graphical element operable to be
 embedded in any document, file, or web page at any suitable time (such as on
 request). Example graphical representations and elements are illustrated in FIGURES
 4A-D. According to certain embodiments, server 102 may also automatically email
 30 the web pages, the graphical representations, or solely one or more performance
 indices to client 104. It will be understood that performance index engine 150 may
 collect real-time service metric data and aggregate it, over a period of time, prior to

determining the performance index without departing from the scope of the disclosure. Moreover, server 102 may be configured to automatically notify an administrator if the performance indices meet certain criteria (such as indicating a performance issue).

5 FIGURE 2 is a more detailed illustration of example server 102 for providing at least one web service performance index to client 104 in accordance with one embodiment of system 100. In this embodiment, server 102 includes two different machines: processing server 202a and generic web server 202b. In addition to certain example components in server 102, processing server 202a includes a memory 220,
10 Java management scheduling engine 230, and Java management 235.

 As with memory 120, memory 220 may include any memory, hard drive, or database module and may take the form of volatile or non-volatile memory including, without limitation, magnetic media, optical media, random access memory (RAM), read-only memory (ROM), removable media, or any other suitable local or remote
15 memory component. Illustrated memory 220 stores, or includes references to, WSDM DB 224, web page images 226, and Java management DB 228. WSDM DB 224 and Java management DB 228 may each comprise a relational database described in terms of SQL statements or scripts or an object-oriented database. Relational
20 databases often use sets of schemas to describe the tables, columns, and relationships in the tables using basic principles known in the field of database design. In another embodiment, WSDM DB 224, web page images 226, and Java management DB 228 may comprise XML documents, HTML files, JavaScripts, flat files, Btrieve files, or comma-separated-value (CSV) files.

 In the illustrated embodiment, service metric data 210 is obtained from web
25 services 106 via regularly scheduled synthetic transactions according to Java management scheduling engine 230. Synthetic transaction creation and scheduling are driven by service clients 222 typically created using Java management 235 and Java management DB 228. Each service client 222 is instrumented with native
observers 223, with each observer operable to watch a particular performance metric.
30 Service metric data, gathered by observers 223 associated with service client 222, are aggregated by the WSDM 140 and placed in WSDM DB 224. For each web service 106, performance index engine 150 extracts individual performance metrics for the

average weekly and daily service response times and generates graphical representations of the calculated performance indices. According to certain embodiments, these values are pre-calculated and in WSDM DB 224.

FIGURES 3A-C illustrate example website and associated web pages in accordance with one embodiment of system 100. FIGURE 3A illustrates an example layout of website 160. Illustrated website 160 includes a home page 310 and in one embodiment at least one child page such as, for example, provider pages 312 and informational pages 314. Generally, home page 310 provides an entry to website 160 for client 104. Home page 310, described in more detail in FIGURE 3B, presents basic information in regard to website 160, as well as one or more comparative performance indices. Each provider page 312 displays descriptive information about the relevant web service provider and the one or more performance indices based on the service metric data collected about the provider. Informational pages 314 comprise any other suitable web pages for presenting informative data to client 104 such as, for example, search pages, contact pages, help pages, website index, links to other websites, legal information, site operation, performance index calculation, web service general description, and others.

FIGURE 3B illustrates an example home page 310. According to one embodiment, the home page includes a banner, an information block, a table of providers, one or more comparative graphical representations of multiple performance indices, and a "more information" panel. The banner generally identifies or promotes the website. The information block providing a brief introduction to the site's function and links to pages containing more detailed descriptions of the website and associated web services management. The table of providers often contains links to provider pages 312 displaying service performance charts and tables. Each link is typically a provider or collection name, but may be any other operative or descriptive word or phrase. The "more information" panel with a combo box containing links to pages provides access to site and web service management information sources.

FIGURE 3C illustrates an example provider page 312. Generally provider page 312 describes the collection of web services associated with the provider and displays some performance measurements for each web service 106 using one or more graphical representations and performance indices. According to the illustrated

embodiment, provider page 312 includes an information panel, a link to the web services description language (WSDL), a service description table, and one or more service performance charts and tables or other graphical representations for performance indices. The information panel includes the web service provider name and a brief description of the functions performed by some or all web services associated with the provider. The service description table provides a brief introduction to the various web services associated with the web service provider. It will be understood that illustrated web pages 310 and 312 are for example purposes only. Accordingly, website 160 may include home page 310 and provider pages 312 in any format or descriptive language and each page may present any appropriate data in any layout without departing from the scope of the disclosure.

