



- (51) **International Patent Classification:** Not classified
- (21) **International Application Number:**  
PCT/US2010/054574
- (22) **International Filing Date:**  
28 October 2010 (28.10.2010)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**  
12/616,796 12 November 2009 (12.11.2009) US
- (71) **Applicant (for all designated States except US):** MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).
- (72) **Inventors:** LIENSBERGER, Christian; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). KHOSRAVY, Moe; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(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, 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, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Declarations under Rule 4.17:**

[Continued on next page]

(54) **Title:** WEB SERVICE INTERFACE AND QUERYING

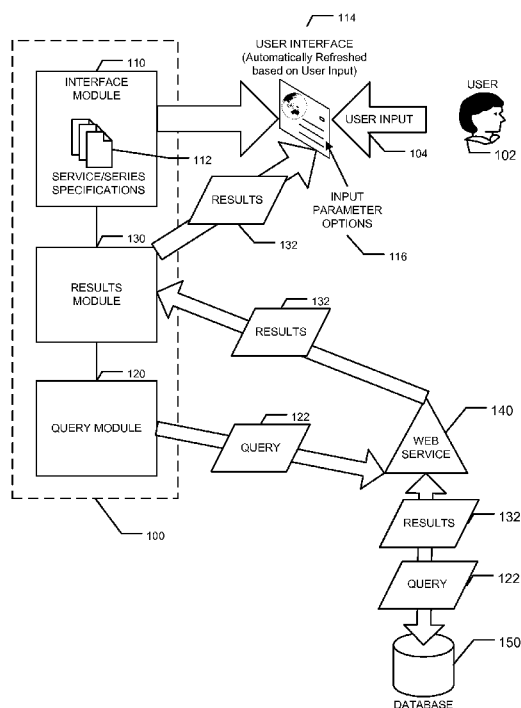


FIG. 1

(57) **Abstract:** Methods, systems, and computer-readable media are disclosed to generate an interface for a web service and to query the web service. A particular method includes generating a user interface associated with a web service. The user interface includes one or more input parameter options associated with the web service. The user interface is generated based on a service specification and one or more series specifications associated with the service specification. User input is received via the user interface, where the user input includes a selection of at least one of the one or more input parameter options. A query of the web service is generated based on the received user input, where the query reflects at least one selected input parameter option. The method includes invoking the generated query with respect to the web service to receive results and automatically adding the received results to the user interface.



- 
- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*
- Published:**
- *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

## WEB SERVICE INTERFACE AND QUERYING

### BACKGROUND

[0001] Web services have become an increasingly popular way to achieve inter-computer communication over a network. Web services are typically accessed via queries that are defined based on application programming interfaces (APIs). However, learning how to use such APIs may be a time-consuming process and may require users to review a significant amount of web service documentation. Further, web servers that host web services often offer dozens of APIs, or more. API-based web service access typically requires computer programming knowledge. Due to the lack of visual feedback in text-based computer programs, API-based web service access may be unintuitive. Thus, constructing a web service query can be a long and demanding process for users.

### SUMMARY

[0002] The present disclosure describes automated interface generation for a web service where the interface enables queries of the web service based on user input. Thus, web service operation may be wholly driven via user input at the interface. The web service interface may be generated based on service and series specifications (e.g., XML files). In response to user input, the web service interface may be automatically refreshed to reflect the received input, thereby providing an “active” interface to the web service. When a query of the web service is invoked, the interface may display a preview of the results generated by the query. Interface generation and query building enables users to leverage web services without learning complex application programming interfaces (APIs). In addition, uniform resource identifiers (URIs) and program classes may be automatically generated and retrievable by users to re-execute previously invoked queries.

[0003] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

### BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIG. 1 is a diagram to illustrate a particular embodiment of a system to generate an interface for a web service and to query the web service;

[0005] FIG. 2 is a diagram to illustrate another particular embodiment of a system to generate an interface for a web service and to query the web service;

[0006] FIG. 3 is a diagram to illustrate a particular embodiment of a service specification that may be used with the system of FIG. 1 or the system of FIG. 2;

[0007] FIG. 4 is a diagram to illustrate a particular embodiment of a series specification that may be used with the system of FIG. 1 or the system of FIG. 2;

5 [0008] FIG. 5 is diagram to illustrate a particular embodiment of automatically refreshing an interface for a web service based on user input;

[0009] FIG. 6 is a diagram to illustrate a particular embodiment of an interface for a web service including previewing results of a query of the web service;

10 [0010] FIG. 7 is a flow diagram to illustrate a particular embodiment of a method to generate an interface for a web service and to query the web service;

[0011] FIG. 8 is a flow diagram to illustrate a particular embodiment of a method of receiving an interface for a web service and generating a query of the web service; and

[0012] FIG. 9 is a block diagram of a computing environment including a computing device operable to support embodiments of computer-implemented methods, computer  
15 program products, and system components as illustrated in FIGS. 1-8.

#### DETAILED DESCRIPTION

[0013] Systems, methods, and computer-readable media to generate an interface for a web service and to query the web service are disclosed. In a particular embodiment, a system includes a memory and a processor coupled to the memory. The processor is configured  
20 to retrieve instructions from the memory that cause execution of an interface module, a query module, and a results module. The interface module is configured to generate a user interface associated with a web service, where the user interface includes one or more input parameter options associated with the web service. The interface module is also configured to receive user input via the user interface, where the user input includes a  
25 selection of at least one of the one or more input parameter options. The interface module is further configured to automatically refresh the one or more input parameter options at the user interface based on the received user input. The query module is configured to generate a query of the web service based on the received user input. The query reflects the at least one selected input parameter option. The query module is also configured to  
30 invoke the generated query with respect to the web service to generate results, and the results module is configured to automatically display the results at the user interface.

[0014] In another particular embodiment, a method includes generating a user interface associated with a web service. The user interface includes one or more input parameter options associated with the web service. The user interface is generated at a computer

system based on an extensible markup language (XML) service specification and one or more XML series specifications associated with the XML service specification.

Alternately, the specifications may be represented in any computer-recognizable language (e.g., text file, rich text format (RTF), or hyper text markup language (HTML)).

5 The method also includes receiving user input via the user interface, where the user input includes a selection of the one or more input parameter options. The method further includes generating a query of the web service based on the received user input, where the query reflects the selection of the one or more input parameter options. The method includes invoking the generated query with respect to the web service and automatically  
10 adding results of the invoked query to the user interface.

**[0015]** In another particular embodiment, a computer-readable medium is disclosed. The computer-readable medium includes instructions, that when executed by a computer, cause the computer to receive a user interface associated with a web service from a web server, where the user interface includes one or more input parameter options associated

15 with the web service. The user interface is generated based on an extensible markup language (XML) service specification and one or more XML series specifications associated with the XML service specification. The computer-readable medium also includes instructions, that when executed by the computer, cause the computer to display the user interface and to receive user input at the user interface. The user input includes a selection of at least one of the one or more input parameter options. The computer-  
20 readable medium further includes instructions, that when executed by the computer, cause the computer to transmit the user input to a web server and to receive a modified user interface from the web server, where the modified user interface reflects the user input.

The computer-readable medium includes instructions, that when executed by the  
25 computer, cause the computer to display the modified user interface, to generate a query of the web service based on the received user input, and to transmit the generated query to the web server. The computer-readable medium also includes instructions, that when executed by the computer, cause the computer to receive results based on an invocation of the generated query from the web server and to display the results at the modified user  
30 interface.

**[0016]** FIG. 1 is a diagram to illustrate a particular embodiment of a system 100 to generate an interface (e.g., an illustrative user interface 114) for a web service 140 and to query the web service 140. The web service 140 is communicatively coupled to the

system 100. The web service 140 may also be communicatively coupled to one or more data sources (e.g., an illustrative database 150).

[0017] The system 100 includes an interface module 110 configured to generate the user interface 114 for the web service 140, where the user interface 114 includes one or more input parameter options 116. In a particular embodiment, the user interface 114 is generated based on service and series specifications 112 associated with the web service 140. For example, the service and series specifications 112 may be extensible markup language (XML) files that include information useable to build a query of the web service 140. Information useable to build a query of the web service 140 may include a paging permission, a service title, a service description, list parameters, series uniform resource identifier (URI) templates, series identifiers, series source paths, and parameter types/bounds/options (e.g., string, integer, floating point, and Boolean). The series source paths included in a service specification may be useable to generate URIs associated with a particular series specification. For example, a series source path may be appended to a series URI template to generate a URI for a particular series specification. An illustrative XML service specification is further described with respect to FIG. 3. An illustrative XML series specification is further described with respect to FIG. 4. Illustrative user interfaces are further described with respect to FIGS. 5-6.

