

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
25 June 2009 (25.06.2009)

PCT

(10) International Publication Number  
WO 2009/078568 A1

- (51) International Patent Classification:  
G06F 17/00 (2006.01)
- (21) International Application Number:  
PCT/KR2008/006031
- (22) International Filing Date: 14 October 2008 (14.10.2008)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
10-2007-0131081  
14 December 2007 (14.12.2007) KR
- (71) Applicant (for all designated States except US): SAM-  
SUNG ELECTRONICS CO., LTD. [KR/KR]; 416, Mae-  
tan-dong, Yeongtong-gu, Suwon-si, Gyeonggi-do 442-742  
(KR).
- (72) Inventors: JUNG, Dong-Shin; 106-805 Seongji Apt.,  
Gwonseon-dong, Gwonseon-gu, Suwon-si, Gyeonggi-do  
441-718 (KR). LEE, Joo-Yeol; 5-1106 Shindong-a Apt.,  
Daerim-dong, Yeongdeungpo-gu, Seoul 150-070 (KR).  
LAWRENCE, Joe, Antony; Samsung India Software

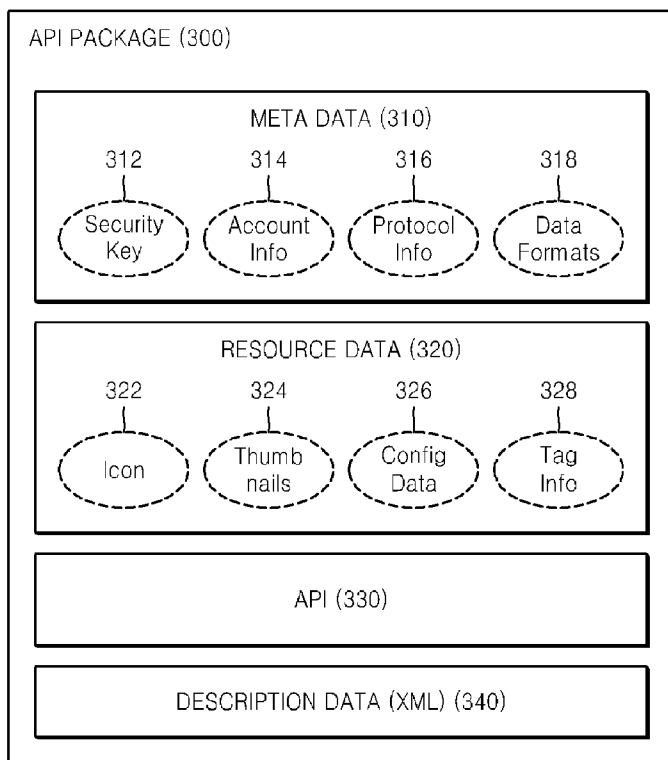
Operations, Pvt. Ltd., Bagmane Lakeview, Block 'B', No. 66/1, Bagmane Tech Park, C V Raman Nagar, Byrasandra, Bangalore, Karnataka 560093 (IN). MALAPATI, Raghavendra; Samsung India Software Operations, Pvt. Ltd., Bagmane Lakeview, Block 'B', No. 66/1, Bagmane Tech Park, C V Raman Nagar, Byrasandra, Bangalore, Karnataka 560093 (IN). K., Subramanian; Samsung India Software Operations, Pvt. Ltd., Bagmane Lakeview, Block 'B', No. 66/1, Bagmane Tech Park, C V Raman Nagar, Byrasandra, Bangalore, Karnataka 560093 (IN). SASIDHARAN, Vinoth; Samsung India Software Operations, Pvt. Ltd., Bagmane Lakeview, Block 'B', No. 66/1, Bagmane Tech Park, C V Raman Nagar, Byrasandra, Bangalore, Karnataka 560093 (IN).

- (74) Agent: Y.PLEE, MOCK & PARTNERS; The Ko-  
ryo Building, 1575-1, Seocho-dong, Seocho-gu, Seoul  
137-875 (KR).
- (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,

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR PROVIDING API SERVICE AND MAKING API MASH-UP, AND COMPUTER READABLE RECORDING MEDIUM THEREOF

FIG. 3



(57) Abstract: The present invention provides an open application program interface (API) service. A method of providing the API service includes generating meta-data for executing an API, generating resource data for generating a mash-up of the API, generating description data corresponding to the API, the meta-data, and the resource data, and generating an API package comprising the API, the meta-data, the resource data, and the description data. Accordingly, mash-up contents can be easily generated from various types of APIs.

WO 2009/078568 A1



CH, 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, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, 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, 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, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, 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:**

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

## Description

# METHOD AND APPARATUS FOR PROVIDING API SERVICE AND MAKING API MASH-UP, AND COMPUTER READABLE RECORDING MEDIUM THEREOF

### Technical Field

- [1] Method and apparatus consistent with the present invention relate to an open application program interface (API) service, and more particularly, to a method and apparatus for providing an API service and generating an API mash-up, and a computer-readable recording medium having embodied thereon a program for executing the method.