FIGURES 4A-D are diagrams illustrating example graphical representations of various performance indices and performance metrics in accordance with one embodiment of the website. Generally, each graphical representation is embedded in one of the web pages, 310 or 312, for publication to the website. FIGURE 4A is an example performance index bar chart comparing the current consistency performance indexes of all providers. The bar chart contains one bar for each provider. The illustrated Y-axis is the most recent consistency performance index for each provider, calculated according to any appropriate technique. FIGURE 4B is an example service description table comprising a list of the web service methods offered by the provider that are currently being monitored by server 102. This list may display some or all of web services offered by the web service provider without departing from the scope of the disclosure. The service description table may be a multi-dimensional data structure that includes at least one web service record. Each record includes multiple columns such as, for example, service method name (“GetMap”) and brief service description (“returns a map image, a map view, and hot area definitions based on map options”). The service method name often will include an on-page link to the service performance charts associated with the service. It will be understood that each record may include none, some, or all of the example columns.

FIGURES 4C-D illustrate example a service performance chart and tables presenting recent service performance. According to one embodiment of provider page 312, there will be one chart and one table for each service within the provider’s

collection. FIGURE 4C, in particular, illustrates one example of the table presenting average daily performance for web service 106 for the past week. Accordingly, this example table includes at least two columns: one displaying the date or day of week and the second presenting the average daily service response times. FIGURE 4D
5 illustrates one example of the line chart presenting the average weekly performance of web service 106 for the past one to ten weeks.

FIGURE 5 is a flowchart illustrating an example method 500 for providing a web service performance index in accordance with one embodiment of the present disclosure. Generally, method 500 describes server 102 retrieving service metric data
10 from a plurality of web services 106, processing the metric data to determine individual and comparative performance indices, and presenting the indices to a client 104 over network 108 through website 160. Method 500 is described in respect to system 100 and, in particular, to server 102. However, any other suitable system or computing device may use appropriate embodiments of method 500 to present at least
15 one web service performance index to client 104 without departing from the scope of this disclosure.

Method 500 begins at step 502 where scheduling engine 130 retrieves registered service clients 122. At step 504, performance index engine 150 generates homepage 310 based on the retrieved service clients 122. It will be understood that
20 any generation of any web page may comprise generating a web page based on the appropriate HTML file, updating the content of a previously generated web page, or any other appropriate painting or graphical process. Scheduling engine 130 loads at least a subset of the registered service clients at step 506. Next, each web service observer associated with the loaded service client 122 is initialized at step 508. Once
25 the appropriate software modules are loaded and initialized, service metric data is collected from the plurality of web services associated with service clients 122 in steps 510 through 526.

Scheduling engine 130 selects the first web service 106a based on the loaded service client 122a at step 510. Next, service client 122a collects, retrieves, or
30 receives service metric data from web service 106a via service client interface 118a at step 512. It will be understood that this collection may occur at any appropriate time such as, for example, in response to a request from an administrator, automatically at

a time determined by scheduling engine 130, or dynamically based on any suitable run-time parameter. Once collected, WSDM engine 140 receives the collected service data and parses it into individual performance metrics at step 514, as well as perform other suitable processing (such as data cleansing/sorting and such). These
5 parsed performance metrics are communicated to performance index engine 150. At step 516, performance index engine 150 then generates at least one graphical representation of each performance metric. As described above, each graphical representation may be a chart, a table, a graph, a text log, or any other appropriate graphical element operable to be embedded in any document, file, or web page. Next,
10 at step 518, performance index engine 150 generates a service web page 312 for the appropriate web service 106a. Performance index engine 150 then embeds each generated graphical representation in provider page 312 at step 520. The performance metrics are then stored in memory 120 at step 522 for subsequent access or query. Next, at decisional step 524, scheduling engine 130 determines if there are more
15 loaded service clients 122, such as service client 122b. If there are more loaded service clients 122, then scheduling engine 130 selects the next web service, such as web service 106b, at step 526 and processing returns to step 512. Once an appropriate amount of service metric data has been collected by server 102, one or more web service performance indices are generated in steps 528 through 536.

20 After any appropriate amount of service metric data has been collected by server 102, then performance index engine 150 processes the collected service metric data to provide an overall or comparative picture of how various web services 106 are performing. For example, performance index engine 150 merges the collected service metric data by performance metric at step 528. This merging may include parsing and
25 storing the collected service metric data into separate tables for each performance metric. Next, at step 530, performance index engine 150 ranks, sorts, or otherwise compares each performance metric within the merged performance metric group. At step 532, performance index engine 150 generates a ranked graphical representation for each metric group. Performance index engine 150 then embeds each ranked or
30 comparative graphical representation in the performance index home page 310 at step 534. At step 536, performance index engine 150 then publishes generated web pages 310 and 312 to the website for appropriate access via network 108 by client 104.