[0018] The interface module 110 is also configured to receive user input 104 (e.g., from the user 102) via the user interface 114. The user input 104 may include a selection of one or more of the input parameter options 116. Upon receiving the user input 104, the interface module 110 may automatically refresh the user interface 114 to reflect the user input 104. In a particular embodiment, the user interface 114 is refreshed in accordance with one or more auto-refresh URIs included in the service and series specifications 112. For example, automatically refreshing the user interface 114 based on the user input 104 may include refreshing the display of graphical user interface (GUI) elements associated with the input parameter options 116 (e.g., a drop down menu associated with a list parameter) or updating one or more bounds (e.g., minimum and maximum values of numeric values or a maximum length of string values) of the input parameter options 116. Auto-refresh operations at an interface associated with a web service are further described with respect to FIG. 5.

[0019] The system 100 includes a query module 120 that is configured to generate a query 122 of the web service 140 based on the received user input 104. The query 122 reflects the selection of the one or more of the input parameter options 116. The query module

120 is also configured to invoke the generated query 122 with respect to the web service 140 to generate results 132 of the query invocation. For example, the web service 140 may provide data querying abilities with respect to the database 150, and the query module 120 may invoke the query 122 (e.g., a query that filters certain tables of the database 150) to generate the results 132 (e.g., the resulting filtered data from the selected tables of the database 150). In a particular embodiment, one or more of the web service 140 and the database 150 are remote to the system 100 and are accessible via one or more data communications networks (e.g., the Internet). It should be noted that although the web service 140 and the database 150 are illustrated as being remote from each other, the web service 140 and the database 150 may instead be located at a single computing device, such as a web server or database server, or at a single location.

**[0020]** In a particular embodiment, the query module 120 is further configured to automatically generate one or more program classes (e.g., C++ classes, C# classes, or Java classes) that are useable to invoke the query 122. For example, the selected one or more input parameter options 116 may be encapsulated within the program classes or may be provided as input parameters to an invocation method of the program classes. The user interface 114 may be operable (e.g., via a hyperlink) by the user 102 to retrieve the one or more program classes. Using program classes to invoke the query 122 may include transmitting authorization information such as an account key and a session identifier (e.g., a globally unique identifier (GUID)) to the web service 140. Thus, program classes may enable execution of the query 122 when the system 100 is in an offline or powered-down state. Hyperlinks for program classes are further illustrated with reference to FIG. 6.

**[0021]** In a particular embodiment, the query module 120 is further configured to generate a representational state transfer (REST) URI that is useable to invoke the query 122 without execution of the query module 120. That is, the REST URI may be used to invoke the query 122 at anytime, including when the system 100 is not in operation. For example, the selected one or more input parameter options 116 may be encapsulated (e.g., in encrypted or clear-text format) as part of the REST URI. When the REST URI is invoked (e.g., at a browser), the selected one or more input parameter options 116 are transmitted to the web service as part of the REST URI. Thus, REST URIs may enable execution of the query 122 when the system 100 is in an offline or powered-down state. REST URIs are further illustrated with reference to FIG. 6.

**[0022]** The system 100 includes a results module 130 that is configured to receive results 132 of the invocation of the query 122 and to automatically display the received results

132 at the user interface 114. In a particular embodiment, the query module 120 is configured to selectively invoke the query 122 in multiple data interchange formats (e.g., RAW XML, JavaScript Object Notation (JSON), or Atom Syndication Format (ATOM 1.0)), and the results module 130 is configured to selectively format the results 132 in any  
5 of the multiple data interchange formats (e.g., RAW XML, JSON, and ATOM 1.0), in addition to a standard tabular format.

**[0023]** In a particular embodiment, a spreadsheet application at the system 100 may accept the results 132 as input data, thereby providing the user 102 with an ability to analyze or perform advanced analytics with respect to the results 132.

10 **[0024]** In operation, the interface module 110 may generate the user interface 114 for the web service 140 and the user 102 may provide user input 104 by interacting with the user interface 114. Based on the user input 104 (e.g., selections of one or more of the input parameter options 116), the user interface 114 may automatically refresh the user interface, update hyperlinks for program classes generated by the query module 120, and  
15 update the display of REST URIs generated by the query module 120.

**[0025]** The query module 120 may generate the query 122 of the web service 140 based on the user input 104 and may invoke the query 122. The query 122 may reflect user selection of one or more of the input parameter options 116. The web service 140 may retrieve results 132 of the query 122 from the database 150. The results module 130 may  
20 receive the results 132 and may display the results 132 at the user interface 114.

**[0026]** It will be appreciated that the system 100 of FIG. 1 may provide an “active” interface for web service query generation by automatically refreshing the user interface 114 to stay current with and reflect user input 104 received from the user 102. It will also be appreciated that the system 100 of FIG. 1 may simplify web service query generation.

25 For example, instead of spending time learning how to use complex application programming interfaces (APIs) of the web service 140, users that know what data they are interested in (e.g., what input parameter options 116 should be reflected by the query 122) may instead operate the user interface 114 to retrieve a dynamically created program class or a REST URI that is useable to invoke the query 122. Such dynamically created  
30 program classes and REST URIs may be used even when the system 100 is not in operation.

**[0027]** It will further be appreciated that the system 100 of FIG. 1 may provide a preview mechanism that may assist query modification. For example, the user 102 may make incremental changes to the query 122 and the system 100 may enable the user 102 to



preview effects of the incremental changes in the results 132 generated by the query 122. It will also be appreciated that web services (e.g., existing web services or newly created web services) may be made compatible with the interface generation and querying capability of the system 100 of FIG. 1 by providing series and service specifications (e.g., the specifications 112) for the web service (e.g., the web service 140). Such series and service specifications may be provided by web service developers. Alternately, series and service specifications may be automatically generated (e.g., based on source code or definition files of web services).

**[0028]** FIG. 2 is a diagram to illustrate another particular embodiment of a system 200 to generate an interface for a web service 240 and to query the web service 240. The web service 240 may be located at a web server 260 that is communicatively coupled to a database 250 and to a display device 206 (e.g., a display coupled a computing device). In an illustrative embodiment, the web service 240 is the web service 140 of FIG. 1 and the database 250 is the database 250 of FIG. 1.

**[0029]** An interface module 210 may generate a user interface 214 for the web service 240 and may transmit the user interface 214 for display at the display device 206. For example, the user interface 214 may be generated based on XML service and series specifications 212 for the web service 240. The interface module 210 may also receive user input 204 from a user 202 that is operating the user interface 214. The user input 204 may include a selection of one or more input parameter options associated with the web service 240 and that are displayed at the user interface 214. The interface module 210 may automatically generate a modified user interface 215 based on the user input 204 and may transmit the modified user interface 215 for display at the display device 206. Thus, from the perspective of the user 202, the user interface 214 may be automatically refreshed to the modified user interface 215.

**[0030]** The web server 260 also includes a query module 220. In a particular embodiment, the query module 220 generates a query of the web service 240 based on the user input 204. In another particular embodiment, the user interface 214 (and the modified user interface 215) include query generation logic that is configured to generate a query 222 based on the user input 204 and to transmit the query 222 to the web server 260. The query module 220 may invoke the query 222 with respect to the web service 240 to generate results 232, and the results 232 may be transmitted for display at the display device 206. In a particular embodiment, the results 232 are retrieved from the database 250 by the web service 240 based on the contents of the query 222.

[0031] In operation, the interface module 210 may generate the user interface 214 for the web service 240 and may transmit the user interface 214 for display at the display device 206. The user 202 may provide user input 204 by interacting with the user interface 214. Based on the user input 204, the interface module 210 may generate a modified user interface 215 and may transmit the modified user interface 215 for display at the display device 206.

[0032] The query 222 may be generated based on the user input 204 and may be invoked by the query module 220 to generate the results 232. For example, the web service 240 may retrieve the results 232 from the database 250 based on the contents of the query 222. The results 232 may be transmitted for display to the display device 206.

[0033] It will be appreciated that the system 200 of FIG. 2 may provide interface generation, interface display, and query generation at different computing devices. For example, a web service developer or vendor associated with the web service 240 and the database 250 may develop a web site at the web server 260. The web site may include the user interface 214, the modified user interface 215, and the results 232. In such an example, users such as the user 202 may be able to visit the web site to drive operation of the web service instead of learning complex APIs associated with the web service. It will thus be appreciated that the system 200 of FIG. 2 may enable software developers and data vendors to simplify web service usage.

[0034] FIG. 3 is a diagram to illustrate a particular embodiment of an XML service specification 300 that may be used with the system 100 of FIG. 1 or the system 200 of FIG. 2. For example, the XML service specification 300 may be included in the service and series specifications 112 of FIG. 1 or the service and series specifications 212 of FIG. 2.

[0035] In a particular embodiment, the XML service specification 300 may be fetched by appending a known string (e.g., “/\$service”) to the end of a base URL of a web service. For example, the service specification for locally running a United States crime statistics service “http://localhost/services/UsCrimeService.svc” may be fetched at the URL “http://localhost/services/UsCrimeService.svc/\$service.” In a particular embodiment, a service specification exists for each data set hosted by the service. For example, a service specification for a “majorcrimes” data set may be fetched at the URL “http://localhost/services/UsCrimeService.svc/majorcrimes/\$service.”