### Background Art

- [2] A mash-up service is a technology producing a new API by putting two or more APIs together in a web. The mash-up service has advantages in that efforts for realizing a new service are reduced by using an established opened API and in that utility of the established opened API can be maximized.
- [3] FIG. 1 is a view of an example of a mash-up service based on an open API.
- [4] FIG. 1 illustrates a mash-up 110 in which an imaging service known as Flickr (an online photo management and sharing application), and a map service from Google are combined together. When a user clicks a specific location on a map, the mash-up 110 displays images related to the location on the map. A hypertext markup language (HTML) code 120 of the mash-up 110 is illustrated on the right side of FIG. 1.
- [5] FIG. 2 is a flowchart of a method of producing the mash-up of FIG. 1.
- [6] Referring to FIG. 2, firstly, a mash-up developer plans to produce a particular kind of mash-up (Operation 202). Then, the mash-up developer searches and selects open APIs (Operation 210) and (Operation 216) that should be used to produce the mash-up (Operation 204). If the mash-up developer decides to produce a mash-up service by using a Google map open API and a Flickr open API, the mash-up developer analyzes the Google map open API and the Flickr open API and grasps characteristics (for example, communication protocol, data format, and input/output data format) of the open API services (Operation 206). An open API service provider generates user account information (Operation 212) and (Operation 218) or a user certification key (Operation 214) and provides them to the mash-up developer.
- [7] The mash-up developer secures the open APIs, embodies a mash-up function with reference to manuals provided by open API service providers, and determines a layout of a HTML page to complete final mash-up contents (Operation 208). Accordingly, the mash-up developer should take charge of works related to an embodiment of a mash-

up such as a communication protocol and data format conversion. In particular, since a method of providing open API services varies according to the open API service providers, firstly the mash-up developer should analyze various open API services and learn the related technologies, and the mash-up depends on the mash-up developer's ability.

- [8] Also, since codes or scripts related to the open APIs consisting of the mash-up are made by the mash-up developer at his or her discretion, it is difficult to renew the open APIs consisting of the mash-up or to add a new open API to the established mash-up.

## **Disclosure of Invention**

### **Technical Solution**

- [9] Exemplary embodiments of the present invention overcome the above disadvantages and other disadvantages not described above. Also, the present invention is not required to overcome the disadvantages described above, and an exemplary embodiment of the present invention may not overcome any of the problems described above.

- [10] The present invention provides a method and apparatus for providing an application program interface (API) service and producing an API mash-up, and a computer-readable recording medium used to easily produce mash-up contents from various types of APIs.

- [11] The present invention also provides a method and apparatus for providing an API service and producing an API mash-up, and a computer-readable recording medium used to easily update mash-up contents comprising various types of APIs.

### **Advantageous Effects**

- [12] According to the present invention, a standardized format of an API package including an API, meta-data, resource data, and description data is provided and a mash-up is generated using the API package, so that mash-up contents can be easily generated from various types of APIs and can be easily updated.

### **Description of Drawings**

- [13] The above and other features and aspects of the present invention will become more apparent by describing in detail exemplary embodiments thereof with reference to the attached drawings in which:

- [14] FIG. 1 is a view of an example of a mash-up service based on an open application program interface (API);

- [15] FIG. 2 is a flowchart of a method of producing the mash-up of FIG. 1;

- [16] FIG. 3 is a view of an example of an API package according to an exemplary embodiment of the present invention;

- [17] FIG. 4 is a block diagram illustrating an apparatus for providing an API service

according to an exemplary embodiment of the present invention;

[18] FIG. 5 is a block diagram illustrating an apparatus for producing an API mash-up according to an exemplary embodiment of the present invention;

[19] FIG. 6 is a view illustrating an apparatus for producing an API mash-up according to another exemplary embodiment of the present invention;

[20] FIG. 7 is a flowchart of a method of providing an API service according to an exemplary embodiment of the present invention;

[21] FIG. 8 is a flowchart of a method of producing an API mash-up according to an exemplary embodiment of the present invention; and

[22] FIG. 9 is a flowchart of an example of a process for producing the mash-up of API packages of FIG. 8.

### **Best Mode**

[23] According to an aspect of the present invention, there is provided a method of providing an application program interface (API) service, the method including: generating meta-data for executing an API; generating resource data for generating a mash-up of the API; generating description data corresponding to the API, the meta-data, and the resource data; and generating an API package comprising the API, the meta-data, the resource data, and the description data.

[24] The API may be an open API or a local API.

[25] The method may further include transmitting the API package to a package providing server or an apparatus for producing the API mash-up.

[26] The meta-data may include information about a communication protocol to communicate with the package providing server or the apparatus for producing the API mash-up, and information about a data format of the API.

[27] The meta-data may further include user account information or a user certification key for using the API.

[28] The resource data may include a configuration data of a user interface to generate the mash-up and a tagging information representing an input/output data format of the API.

[29] The resource data may further include information about an icon or a thumb nail for executing the mash-up of the API in the user interface.

[30] The description data may include information about one of components of the API package, a version of the API, and a library function for executing the API.

[31] According to another aspect of the present invention, there is provided a method of generating an API mash-up, the method including: obtaining at least two API packages which respectively includes an API, meta-data for executing the API, resource data for generating a mash-up of the API, and description data corresponding to the API, the meta-data, and the resource data; extracting the API, the meta-data, and the resource

data from each of the at least two API packages by using the description data; and generating a mash-up of the at least two API packages by using the API, the meta-data, and the resource data.

- [32] The generating the mash-up of the at least two API packages may include: analyzing an input/output data format of the API; searching another API package to perform the mash-up with the at least two API packages selectively on the basis of the input/output data format of the API; and generating the mash-up by using the at least two API packages and another API package.
- [33] The method may further include replacing one of the at least two API packages in the mash-up to another API package.
- [34] The method may further include adding another API package to the mash-up.
- [35] The method may further include removing a part of the at least two API packages from the mash-up.
- [36] According to another aspect of the present invention, there is provided an apparatus for providing an API service, the apparatus including: a meta-data generating unit which generates meta-data for executing an API; a resource data generating unit which generates resource data for generating a mash-up of the API; a description data generating unit which generates description data corresponding to the API, the meta-data, and the resource data; and an API package generating unit which generates an API package comprising the API, the meta-data, the resource data, and the description data.
- [37] According to another aspect of the present invention, there is provided an apparatus for producing an API mash-up, the apparatus including: an API package obtaining unit which obtains at least two API packages which respectively comprise an API, meta-data for executing the API, resource data for generating a mash-up of the API, and description data corresponding to the API, the meta-data, and the resource data; a data extracting unit which extracts the API, the meta-data, and the resource data from each of the API packages by using the description data; and a mash-up generating unit which generates a mash-up of the API packages by using the API, the meta-data, and the resource data.
- [38] The apparatus may further include a package download unit which downloads the API packages from a service provider or a server of a specified location and a storing unit which stores the downloaded API packages.
- [39] The API package obtaining unit may obtain the at least two API packages from the storing unit.
- [40] According to another aspect of the present invention, there is provided a computer-readable recording medium having embodied thereon a program for executing a method of providing an API service, the method including: generating meta-data for