Server 102 receives a request for access to the website from client 104 at step 538. Server 102 analyzes the request and determines various characteristics of client 104 and the request such as, for example, client authority, request parameters, client pay tier, or any other appropriate characteristic at step 540. Based on the determined characteristics, server 102 selects at least a subset of the published web pages 310 and 312 at step 542. If client 104 has appropriate authority, then server 102 grants client 104 access to the selected subset of published web pages at step 544. Next, at step 546, server 102 communicates any appropriate content of the selected subset of published web pages to requesting client 104.

10 The preceding flowchart focuses on the operation of example system 100 described in FIGURES 1, 2, and 3 as these example diagrams illustrate various functional elements that implement some or all of the preceding techniques for creating and presenting at least one web service performance index via GUI 116. However, as noted above, system 100 contemplates using any suitable combination and arrangement of functional elements for providing these operations, and these techniques can be combined with other techniques as appropriate. Further, various changes may be made to the preceding flowcharts without departing from the scope of this disclosure. In other words, many of the steps in these flowcharts may take place simultaneously and/or in different orders than as shown. Moreover, system 100 may implement methods with additional steps, fewer steps, and/or different steps, so long as the methods remain appropriate. For example, the initialization steps in method 500 may occur only upon power-up of server 102, collection and processing of the service metric data from multiple web services 106 may occur in parallel as opposed to serially, the collected service metric data may be aggregated over a period of time before being published to the website, or any other appropriate customization or implementation within the scope of this disclosure.

Although this disclosure has been described in terms of certain embodiments and generally associated methods, alterations and permutations of these embodiments and methods will be apparent to those skilled in the art. Accordingly, the above description of example embodiments does not define or constrain this disclosure. Other changes, substitutions, and alterations are also possible without departing from the spirit and scope of this disclosure.

WHAT IS CLAIMED IS:

1. A method for providing a web service performance index comprises:
collecting service metric data from each of a plurality of web services, at least
one of the web services comprising a remote web service;
5 generating a web service performance index based on the collected service
metric data; and
publishing the web service performance index to a website.
2. The method of Claim 1, at least one of the web services comprising a
public web service and a second one of the web services comprising a private web
10 service.
3. The method of Claim 1, the service metric data comprising at least one
of the following performance metrics:
service response time;
transaction volume; and
15 fault frequency.
4. The method of Claim 1, wherein generating a web service performance
index comprises generating at least one graphical representation for the service metric
data collected from each web service.
5. The method of Claim 4, further comprising automatically e-mailing the
20 one or more generated graphical representations to a client.
6. The method of Claim 4, the graphical representation selected from the
group consisting of at least one of a chart, a table, a graph, and a log.
7. The method of Claim 1, wherein collecting service metric data
comprises collecting real-time performance metrics.
- 25 8. The method of Claim 1, further comprising:
merging the collected service data based on individual performance metrics;
ranking the merged service data; and
generating at least one graphical representation based on the ranked service
data.
- 30 9. The method of Claim 1, further comprising generating a plurality of
web pages for publishing to the website based on the web service performance index.

10. The method of Claim 9, each web service associated with a provider and comprising at least one web service method.

11. The method of Claim 10, the plurality of web pages comprising a home page and a plurality of provider pages, each provider page associated with one provider from the plurality of web services and the home page comprising a link to each provider page.

12. The method of Claim 9, further comprising:
receiving a request from a client to access the website; and
communicating at least a portion of the generated web pages to the client based on at least one client characteristic and at least partially in response to the request.

13. The method of Claim 1, each web service comprising a registered web service associated with a particular service client.

14. The method of Claim 13, further comprising:
receiving a registration for a new web service;
automatically generating a new service client associated with the new web service; and
adding the new web service to the plurality of web services to collect service metric data from.

15. Software for providing a web service performance index operable to:
collect service metric data from each of a plurality of web services, at least one of the web services comprising a remote web service;
generate a web service performance index based on the collected service metric data; and
publish the web service performance index to a website.

16. The software of Claim 15, at least one of the web services comprising a public web service and a second one of the web services comprising a private web service.

17. The software of Claim 15, the service metric data comprising at least one of the following performance metrics:

service response time;
transaction volume; and
5 fault frequency.

18. The software of Claim 15, wherein the software operable to generate a web service performance index comprises software operable to generate at least one graphical representation for the service metric data collected from each web service.

19. The software of Claim 18, further operable to automatically e-mail the
10 one or more generated graphical representations to a client.

20. The software of Claim 18, the graphical representation selected from the group consisting of at least one of a chart, a table, a graph, and a log.

21. The software of Claim 15, wherein the software operable to collect service metric data comprises software operable to collect real-time performance
15 metrics.