[0036] The XML service specification 300 may include a plurality of nodes. For example, the service specification may include a root node (illustratively labeled as “<service>” in

FIG. 3) that includes general nodes (e.g., a Boolean paging permission 302, a string service title 304, a string service description 306, and a logo URI operable to retrieve a logo image associated with the service).

[0037] The root node may also include parameter nodes, which include list parameters (e.g., a list parameter 308), and URI nodes. Each list parameter may represent a list of possible data values that are displayed as a drop down menu. In a particular embodiment, each list parameter includes a display name that is used as a label for the drop down menu, a unique series identifier (ID) 312, and a series source path 314. A URI node may include one or more series nodes, where each series node includes a series URI template 310 that may be used to generate a URI for a particular series specification. For example, the series source path 314 may be appended to the series URI template 310 to generate a URI for a particular series specification. For example, if the value of “services” is 4, the URI may be “http://localhost/services/UsCrimeService.svc/\$series?series=4.”

[0038] FIG. 4 is a diagram to illustrate a particular embodiment of an XML series specification 400 that may be used with the system 100 of FIG. 1 or the system 200 of FIG. 2. For example, the XML series specification 400 may be included in the service and series specifications 112 of FIG. 1 or the service and series specifications 212 of FIG. 2.

[0039] As described with respect to FIG. 3, a web service may host one or more data series. Each of the one or more data series may have an associated series specification. The URI for each particular series specification may be generated based on a series URI template (e.g., the series URI template 310 of FIG. 3) and a series source path (e.g., the series source path 314 of FIG. 3) included in a service specification (e.g., the XML service specification 300 of FIG. 3).

[0040] The series specification 400 may include a plurality of nodes. For example, the series specification may include a root node (illustratively labeled as “<series>” in FIG. 4) that includes one or more of a parameters node, a result set node, an actions node, and a URIs node. The parameters node may include parameters 402 that are defined for the particular series. For example, in the particular embodiment illustrated in FIG. 4, the parameters “State,” “City,” and “Year” are defined for the US statistics service, where each of the parameters is a list parameter (e.g., the service is capable of retrieving statistics for multiple states, cities, and years). Each such list parameter may be displayed as a drop down list at a user interface generated in accordance with the series specification 400. The parameters 402 may also include string parameters, integer parameters, and floating point parameters.

**[0041]** Each list parameter may point to a list document (e.g., “statesList,” “citiesList,” and “yearsList” in FIG. 4). Each list document may be an XML file with item nodes representing the values of the drop down menu. For example, the “statesList” document may include:

```

5      <list>
        <item key="Alabama">Alabama<item>
        <item key="Alaska">Alaska<item>
        <item key="Arizona">Arizona<item>
        <item key="Arkansas">Arkansas<item>
10     <item key="California">California<item>
        <item key="Colorado"> Colorado <item>
        <item key="Connecticut"> Connecticut <item>
        <item key="Delaware">Delaware<item>
        . . .
15     </list>

```

**[0042]** The result set node may include data fields 404 that are returned and populated by a query invocation. For example, in the particular embodiment illustrated in FIG. 4, invoking a query with respect to the US crime statistics service returns a string “Name” field and integer fields for “Violent Crime,” “Robbery,” “Aggravated Assault,” “Property Crime,” “Burglary,” “Larceny,” “MotorVehicleTheft,” and “Arson.”

**[0043]** The actions node may include actions that are performed based on selection of the parameters 402. Thus, a different action may be associated with each list value of a drop down menu at a web service interface. For example, in the particular embodiment illustrated in FIG. 4, the actions node contains an auto-refresh URI 406 that is useable to automatically refresh a user interface generated in accordance with the XML series specification 400 when a user selects one or more of the parameters 402. For example, if a user selects the state “Texas,” a drop down menu for “Cities” may be automatically refreshed in accordance with the auto-refresh URI 406 (e.g., the drop down menu for “Cities” may be changed to include cities in Texas and not include cities in any other state). Auto-refresh operations at an interface associated with a web service are further described with respect to FIG. 5.

**[0044]** The URIs node may include a query node, an analyze node, and a preview node. The query node may include a query URI 408 that is useable to query the web service for data. The analyze node may include a URI that is useable to perform advanced data analysis (e.g., at a spreadsheet application) on data generated by querying the web service. The preview node may include a results URI 410 that is useable to generate a preview of

the data generated by querying the web service. In a particular embodiment, the results URI is used to display results at a user interface as described with reference to the results 132 of FIG. 1 and the results 232 of FIG. 2.

[0045] It should be noted that although the XML service specification 300 and the XML series specification 400 are illustrated in FIGS. 3-4 as XML files, service specifications and series specifications may instead be represented in any computer-recognizable language (e.g., text file, rich text format (RTF), or hyper text markup language (HTML)).

[0046] FIG. 5 is diagram to illustrate a particular embodiment 500 of automatically refreshing an interface for a web service based on user input. In the particular embodiment illustrated in FIG. 5, the user interface 510 is refreshed to the user interface 520. For illustrative purposes, the user interfaces 510 and 520 are depicted as interfaces to the US crime statistics web service described with respect to FIG. 3 and FIG. 4.

[0047] User interfaces for web services may be refreshed in response to user input. For example, user input 502 indicating a user selection of “Texas” in the State drop down menu at the user interface 510 may trigger an auto-refresh operation. The result of the auto-refresh operation may be the user interface 520, where bounds of the City and Year drop down menus are automatically updated 522 based on series specification(s) associated with Texas. For example, in the particular embodiment illustrated in FIG. 5, the initial City drop down reflects an unbounded set of available cities in the USA, and the initial Year drop down menu reflects an unbounded set of all available years (e.g., 1990-2000). In response to the user input 502, the City drop down menu is automatically set to “Abilene” (e.g., the first city in Texas, in alphabetical order, for which the US crime statistics service is able to retrieve data) and the Year drop down menu is automatically set to 1995 (e.g., the earliest year for which the US crime statistics service is able to retrieve data for Abilene, Texas). The bounds of City drop down menu are set (e.g., available cities in Texas), and minimum/maximum values of the Year drop down menu are also set (e.g., to the earliest and latest years for which data related to the state of Texas is available).

[0048] It will be appreciated that automatically refreshing user interfaces for web services as illustrated in FIG. 5 may provide an “active” interface for web service query generation. For example, user interfaces may be automatically refreshed to stay current with and reflect user input (e.g., the user input 502) received from a user.

[0049] FIG. 6 is a diagram to illustrate a particular embodiment of an interface 600 for a web service, where the interface may preview results of a query of the web service. The

interface 600 includes a query generation area 610, a results preview area 620, and a web service information area 630. In the particular embodiment illustrated in FIG. 6, the interface 600 is an interface to data provided by the Food and Agricultural Organization (FAO) of the United Nations (UN).

5 [0050] The web service information area 630 may include a title, description, and logo of the web service. In a particular embodiment, the title, description and logo are retrieved from a service specification for the web service as described with respect to the XML service specification 300 of FIG. 3.

[0051] The query generation area 610 may also include various input parameter options and query generation options in accordance with service and series specifications. For 10 example, the query generation area 610 may include a display of service parameters 611. For example, the service parameters 611 include “Series,” “Country or Area,” and “Year.” The particular embodiment illustrated in FIG. 6 indicates a user selection of the Agricultural Production Index (“Agri. Prodn. Indx.”) series. The query generation area 610 15 may also include a display of paging information (e.g., whether the web service supports paging). The query generation area 610 may further include a display of an Account Key 613, a SessionID 614 (e.g., a GUID), a REST URI 617, and a program class (e.g., a proxy C# class) download hyperlink 616. The query generation area 610 may also include an option to invoke a query in multiple available data interchange formats 622. For example, 20 in the particular embodiment illustrated in FIG. 6, the query of the web service is invoked as an ATOM 1.0 feed. The query of the web service may alternately be invoked in any other data interchange format, such as RAW XML or JSON.

[0052] The results preview area 620 may include a display of the results generated by invoking a query of the web service. For example, in the particular embodiment 25 illustrated in FIG. 6, the results preview area 620 displays data from the Agricultural Production Index. The results preview area 620 may display the results in multiple available data interchange formats 622. For example, although the particular embodiment illustrated in FIG. 6 depicts the results displayed in a tabular format, the results may alternately be displayed at the results preview area 620 in a RAW XML format, a JSON 30 format, or an ATOM 1.0 format. When user input changes the value of the service parameters 611, the results preview area 620 may be updated to display a preview of new results that are based on the changed service parameters 611.