executing an API; generating resource data for generating a mash-up of the API; generating description data corresponding to the API, the meta-data, and the resource data; and generating an API package comprising the API, the meta-data, the resource data, and the description data.

[41] According to another aspect of the present invention, there is provided a computer-readable recording medium having embodied thereon a program for executing a method of producing an API mash-up, the method including: obtaining at least two API packages which respectively comprise an API, meta-data for executing the API, resource data for generating a mash-up of the API, and description data corresponding to the API, the meta-data, and the resource data; extracting the API, the meta-data, and the resource data from each of the at least two API packages by using the description data; and generating a mash-up of the at least two API packages by using the API, the meta-data, and the resource data.

[42] According to another aspect of the present invention, there is provided a method of generating an Application Program Interface (API) mash-up, the method includes obtaining at least two API packages, extracting data from each of the at least two API packages and generating a mash-up of the at least two API packages by using the extracted data.

[43] The extracted data may include an API, meta-data for executing the API, resource data for generating a mash-up of the API, and description data corresponding to the API, the meta-data, and the resource data.

[44] Each of the at least two API packages may include an API, meta-data, resource data, and description data corresponding to the API, the meta-data, and the resource data .

### **Mode for Invention**

[45] Exemplary embodiments of the present invention will now be described in detail with reference to the appended drawings.

[46] FIG. 3 is a view of an example of an application program interface (API) package according to an exemplary embodiment of the present invention.

[47] Referring to FIG. 3, an API package 300 according to an exemplary embodiment the present invention includes meta-data 310, resource data 320, an API 330, and description data 340. The API package 300, which provides API services to an apparatus for producing an API mash-up, is used to provide information related to an API necessary to produce the mash-up in a standardized format.

[48] The meta-data 310 includes information necessary to execute the API 330. The meta-data 310 includes communication protocol information 316 used to send data to an API service provider and receive data from the API service provider, as a representative example, and data format information 318 used to input and output data of the API 330. The API service provider transmits its own API to the apparatus for producing the

API mash-up to be described later through a package providing server or transmits it directly. Accordingly, the meta-data 310 should include the communication protocol information 316 and the data format information 318 that are transmitted to/from the package providing server or the apparatus for producing the API mash-up. The communication protocol may be JavaScript, representational state transfer (REST), simple object access protocol (SOAP), and the like, and the data format may be XML, JSON, PHP, or the like.

- [49] The meta-data 310 may further include additional information such as user account information 314 or a user certification key 312 used to provide API services that are specialized according to users so that a mash-up developer can use the API.
- [50] The resource data 320 is used to produce a mash-up of the API 330. For example, the resource data 320 can be used to provide a user interface for producing the mash-up. The resource data 320 includes configuration data 326 of the user interface or tagging information 328 representing input/output data format of the API 330.
- [51] For example, the configuration data 326 includes information about a language, a resolution, a zoom, and the like that are applied to the user interface.
- [52] The tagging information 328 represents input/output data format of the API 330. For example, if the mash-up developer uses a Google map open API, the input data format and the output data format may be respectively a specified location and a set of coordinates. On the other hand, if the mash-up developer uses a Flickr open API, the input data format and the output data format may be respectively the name of a place and an image.
- [53] Therefore, in the case of the Google map open API, the tagging information 328 can be defined in a standardized format, for example, 'input = location' and 'output = coordinates'. This is for defining pairs of an input and output of the API with tags, because functions of each API are different. Also, a plurality of pieces of tagging information 328 can be input.
- [54] The resource data 320 may include information about icons or thumb nails for executing a mash-up of the corresponding API in a user interface for producing a mash-up.
- [55] The API 330, which is an entity of an API service, includes exemplary embodiments for substituting for arbitrary codes or scripts produced by the mash-up developer. The API 330 may be specialized according to development languages such as C or Java and may take the form of an API wrapper based on an extensible markup language (XML).
- [56] However, the API 330 may not only be an open API but also a local API. The local API is not allowed to be used by anybody as is the case with the open API, and is only allowed to be used by a special apparatus or a special person. For example, the local API is allowed to be used in a cellular phone of 'A' brand, and the mash-up developer

(that is, a user of the cellular phone) can produce the mash-up by putting the local API and the open API together or by putting the local API and another local API together. In other words, the local API, which is a peculiar API package included in an apparatus, may be installed inside the apparatus as a standardized package at the time of apparatus development.