22. The software of Claim 15, further operable to:
merge the collected service data based on individual performance metrics;
rank the merged service data; and
generate at least one graphical representation based on the ranked service data.

23. The software of Claim 15, further operable to generate a plurality of
20 web pages for publishing to the website based on the web service performance index.

24. The software of Claim 23, each web service associated with a provider and comprising at least one web service method.

25. The software of Claim 24, the plurality of web pages comprising a
25 home page and a plurality of provider pages, each provider page associated with one provider from the plurality of web services and the home page comprising a link to each provider page.

26. The software of Claim 23, further operable to:
receive a request from a client to access the website; and
30 communicate at least a portion of the generated web pages to the client based on at least one client characteristic and at least partially in response to the request.

27. The software of Claim 15, each web service comprising a registered web service associated with a service client.

28. The software of Claim 27, further operable to:

receive a registration for a new web service;

5 automatically generate a new service client associated with the new web service; and

add the new web service to the plurality of web services to collect service metric data from.

29. A server for providing a web service performance index comprises:

10 memory operable to store a plurality of service clients, each service client associated with a web service; and

one or more processors operable to:

collect service metric data from a subset of a plurality of web services associated with each service client, at least one of the web services comprising a

15 remote web service;

generate a web service performance index based on the collected service metric data; and

publish the web service performance index to a website.

30. The server of Claim 29, at least one of the web services comprising a public web service and a second one of the web services comprising a private web service.

31. The server of Claim 29, the service metric data comprising at least one of the following performance metrics:

service response time;

25 transaction volume; and

fault frequency.

32. The server of Claim 29, wherein the processors operable to generate a web service performance index comprises processors operable to generate at least one graphical representation for the service metric data collected from each web service.

30 33. The server of Claim 32, the processors further operable to automatically e-mail the one or more generated graphical representations to a client.

34. The server of Claim 32, the graphical representation selected from the group consisting of at least one of a chart, a table, a graph, and a log.

35. The server of Claim 29, wherein the processors operable to collect service metric data comprises processors operable to collect real-time performance
5 metrics.

36. The server of Claim 29, the processors further operable to:
merge the collected service data based on individual performance metrics;
rank the merged service data; and
generate at least one graphical representation based on the ranked service data.

10 37. The server of Claim 29, the processors further operable to generate a plurality of web pages for publishing to the website based on the web service performance index.

38. The server of Claim 37, each web service associated with a provider and comprising at least one web service method.

15 39. The server of Claim 38, the plurality of web pages comprising a home page and a plurality of provider pages, each provider page associated with one provider from the plurality of web services and the home page comprising a link to each provider page.

20 40. The server of Claim 29, each web service comprising a registered web service.

41. The server of Claim 40, the processors further operable to:
receive a registration for a new web service;
automatically generate a new service client associated with the new web service, thereby registering the web service; and
25 add the new web service to the plurality of web services to collect service metric data from.

42. The server of Claim 29, the processors further operable to:
receive a request from a client to access the website; and
communicate a portion of the web service performance index based on
30 at least one client characteristic and at least partially in response to the request.

43. The server of Claim 29, each service client associated with a service client interface, the service client interface operable to communicate with the web service.

44. The server of Claim 29, each service client associated with a Web Services Distributed Management (WSDM) interface.

45. A system for providing a web service performance index comprises:
means for collecting service metric data from each of a plurality of web services, at least one of the web services comprising a remote web service;
means for generating a web service performance index based on the collected service metric data; and
means for publishing the web service performance index to a website.

46. Software for providing a web service performance index operable to:
collect service metric data from each of a plurality of web services, at least one of the web services comprising a remote web service;

generate a web service performance index based on the collected service metric data;

publish the web service performance index to a website, the website comprising a plurality of pages;

receive a registration for a new web service;

add the new web service to the plurality of web services;

receive a request from a client to access the website; and

communicate at least a portion of the web pages to the client based on at least one client characteristic and at least partially in response to the request.