[0053] FIG. 7 is a flow diagram to illustrate a particular embodiment of a method 700 to generate an interface for a web service and to query the web service. In an illustrative embodiment, the method 700 may be performed by the system 100 of FIG. 1.

[0054] The method 700 includes generating a user interface associated with a web service, at 702. The user interface includes one or more input parameter options associated with the web service and is generated based on an XML service specification and one or more XML series specifications associated with the XML service specification. For example, in FIG. 1, the interface module 110 may generate the user interface 114 based on the service and series specifications 112 associated with the web service 140, where the user interface 114 includes the input parameter options 116. As another example, as illustrated in FIG. 6, the interface 600 may be generated, where the interface 600 includes the service parameters 611.

[0055] The method 700 also includes receiving user input via the user interface, where the user input includes a selection of at least one of the one or more input parameter options, at 704. For example, in FIG. 1, the user input 104 may be received via the user interface 114, where the user input 104 includes a selection of the input parameter options 116. As another example, as illustrated in FIG. 6, the Agricultural Production Index may be selected.

[0056] The method 700 further includes generating a query of the web service based on the received user input, where the query reflects the at least one selected input parameter option, at 706. For example, in FIG. 1, the query module 120 may generate the query 122. As another example, as illustrated in FIG. 6, a query of the Agricultural Production Index data series (e.g., a structured query language (SQL) query “select \* from AgriProdIndx”) may be generated.

[0057] The method 700 includes invoking the generated query with respect to the web service to receive results, at 708. For example, in FIG. 1, the query module 120 may invoke the query 122 and the results module 130 may receive the results 132. As another example, as illustrated in FIG. 6, the query of the Agricultural Production Index may be invoked and the results displayed in the results preview area 620 may be received.

[0058] The method 700 also includes automatically adding the received results to the user interface, at 710. For example, in FIG. 1, the results module 130 may automatically add the results 132 to the user interface 114. As another example, as illustrated in FIG. 6, the results may be displayed in the results preview area 620 of the interface 600.

[0059] FIG. 8 is a flow diagram to illustrate a particular embodiment of a method 800 of receiving an interface for a web service and generating a query of the web service. In an illustrative embodiment, the method 800 may be performed at the system 200 of FIG. 2.

[0060] The method 800 includes receiving a user interface associated with a web service from a web server, at 802, and displaying the user interface, at 804. The user interface includes one or more input parameter options associated with the web service and is generated based on an XML service specification and one or more XML series specifications associated with the XML service specification. For example, in FIG. 2, the user interface 214 associated with the web service 240 may be received from the server 260 and displayed at the display device 206, where the user interface 214 is generated based on the series and service specifications 212. As another example, as illustrated in FIG. 6, the interface 600 may be displayed, where the interface 600 is generated based on XML service and series specifications.

[0061] The method 800 also includes receiving user input at the user interface, at 806, and transmitting the user input to the web server, at 808. The user input includes a selection of at least one of the one or more input parameter options. For example, in FIG. 2, the user input 204 may be received and transmitted to the web server 260. As another example, as illustrated in FIG. 6, the Agricultural Production Index may be selected.

[0062] The method 800 further includes receiving a modified user interface from the web server and displaying the modified user interface, at 810. The modified user interface reflects the user input. For example, in FIG. 2, the modified user interface 215 may be received from the web server 260 and displayed at the display device 206. As another example, as illustrated in FIG. 6, the selection of the Agricultural Production Index may be indicated by the “Series” drop down menu at the interface 600.

[0063] The method 800 includes generating a query of the web service based on the received user input, where the query reflects the at least one selected input parameter option, at 812. For example, in FIG. 2, the query 222 may be generated. As another example, as illustrated in FIG. 6, a query of the Agricultural Production Index data series may be generated.

[0064] The method 800 also includes transmitting the generated query to the web server and receiving results based on an invocation of the generated query from the web server, at 814. For example, in FIG. 2, the query 222 may be transmitted to the web server 260 and the results 232 based on an invocation of the query 222 may be transmitted from the web server 260. As another example, as illustrated in FIG. 6, the query of the Agricultural



Production Index may be invoked and the results displayed in the results preview area 620 may be received.

[0065] The method further includes displaying the results at the modified user interface, at 816. For example, in FIG. 2, the results 232 may be displayed at the modified user interface 215 at the display device 206. As another example, as illustrated in FIG. 6, the results may be displayed in the results preview area 620 of the interface 600.

[0066] FIG. 9 depicts a block diagram of a computing environment 900 including a computing device 910 operable to support embodiments of computer-implemented methods, computer program products, and system components according to the present disclosure. In an illustrative embodiment, the computing device 910 may include one or more of the modules 110, 120, 130 of FIG. 1, the web service 140 of FIG. 1, and the display device 206 of FIG. 2. Each of the modules 110, 120, 130 of FIG. 1, the web service 140 of FIG. 1, and the display device 206 of FIG. 2 may include or be implemented using the computing device 910 or a portion thereof.

[0067] The computing device 910 includes at least one processor 920 and a system memory 930. Depending on the configuration and type of computing device, the system memory 930 may be volatile (such as random access memory or “RAM”), non-volatile (such as read-only memory or “ROM,” flash memory, and similar memory devices that maintain stored data even when power is not provided), or some combination of the two. The system memory 930 typically includes an operating system 932, one or more application platforms 934, one or more applications 936, and may include program data 938 associated with the one or more applications.

[0068] The computing device 910 may also have additional features or functionality. For example, the computing device 910 may also include removable and/or non-removable additional data storage devices such as magnetic disks, optical disks, tape, and standard-sized or miniature flash memory cards. Such additional storage is illustrated in FIG. 9 by removable storage 940 and non-removable storage 950. Computer storage media may include volatile and/or non-volatile storage and removable and/or non-removable media implemented in any technology for storage of information such as computer-readable instructions, data structures, program components or other data. The system memory 930, the removable storage 940 and the non-removable storage 950 are all examples of computer storage media. The computer storage media includes, but is not limited to, RAM, ROM, electrically erasable programmable read-only memory (EEPROM), flash memory or other memory technology, compact disks (CD), digital versatile disks (DVD)

or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to store information and that can be accessed by the computing device 910. Any such computer storage media may be part of the computing device 910. The computing device 910 may also have input device(s) 960, such as a keyboard, mouse, pen, voice input device, touch input device, etc. Output device(s) 970, such as a display, speakers, printer, etc. may also be included. A user 902 may be able to interact with the computing device 910 via the input device(s) 960 and the output device(s) 970.

[0069] The computing device 910 also contains one or more communication connections 980 that allow the computing device 910 to communicate with other computing devices (e.g., a web server 990 and a database 992) over a wired or a wireless network. In an illustrative embodiment, the web server 990 is the web server 260 of FIG. 1 and the database 992 is the database 150 of FIG. 1 or the database 250 of FIG. 2.

[0070] It will be appreciated that not all of the components or devices illustrated in FIG. 9 or otherwise described in the previous paragraphs are necessary to support embodiments as herein described. For example, the removable storage 940 may be optional.

[0071] The illustrations of the embodiments described herein are intended to provide a general understanding of the structure of the various embodiments. The illustrations are not intended to serve as a complete description of all of the elements and features of apparatus and systems that utilize the structures or methods described herein. Many other embodiments may be apparent to those of skill in the art upon reviewing the disclosure. Other embodiments may be utilized and derived from the disclosure, such that structural and logical substitutions and changes may be made without departing from the scope of the disclosure. Accordingly, the disclosure and the figures are to be regarded as illustrative rather than restrictive.

[0072] Those of skill would further appreciate that the various illustrative logical blocks, configurations, modules, and process steps or instructions described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. Various illustrative components, blocks, configurations, modules, or steps have been described generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for

each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present disclosure.

[0073] The steps of a method described in connection with the embodiments disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in computer readable media, such as random access memory (RAM), flash memory, read only memory (ROM), registers, a hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary storage medium is coupled to a processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor or the processor and the storage medium may reside as discrete components in a computing device or computer system.

[0074] Although specific embodiments have been illustrated and described herein, it should be appreciated that any subsequent arrangement designed to achieve the same or similar purpose may be substituted for the specific embodiments shown. This disclosure is intended to cover any and all subsequent adaptations or variations of various embodiments.

[0075] The Abstract of the Disclosure is provided with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. In addition, in the foregoing Detailed Description, various features may be grouped together or described in a single embodiment for the purpose of streamlining the disclosure. This disclosure is not to be interpreted as reflecting an intention that the claimed embodiments require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter may be directed to less than all of the features of any of the disclosed embodiments.

[0076] The previous description of the embodiments is provided to enable a person skilled in the art to make or use the embodiments. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined 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 novel features as defined by the following claims.