- [57] The description data 340 includes detailed information about the API 330, the meta-data 310, and the resource data 320. An apparatus for producing an API mash-up to be described later can obtain detailed information about the API package 300 through the description data 340. For example, the description data 340 may include components of the API 330, a version of the API 330, a library function necessary to execute the API 330, and details about the meta-data 310 or the resource data 320, etc. For example, the description data 340 may be generated by an XML.
- [58] The above-described API package 300 is generated by an API service provider and can be transmitted to the apparatus for producing the API mash-up through the package providing server or can be transmitted directly.
- [59] FIG. 4 is a block diagram illustrating an apparatus for providing an API service 400 according to an exemplary embodiment of the present invention.
- [60] Referring to FIG. 4, the apparatus for providing the API service 400 includes a meta-data generating unit 420, a resource data generating unit 430, a description data generating unit 440, and an API package generating unit 450.
- [61] The apparatus for providing the API service 400 generates and provides the API package described in FIG. 3.
- [62] The meta-data generating unit 420 generates meta-data for executing the API 410. The meta-data includes information necessary to execute the API 410. The meta-data is transmitted to the API package generating unit 450.
- [63] The resource data generating unit 430 generates resource data for generating a mash-up of the API 410. The resource data is transmitted to the API package generating unit 450.
- [64] The description data generating unit 440 generates description data including detailed information about the meta-data transmitted from the API 410 and the meta-data generating unit 420, and about the resource data transmitted from the resource data generating unit 430.
- [65] The API package generating unit 450 inputs the meta-data that is outputted from the API 410 and the meta-data generating unit 420, the resource data that is outputted from the resource data generating unit 430, and the description data that is outputted from the description data generating unit 440, and generates and outputs an API package including those items of data.
- [66] FIG. 5 is a block diagram illustrating an apparatus for producing an API mash-up

500 according to an exemplary embodiment of the present invention.

- [67] Referring to FIG. 5, the apparatus for producing the API mash-up 500 includes an API package obtaining unit 510, a data extracting unit 520, and a mash-up generating unit 530.
- [68] The apparatus for producing the API mash-up 500 generates a mash-up using the API package received directly or through a package providing server from the apparatus for providing the API service 400 of FIG. 4.
- [69] The API package obtaining unit 510 obtains a plurality of API packages received directly or through the package providing server from the apparatus for providing the API service 400 of FIG. 4. For example, if the API package obtaining unit 510 obtains a map service of Google, an image sharing service of Yahoo, and a third service of a third subject, the API package obtaining unit 510 can receive a standardized API package provided by each service provider through a specified package providing server or can directly receive the standardized API package provided by each service provider from the apparatus for providing the API service.
- [70] As described above, each of the API packages includes an API, meta-data, resource data, and description data.
- [71] The data extracting unit 520 extracts the API, the meta-data, and the resource data from each of the API packages by using information included in the description data.
- [72] The mash-up generating unit 530 generates a mash-up of the API packages by using the API, the meta-data, and the resource data. Since information necessary to generate the mash-up is included in the meta-data, the mash-up generating unit 530 can use the information.
- [73] The mash-up generating unit 530 can determine an input/output data format between API services and can determine a method of communicating with a server, by using information about the provided communication protocol or data format.
- [74] The mash-up generating unit 530 realizes a user interface by using the resource data and reflects the API package in mash-up contents according to directions input from a user, on the basis of the user interface, so that different user interfaces can be produced for one mash-up. The mash-up generating unit 530 can have the same effect as a process in which a developer adds codes on the basis of the user interface and can reflect necessary information from the resource data and the API.
- [75] FIG. 6 is a view illustrating an apparatus for producing an API mash-up 600 according to another exemplary embodiment of the present invention.
- [76] FIG. 6 illustrates the apparatus for producing the API mash-up 600 having a more detailed configuration than the apparatus for producing the API mash-up 500 of FIG. 5.
- [77] A package download unit 610 downloads an API package from a service provider

- 670 (or a server of a specified location) according to a request of a mash-up generating unit 640.
- [78] A storing unit 620 stores API packages (that is, an open API package 622 and a local API package 624) downloaded by the package download unit 610 from a service provider 670. Also, the storing unit 620 stores mash-up contents 650 generated by the mash-up generating unit 640.
- [79] A data extracting unit 630 extracts API, meta-data, resource data, and description data from the API package. Each of the extracted data is transmitted to the mash-up generating unit 640.
- [80] The mash-up generating unit 640 generates mash-ups of the API packages by using the API, the meta-data, and the resource data. When another API package is required besides the API packages downloaded from the package download unit 610 in order to generate the mash-up, the mash-up generating unit 640 requests the package download unit 610 to download the corresponding API package. An example of the case in which another API package is required will be described later.
- [81] A mash-up driving unit 660 drives the mash-up contents 650 stored in the storing unit 620.
- [82] FIG. 7 is a flowchart of a method of providing an API service according to an exemplary embodiment of the present invention.
- [83] Referring to FIG. 7, an apparatus for providing an API service generates meta-data for executing an API (Operation 710). The API may be an open API or a local API. The meta-data includes information about a communication protocol to communicate with a package providing server or an apparatus for producing the API mash-up and information about a data format of the API. The meta-data may further include user account information or a user certification key for using the API.
- [84] The apparatus for providing the API service generates resource data for generating a mash-up of the API (Operation 720). The resource data comprises a configuration data of a user interface to generate the mash-up and a tagging information representing an input/output data format of the API. The resource data may further include information about an icon or a thumb nail for executing a mash-up of an API in the user interface.
- [85] The apparatus for providing the API service generates description data regarding the API, the meta-data, and the resource data (Operation 730). For example, the description data may include information about components of an API package, a version of the API, or a library function for executing the API.
- [86] The apparatus for providing the API service generates an API package comprising the API, the meta-data, the resource data, and the description data (Operation 740).
- [87] The apparatus for providing the API service transmits the API package to a package providing server or an apparatus for producing the API mash-up (Operation 750).