25

FIG. 1

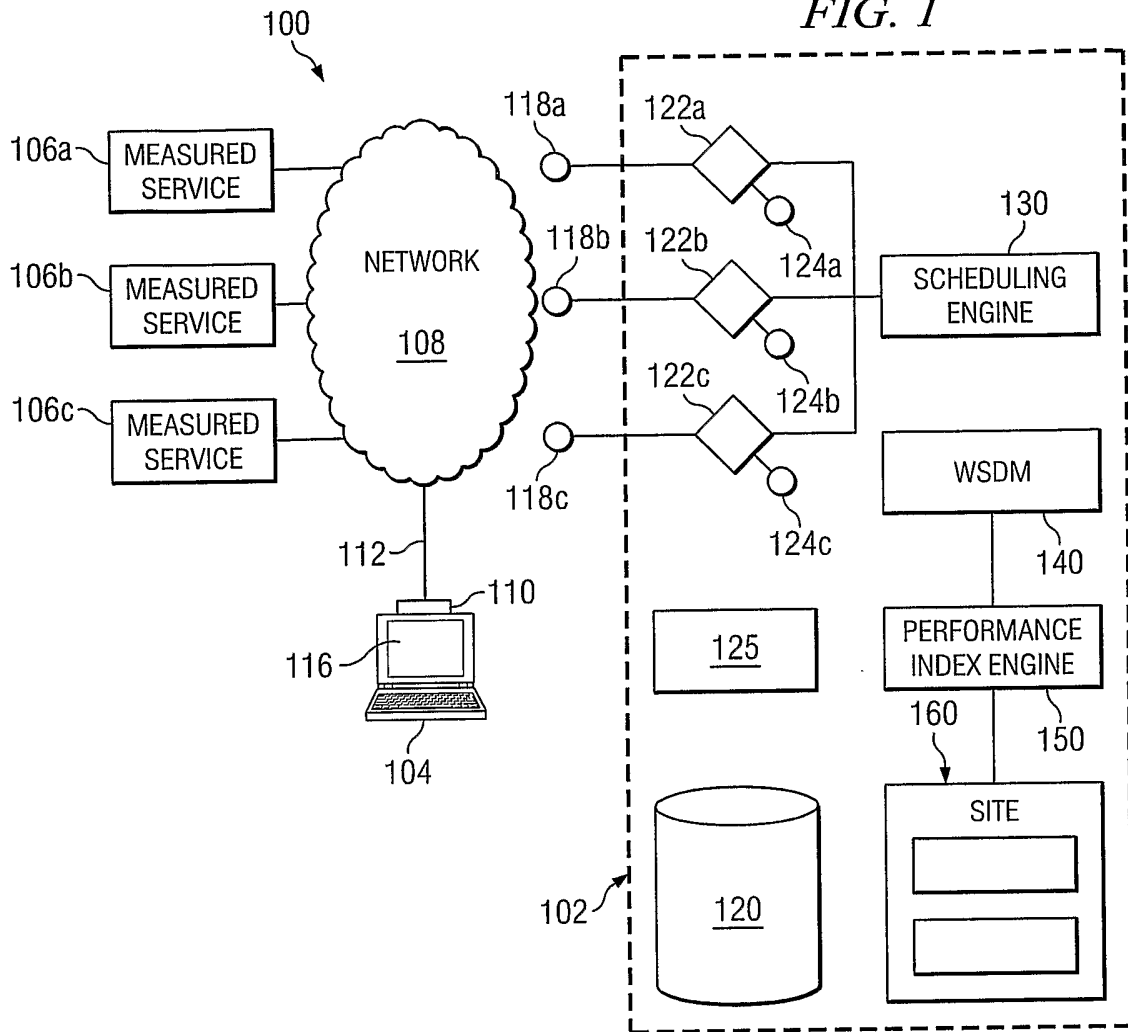
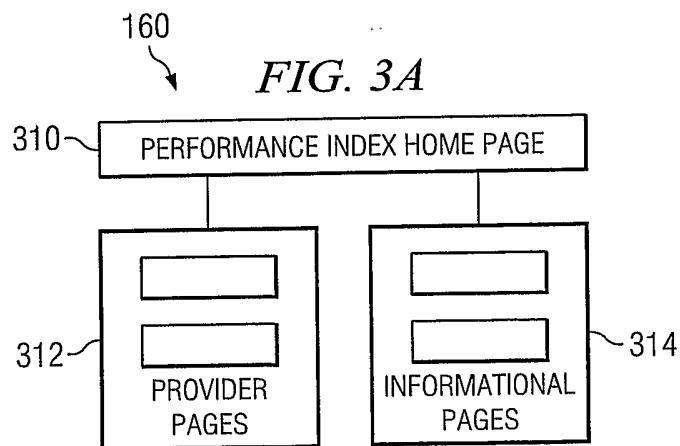