## CLAIMS

1. A system, comprising:  
a memory; and  
a processor coupled to the memory, wherein the processor is configured to retrieve instructions from the memory that cause execution of:

5 an interface module configured to:  
generate a user interface associated with a web service, wherein the user interface includes one or more input parameter options associated with the web service;  
receive user input via the user interface, wherein the user input  
10 comprises a selection of at least one of the one or more input parameter options; and  
automatically refresh the one or more input parameter options at the user interface based on the received user input;  
a query module configured to:  
15 generate a query of the web service based on the received user input, wherein the query reflects the at least one selected input parameter option; and  
invoke the generated query with respect to the web service to receive results; and  
20 a results module configured to automatically display the received results at the user interface.

2. The system of claim 1, wherein the user interface is generated based on a service specification and one or more series specifications associated with the service specification.

25 3. The system of claim 2, wherein the service specification and the one or more series specifications are represented in extensible markup language (XML).

4. The system of claim 2, wherein the service specification includes a service paging permission, a service title, a service description, or any combination thereof.

5. The system of claim 2, wherein the service specification includes one or more list  
30 parameters and a series uniform resource identifier (URI) template, wherein the one or more list parameters are displayed as a drop down list at the user interface, and wherein each of the one or more list parameters includes a series identifier and a series source path.

6. The system of claim 5, wherein the series source path is useable to generate a URI associated with a particular series specification of the one or more series specifications by appending the series source path to the series URI template.

7. The system of claim 2, wherein each of the one or more series specifications  
5 includes a string parameter that is used in generating the query, an integer parameter that is used in generating the query, a floating point parameter that is used in generating the query, or any combination thereof.

8. The system of claim 1, wherein the query module is further configured to  
10 automatically generate one or more program classes useable to invoke the query, wherein the user interface is operable to retrieve the one or more program classes, and wherein using the one or more program classes to invoke the query includes transmitting an account key and a session identifier to the web service.

9. The system of claim 8, wherein the session identifier is a globally unique identifier (GUID).

15 10. The system of claim 1, wherein the query module is further configured to generate a representational state transfer (REST) uniform resource identifier (URI), wherein the REST URI is useable to invoke the query without execution of the query module.

11. The system of claim 1, wherein the query module is further configured to  
20 selectively invoke the query in a plurality of data interchange formats; and wherein the results module is further configured to selectively display the results in any of the plurality of data interchange formats.

12. The system of claim 1, further comprising:

a database, wherein the web service is configured to retrieve the results from the database in response to the query; and

25 a spreadsheet application configured to accept results from the results module as input data.

13. The system of claim 1, wherein automatically refreshing the one or more input  
parameter options based on the received user input includes refreshing one or more drop down menus at the user interface, updating one or more input parameter bounds at the user  
30 interface, or any combination thereof.

14. A method, comprising:

at a computer system, generating a user interface associated with a web service, wherein the user interface includes one or more input parameter options associated with the web service and wherein the user interface is generated based on an extensible markup

language (XML) service specification and one or more XML series specifications associated with the XML service specification;

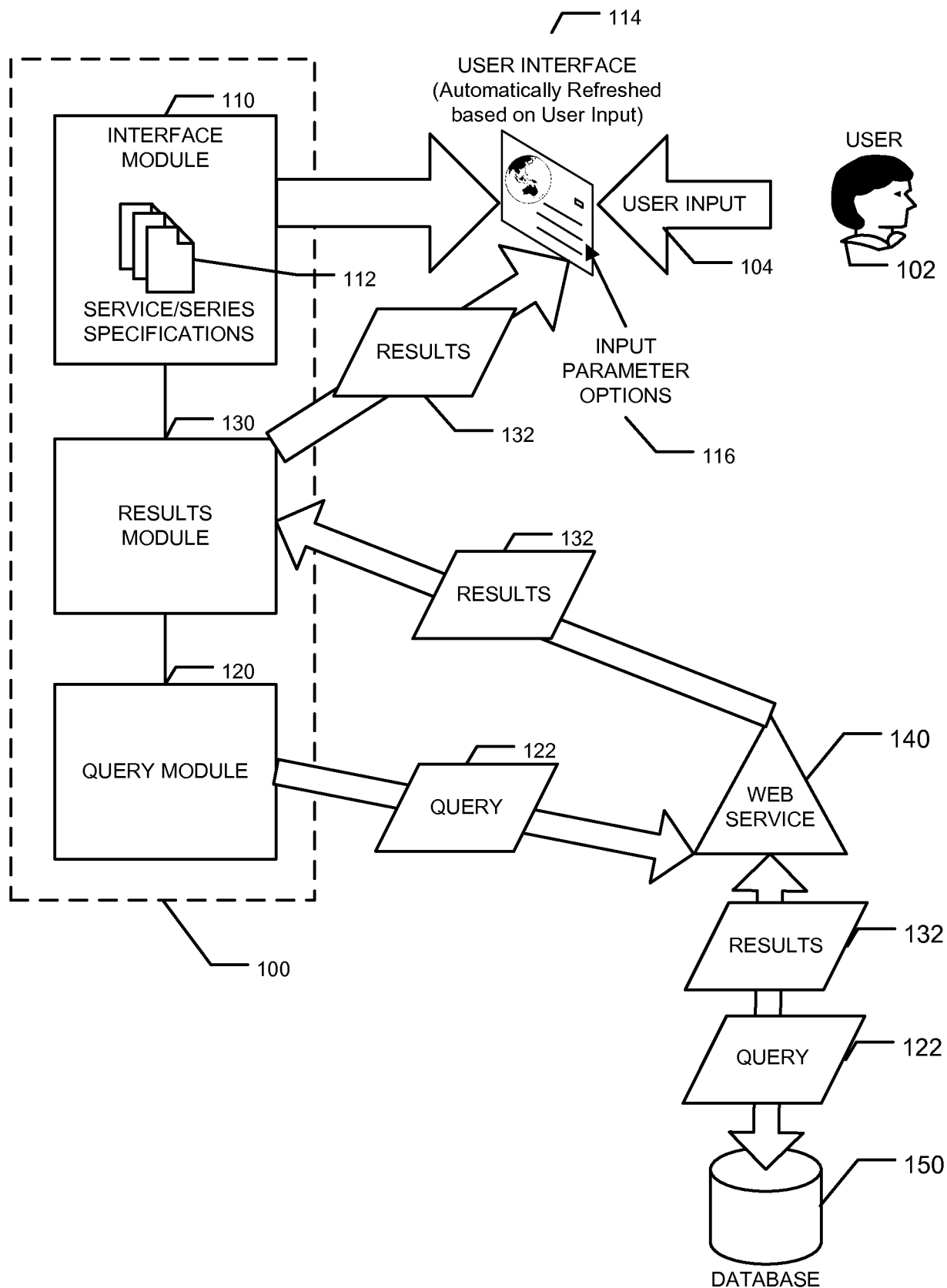
receiving user input via the user interface, wherein the user input comprises a selection of at least one of the one or more input parameter options;

5 generating a query of the web service based on the received user input, wherein the query reflects the at least one selected input parameter option;

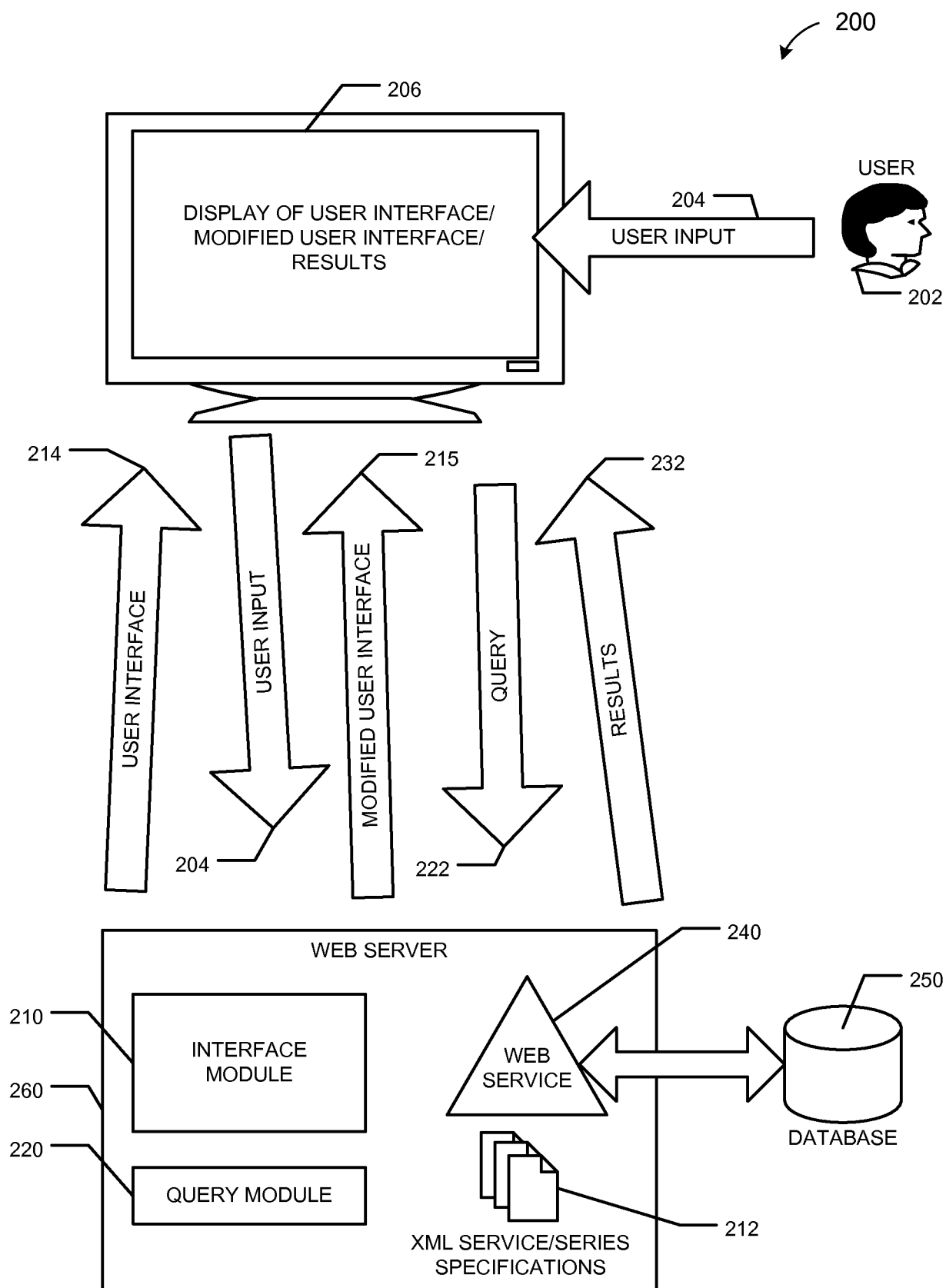
invoking the generated query with respect to the web service to receive results; and automatically adding the received results to the user interface.