- [88] FIG. 8 is a flowchart of a method of producing an API mash-up according to an exemplary embodiment of the present invention.
- [89] Referring to FIG. 8, an apparatus for producing the API mash-up obtains at least two or more API packages which respectively comprise an API, meta-data, resource data, and description data (Operation 810). The API may be an open API or a local API.
- [90] The apparatus for producing the API mash-up can obtain the API packages from a package providing server storing a plurality of API packages provided by a plurality of API packages or can directly receive the API packages from an apparatus for providing an API service.
- [91] The meta-data includes information about a communication protocol to communicate with the package providing server or the apparatus for producing the API mash-up and information about a data format of the API. The meta-data may further include user account information or a user certification key for using the API.
- [92] The resource data comprises a configuration data of a user interface to generate the mash-up and a tagging information representing an input/output data format of the API. The resource data may further include information about an icon or a thumb nail for executing a mash-up of the API in the user interface.
- [93] The description data may include information about components of the API package, a version of the API, or a library function for executing the API.
- [94] The apparatus for producing the API mash-up extracts the API, the meta-data, and the resource data from each of the API packages by using the description data (Operation 820).
- [95] The apparatus for producing the API mash-up generates a mash-up of the API packages by using the API, the meta-data, and the resource data (Operation 830)
- [96] The apparatus for producing the API mash-up may replace one of the API packages in the mash-up with another API package, or may add another API package to the mash-up, or may remove a part of the API packages from the mash-up (Operation 840).
- [97] FIG. 9 is a flowchart of an example of a process for producing a mash-up of API packages as described in relation to FIG. 8.
- [98] Referring to FIG. 9, an apparatus for producing the API mash-up analyzes an input/output data format of an API (Operation 910). For example, if a mash-up developer performs a mash-up on a Google map open API and a Flickr open API, the input data format and the output data format of the Google map open API may be a specialized location and coordinates of the map, respectively. Also, the input data format and the output data format of the Flickr open API may be the name of a place and an image, respectively.
- [99] For example, the input/output data formats may be analyzed by using data format in-

formation of an API and meta-data, or tagging information of resource data.

- [100] The apparatus for producing the API mash-up searches another API package to perform the mash-up with the API packages selectively on the basis of the input/output data format of the API (Operation 920). As described above, since the output data format of the Google map open API is a set of coordinates and the input data format of the Flickr open API is the name of a place, the mash-up cannot be performed on the Google map open API and the Flickr open API as it is. Accordingly, the apparatus for producing the API mash-up searches other API packages in which the input data format is a set of coordinates and the output data format is the name of a place. For example, the apparatus for producing the API mash-up can search the corresponding API through the above-described package providing server. Also, the apparatus for producing the API mash-up can search an open API or a local API stored in a storing unit.
- [101] The apparatus for producing the API mash-up generates a mash-up by using API packages and another API package (Operation 930).
- [102] The present invention can also be embodied as computer readable code on a computer readable recording medium. The computer readable recording medium is any data storage device that can store data which can be thereafter read by a computer system. Examples of the computer readable recording medium include read-only memory (ROM), random-access memory (RAM), CD-ROMs, magnetic tapes, floppy disks, optical data storage devices, and flash memory.
- [103] While the present invention has been particularly shown and described with reference to exemplary embodiments thereof, it will be understood by one of ordinary skill in the art that various changes in form and details may be made therein without departing from the spirit and scope of the present invention as defined by the following claims.

## Claims

- [1] 1. A method of providing an Application Program Interface (API) service, the method comprising:  
generating meta-data for executing an API;  
generating resource data for generating a mash-up of the API;  
generating description data corresponding to the API, the meta-data, and the resource data; and  
generating an API package comprising the API, the meta-data, the resource data, and the description data.
- [2] 2. The method of claim 1, wherein the API comprises one of an open API and a local API.
- [3] 3. The method of claim 1, further comprising transmitting the API package to a package providing server or an apparatus for producing the mash-up of the API.
- [4] 4. The method of claim 2, wherein the meta-data comprises information about a communication protocol to communicate with the package providing server or the apparatus for producing the mash-up of the API, and information about a data format of the API.
- [5] 5. The method of claim 4, wherein the meta-data further comprises at least one of user account information and a user certification key for using the API.
- [6] 6. The method of claim 1, wherein the resource data comprises a configuration data of a user interface to generate the mash-up and a tagging information representing an input/output data format of the API.
- [7] 7. The method of claim 6, wherein the resource data further comprises information about an icon or a thumb nail for executing the mash-up of the API in the user interface.
- [8] 8. The method of claim 1, wherein the description data comprises information about one of components of the API package, a version of the API, and a library function for executing the API.
- [9] 9. A method of generating an Application Program Interface (API) mash-up, the method comprising:  
obtaining at least two API packages which respectively comprise an API, meta-data for executing the API, resource data for generating a mash-up of the API, and description data corresponding to the API, the meta-data, and the resource data;  
extracting the API, the meta-data, and the resource data from each of the at least two API packages by using the description data; and  
generating a mash-up of the at least two API packages by using the API, the

meta-data, and the resource data.

- [10] 10. The method of claim 9, wherein the API comprises one of an open API and a local API.
- [11] 11. The method of claim 9, wherein the obtaining of the API packages comprises:  
receiving the at least two API packages from a package providing server or an apparatus for providing an API service.
- [12] 12. The method of claim 10, wherein the meta-data comprises information about a communication protocol to communicate with the package providing server or the apparatus for producing the API mash-up and information about a data format of the API.
- [13] 13. The method of claim 12, wherein the meta-data further comprises at least one of user account information and a user certification key for using the API.
- [14] 14. The method of claim 9, wherein the resource data comprises a configuration data of a user interface to generate the mash-up and a tagging information representing an input/output data format of the API.
- [15] 15. The method of claim 9, wherein the resource data further comprises information about an icon or a thumb nail for executing the mash-up of the API in the user interface.
- [16] 16. The method of claim 14, wherein the description data comprises information about one of components of the API package, a version of the API, and a library function for executing the API.
- [17] 17. The method of claim 9, wherein the generating of the mash-up of the at least two API packages comprises:  
analyzing an input/output data format of the API;  
searching another API package to perform the mash-up with the at least two API packages selectively on the basis of the input/output data format of the API; and  
generating the mash-up by using the at least two API packages and the another API package.
- [18] 18. The method of claim 9, further comprising:  
replacing one of the at least two API packages in the mash-up with another API package.
- [19] 19. The method of claim 9, further comprising:  
adding another API package to the mash-up.
- [20] 20. The method of claim 9, further comprising:  
removing a part of the API packages from the mash-up.
- [21] 21. An apparatus for providing an API service, the apparatus comprising:  
a meta-data generating unit which generates meta-data for executing an API;