FIG. 3A



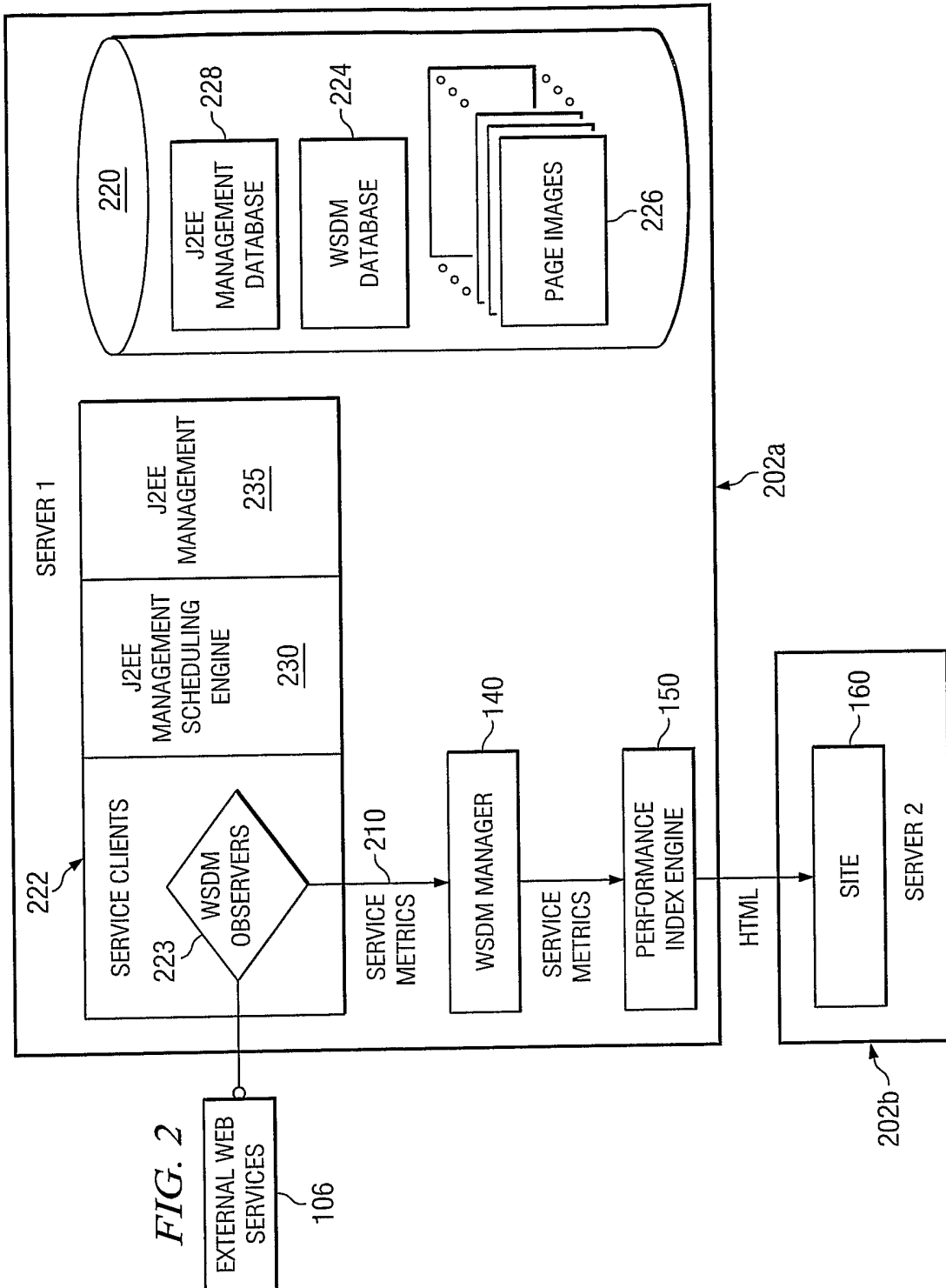


FIG. 3B

310