15. The method of claim 14, wherein invoking the query includes performing one or  
10 more database operations at a database and wherein the results include one or more data items from the database.

1/9

**FIG. 1**

2/9

**FIG. 2**



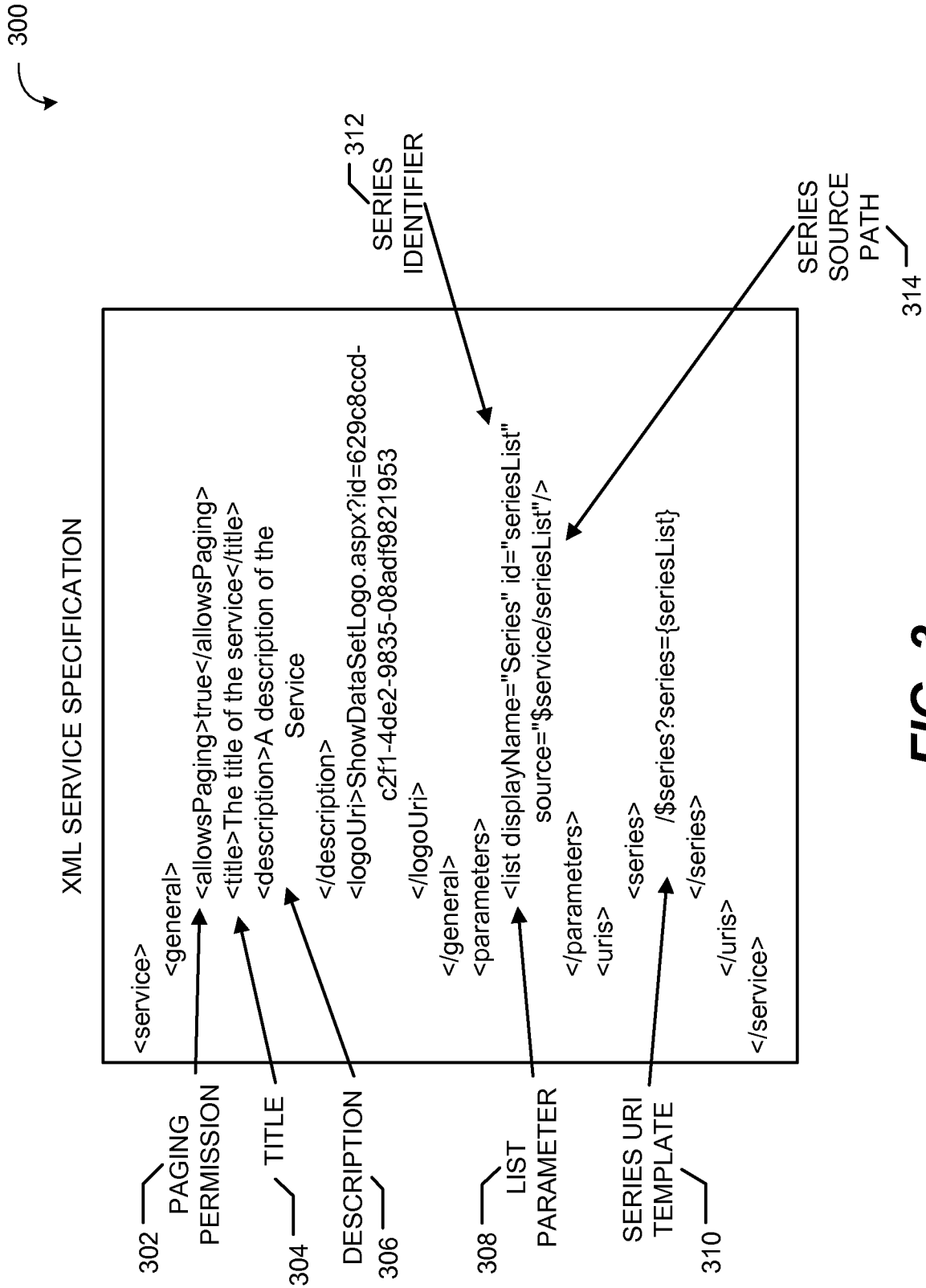


FIG. 3

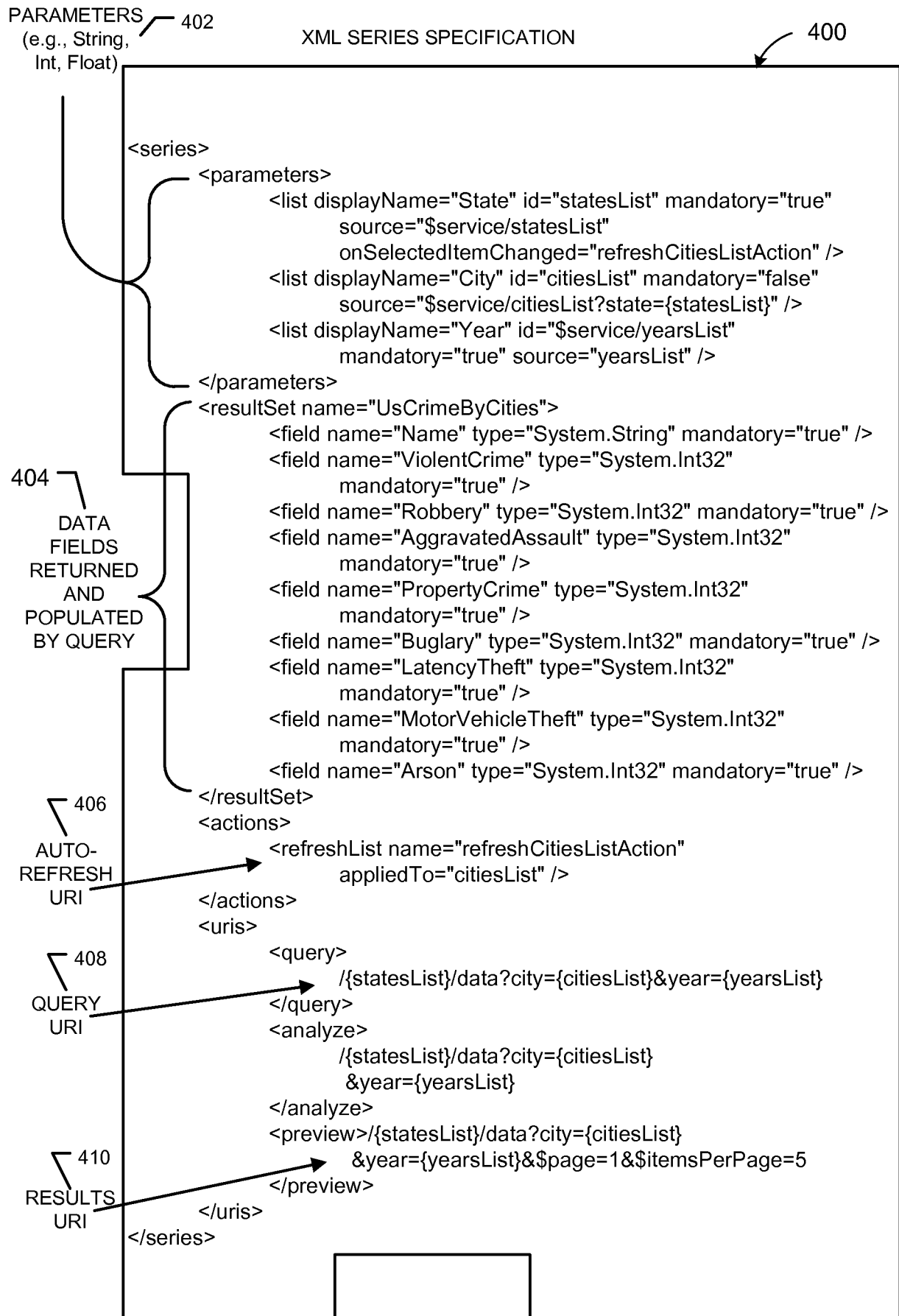


FIG. 4

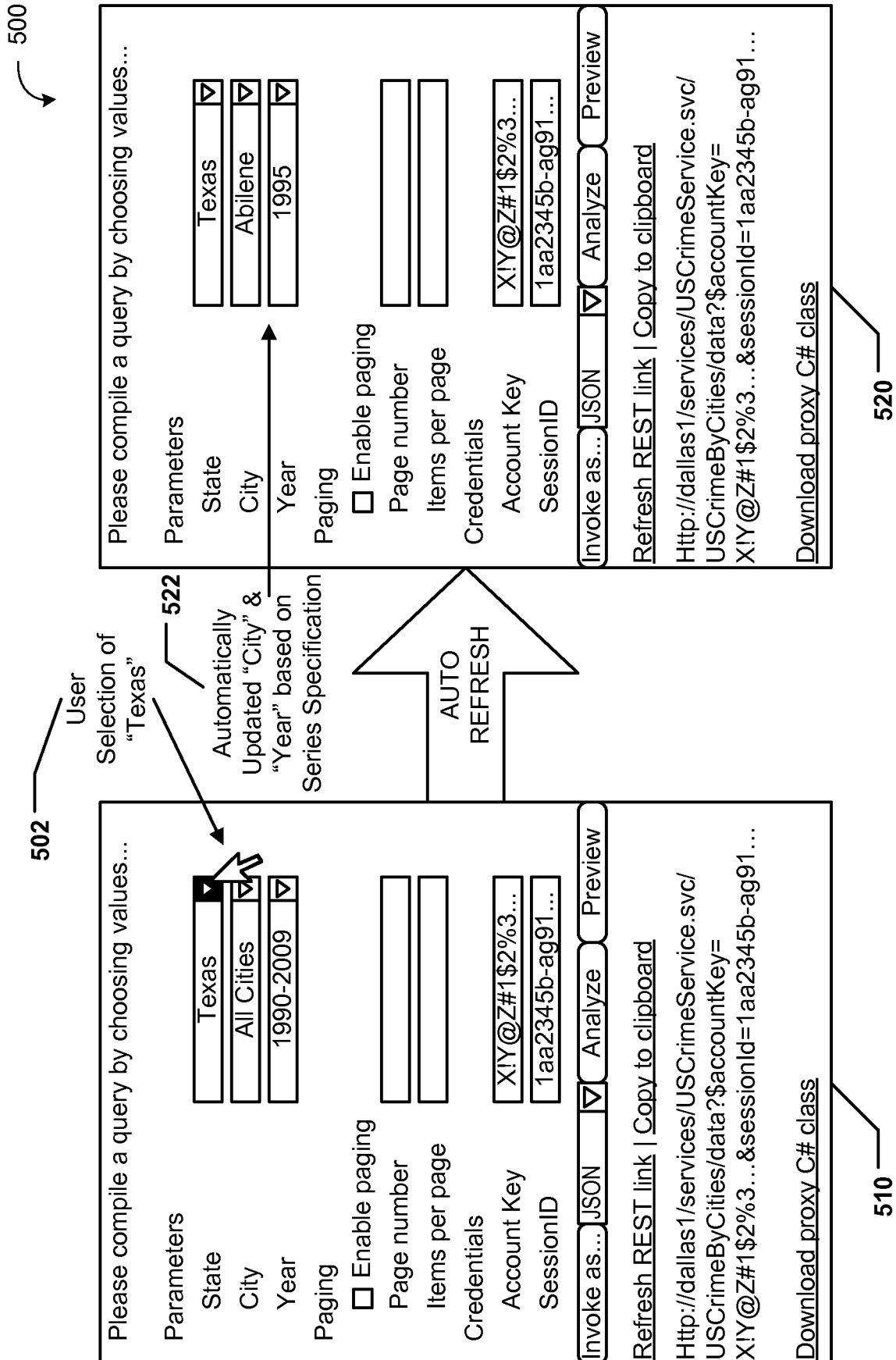



FIG. 5

600



FAO DATA – Food And Agriculture Organization  
The Food and Agriculture Organization provides UN Data with two data...

Please compile a query by choosing values...

Service Parameters

Series

Country or Area

Year

Paging

☐ Enable paging

Page number

Items per page

Credentials

Account Key

SessionID

Invoke as...

[Refresh REST link](#) | [Copy to clipboard](#)

Http://dallas1/services/UnService.svc/FAO(3510)/data?\$accountKey=XIY@Z#1\$2%3...&sessionId=1aa2345b-ag91...

[Download proxy C# class](#)

REST URI

PROGRAM CLASS  
DOWNLOAD

Preview Format : **Table** | [XML](#) | [JSON](#) | [ATOM 1.0](#)

seriesCode	countryOr AreaCode	countryOr Area	Year	Value
3510	4	Afghanistan	1989	71.4
3510	4	Afghanistan	1988	72.4
3510	4	Afghanistan	1987	73.9
3510	4	Afghanistan	1986	70.6
3510	4	Afghanistan	1986	78.4