a resource data generating unit which generates resource data for generating a mash-up of the API;

a description data generating unit which generates description data corresponding to the API, the meta-data, and the resource data; and

an API package generating unit which generates an API package comprising the API, the meta-data, the resource data, and the description data.

[22] 22. An apparatus for producing an API mash-up, the apparatus comprising:  
an API package obtaining unit which obtains at least two API packages which respectively comprise an API, meta-data for executing the API, resource data for generating a mash-up of the API, and description data corresponding to the API, the meta-data, and the resource data;

a data extracting unit which extracts the API, the meta-data, and the resource data from each of the API packages by using the description data; and

a mash-up generating unit which generates a mash-up of the API packages by using the API, the meta-data, and the resource data.

[23] 23. The apparatus for producing an API mash-up according to claim 22, the apparatus further comprising:

a package download unit which downloads the API packages from a service provider or a server of a specified location; and

a storing unit which stores the downloaded API packages, wherein the API package obtaining unit obtains the at least two API packages from the storing unit.

[24] 24. A computer-readable recording medium having embodied thereon a program for executing a method of providing an API service, the method comprising:  
generating meta-data for executing an API;  
generating resource data for generating a mash-up of the API;  
generating description data corresponding to the API, the meta-data, and the resource data; and

generating an API package comprising the API, the meta-data, the resource data, and the description data.

[25] 25. A computer-readable recording medium having embodied thereon a program for executing a method of producing an API mash-up, the method comprising:  
obtaining at least two API packages which respectively comprises an API, meta-data for executing the API, resource data for generating a mash-up of the API, and description data corresponding to the API, the meta-data, and the resource data;

extracting the API, the meta-data, and the resource data from each of the at least two API packages by using the description data; and

generating a mash-up of the at least two API packages by using the API, the meta-data, and the resource data.

- [26] 26. A method of generating an Application Program Interface (API) mash-up, the method comprising:  
obtaining at least two API packages;  
extracting data from each of the at least two API packages; and  
generating a mash-up of the at least two API packages by using the extracted data.
- [27] 27. The method of generating an Application Program Interface mash-up according to claim 26, wherein the extracted data comprise an API, meta-data for executing the API, resource data for generating a mash-up of the API, and description data corresponding to the API, the meta-data, and the resource data.
- [28] 28. The method of generating an Application Program Interface mash-up according to claim 26, wherein each of the at least two API packages comprises an API, meta-data, resource data, and description data corresponding to the API, the meta-data, and the resource data .