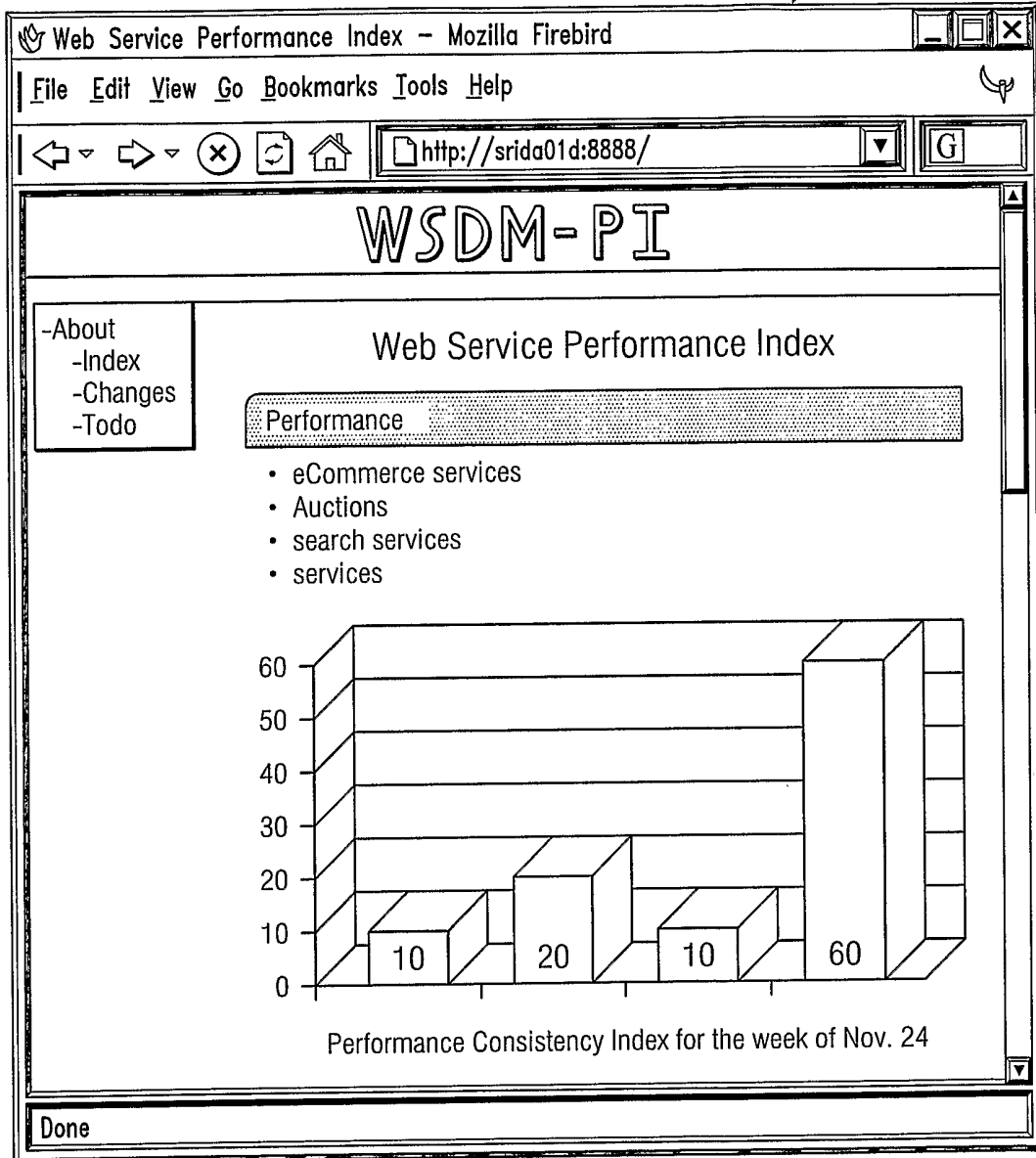
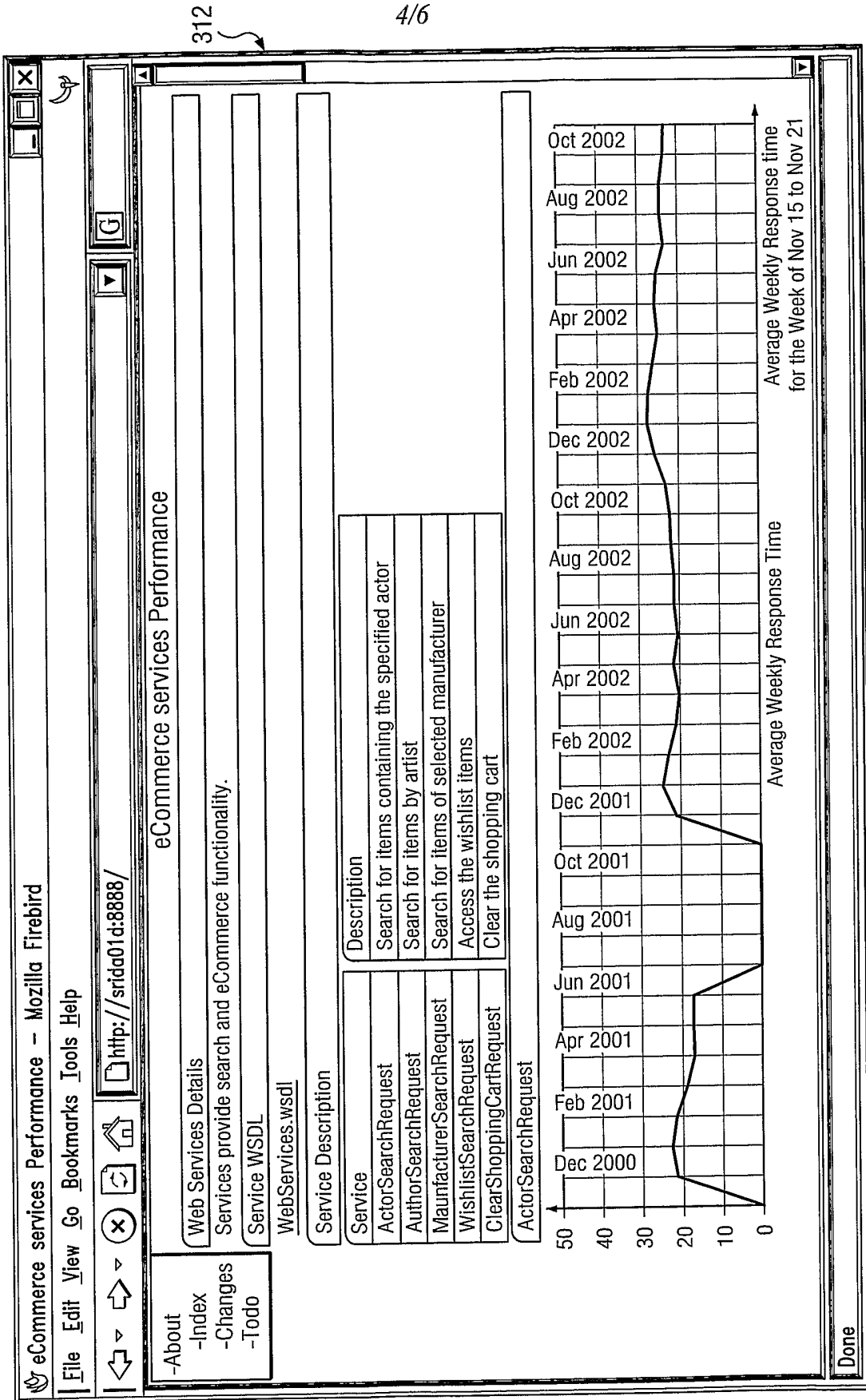