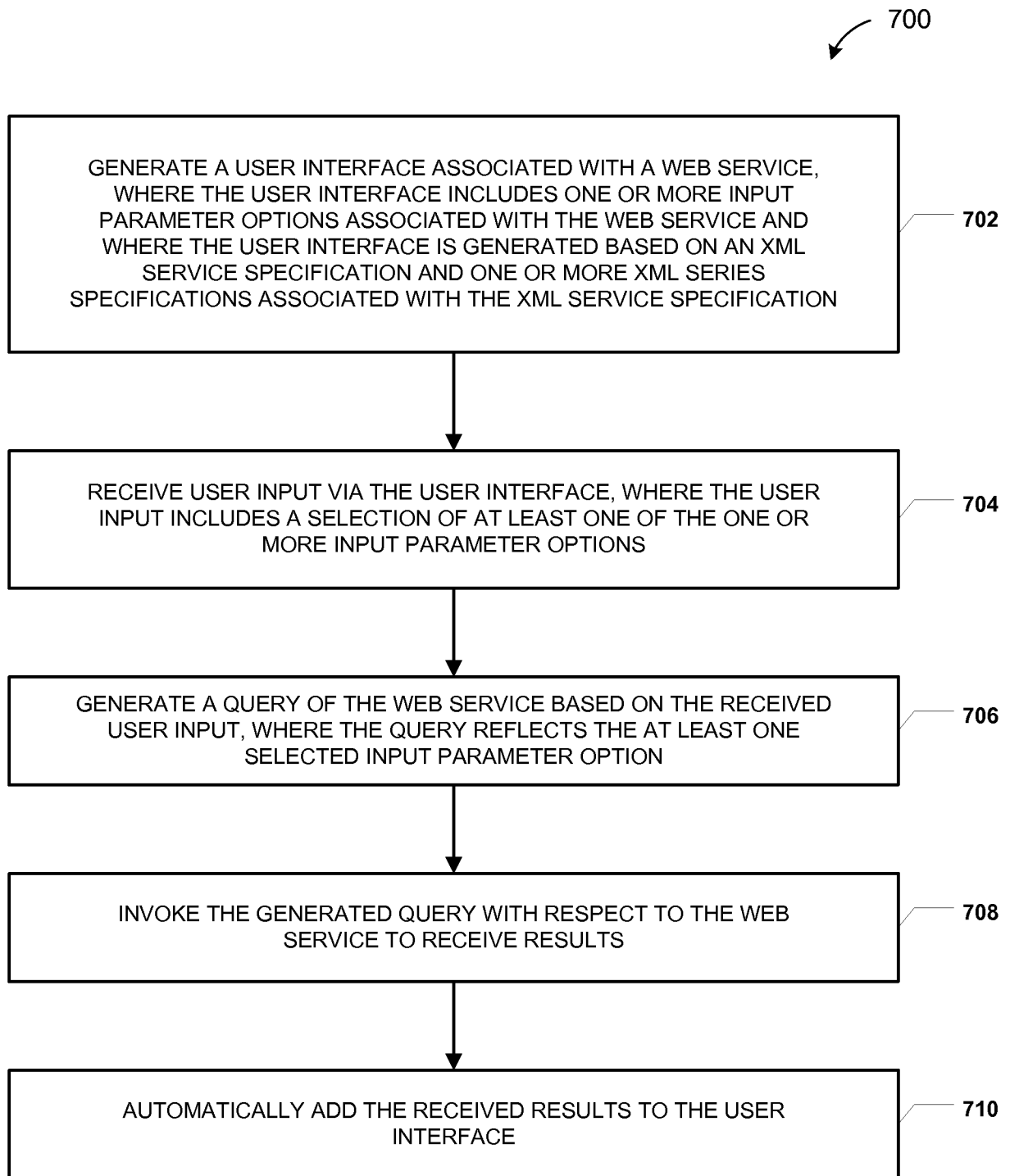
AVAILABLE DATA  
INTERCHANGE  
FORMATS

620

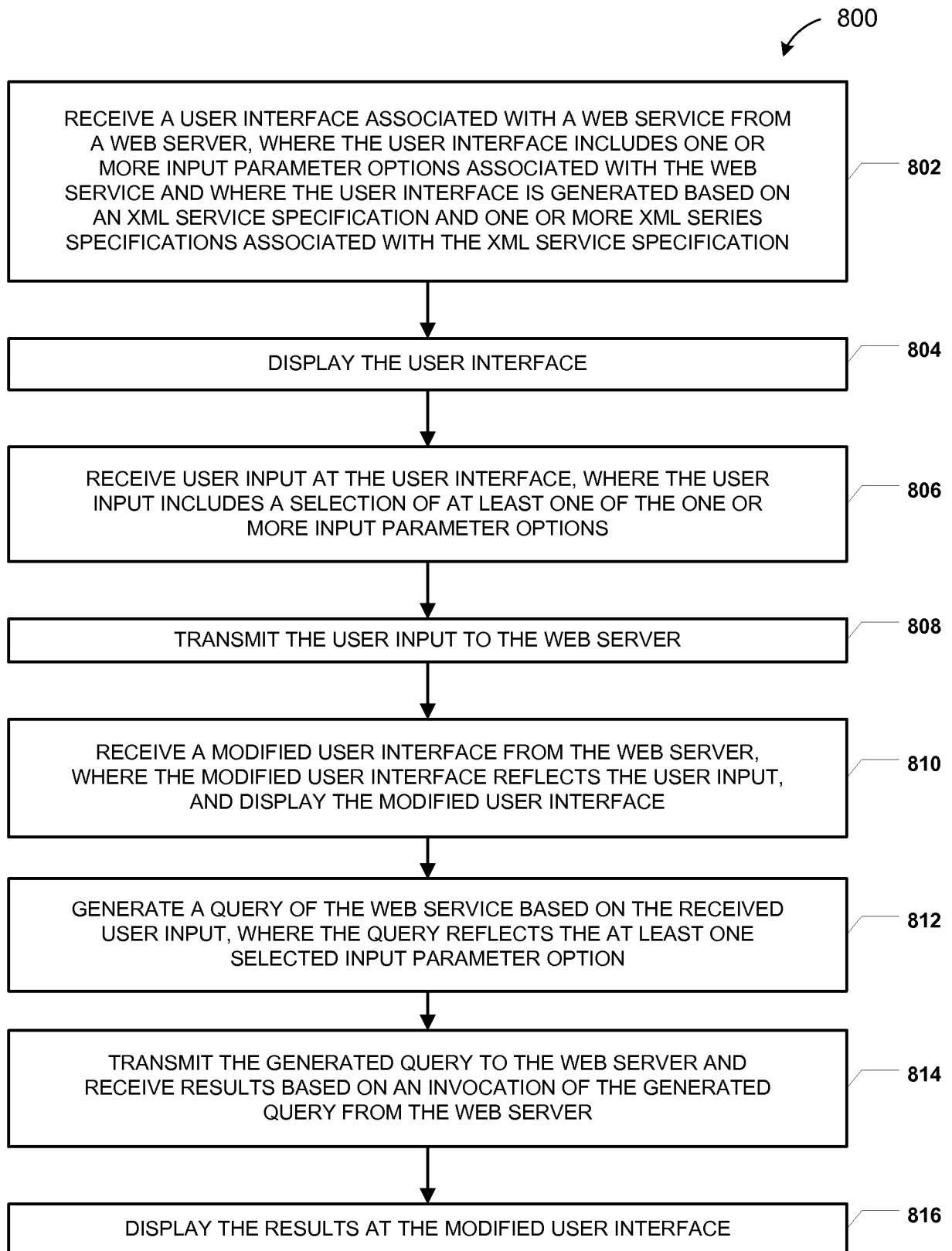
622

FIG. 6

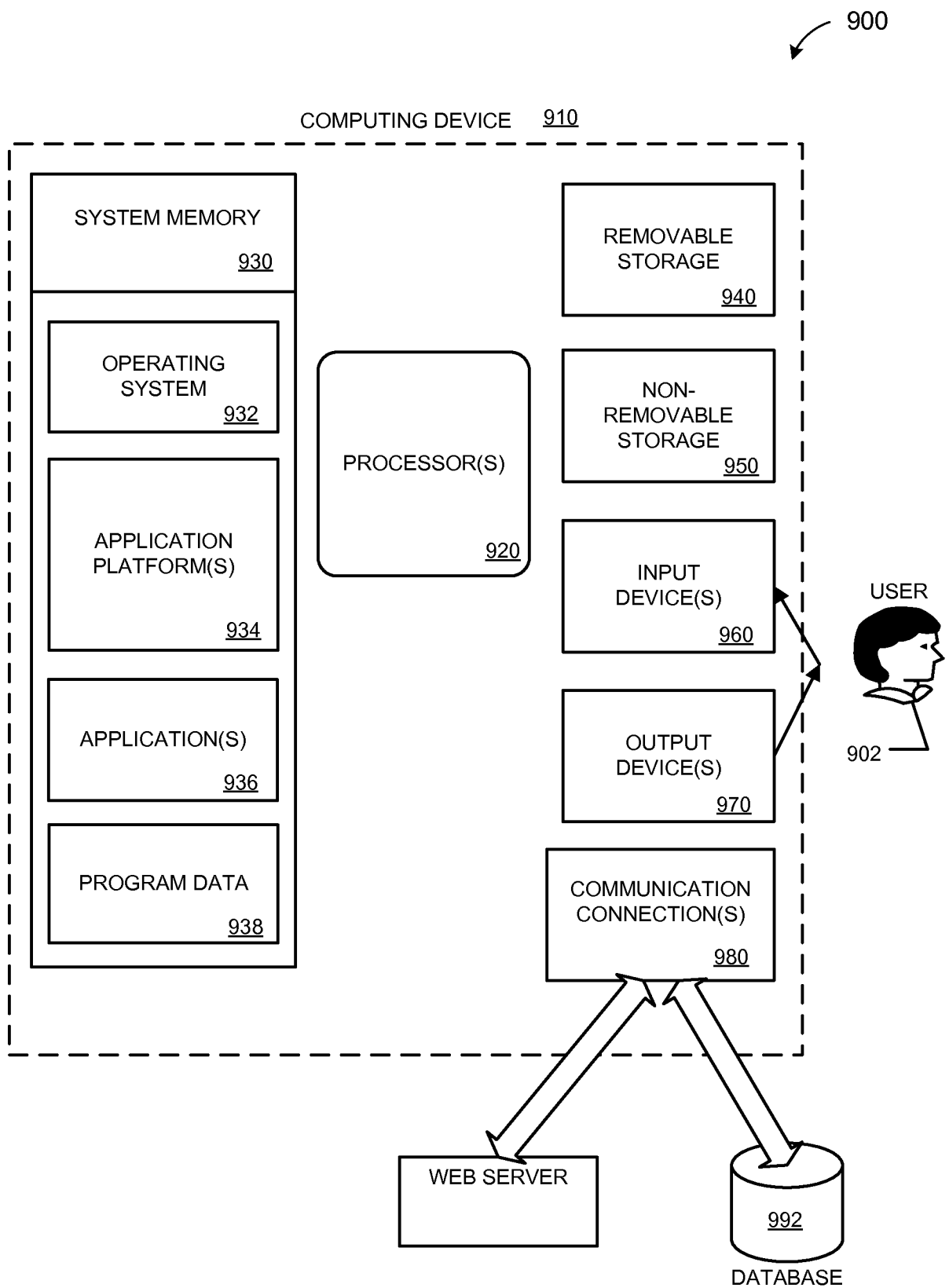
7/9

**FIG. 7**

8/9

**FIG. 8**

9/9

**FIG. 9**