FIG. 3C



312

4/6

FIG. 4A

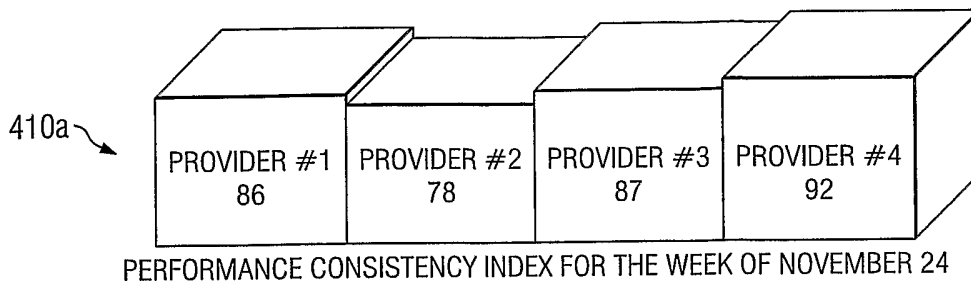


FIG. 4B

410b

SERVICE	DESCRIPTION
ActorSearchRequest	SEARCH FOR ITEMS CONTAINING SPECIFIED ACTOR
ArtistSearchRequest	SEARCH FOR ITEMS BY ARTIST
ManufacturerSearchRequest	SEARCH FOR ITEMS OF SELECTED MANUFACTURER
WishlistSearchRequest	ACCESS WISHLIST ITEMS
ClearShoppingCartRequest	CLEAR THE SHOPPING CART

410c

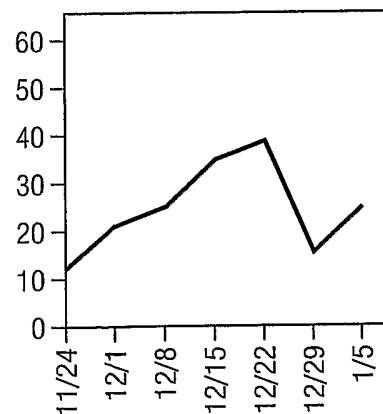
FIG. 4C

DAY OF WEEK	AVERAGE RESPONSE TIME (msec)
MONDAY - 11/15	36
TUESDAY - 11/16	42
WEDNESDAY - 11/17	24
THURSDAY - 11/18	19
FRIDAY - 11/19	37
SATURDAY - 11/20	43
SUNDAY - 11/21	29

ActorSearchRequest
AVERAGE RESPONSE TIME FOR THE WEEK OF NOVEMBER 15 TO 21

FIG. 4D

410d



ActorSearchRequest
AVERAGE WEEKLY RESPONSE TIME

FIG. 5 6/6