FIG. 1

120

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>GMapFlickr</title>
<script src="http://maps.google.com/maps?file=api&api.v=2&amP=key=ABQIAAAAV11XDdMl6ArUVEsqDN_r1BRsBFGLz95u3NNH023FYmaV1AvclhSjzPbXfntXJBNdAgVRLQG4Dx3nw" type="text/javascript">
</script>
<script src="http://www.google.com/jsapi?key=ABQIAAAAV11XDdMl6ArUVEsqDN_r1BRsBFGLz95u3NNH023FYmaV1AvclhSjzPbXfntXJBNdAgVRLQG4Dx3nw" type="text/javascript"></script><script type="text/javascript">
Function load() {...}
Function getClickedLocation(latitude, longitude) {...}
Function mapClicked(marker, point) {...}
Function getPhotosFromFlickr(location) {...}
Function CreateLHTTPRequest(xmlhttp) {...}
</script>
</head>
<body onload="load()" onunload="GUnload()">
<div id="map" style="width:170px;height:170px"><div>
<table border="1">
<tr>
<td><img src="" alt="" name="Flickrimage1"></img></td>
<td><img src="" alt="" name="Flickrimage2"></img></td>
<td><img src="" alt="" name="Flickrimage3"></img></td>
<td><img src="" alt="" name="Flickrimage4"></img></td>
<td><span id="img5"></span></td>
</tr>
</table>
<td><span id="ClickedLocation"></span></td>
</tr>
</table>
<table border="1">
<tr>
<td><div id="searchcontrol">Location Results</div></td>
</tr>
<tr>
<td><span id="ClickLocn"></span></td>
</tr>
</table>
</body>
</html>

```

Google+ flickr

110

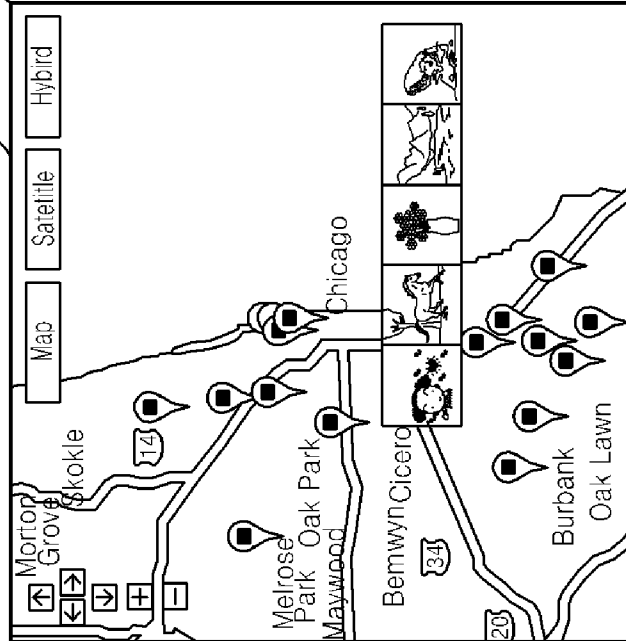
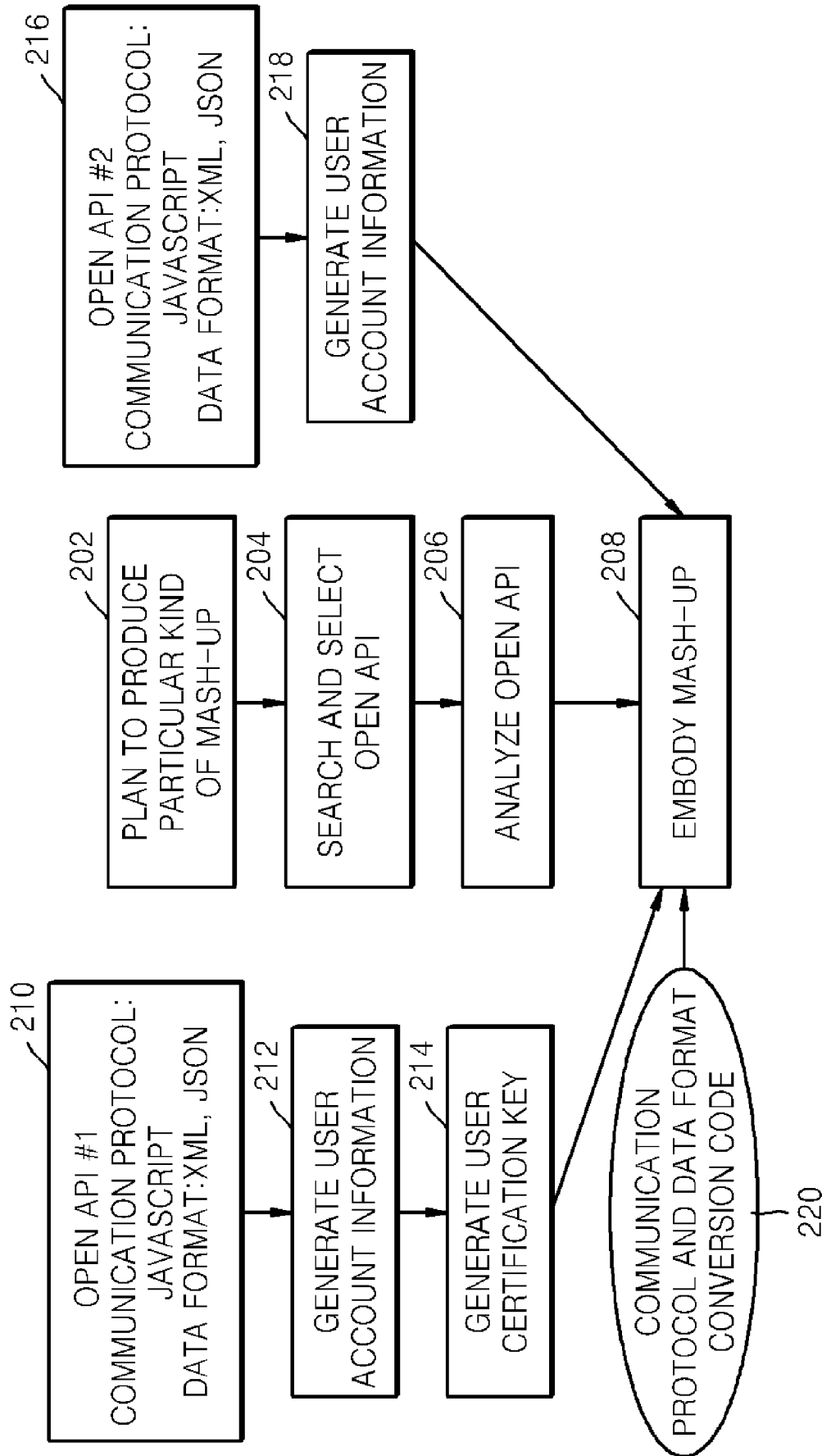


FIG. 2



**FIG. 3**

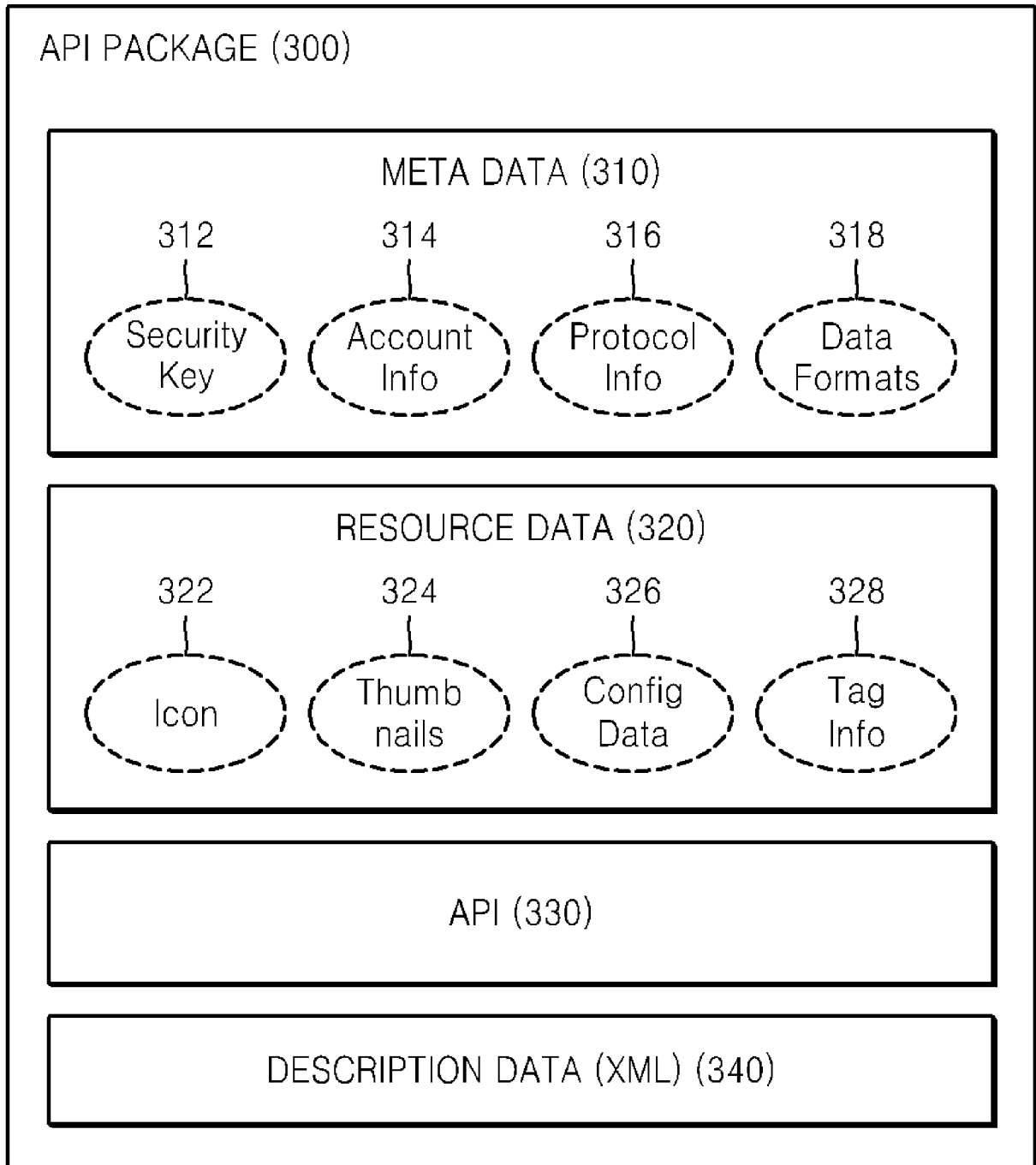


FIG. 4

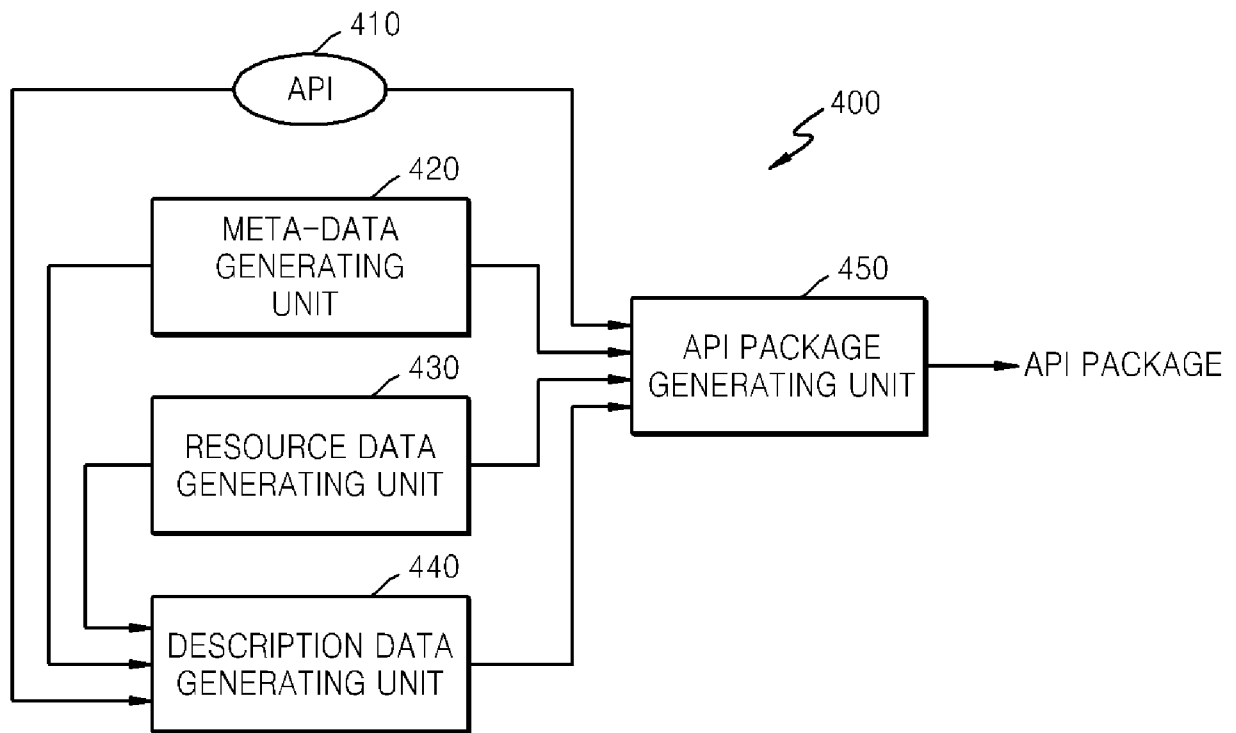


FIG. 5

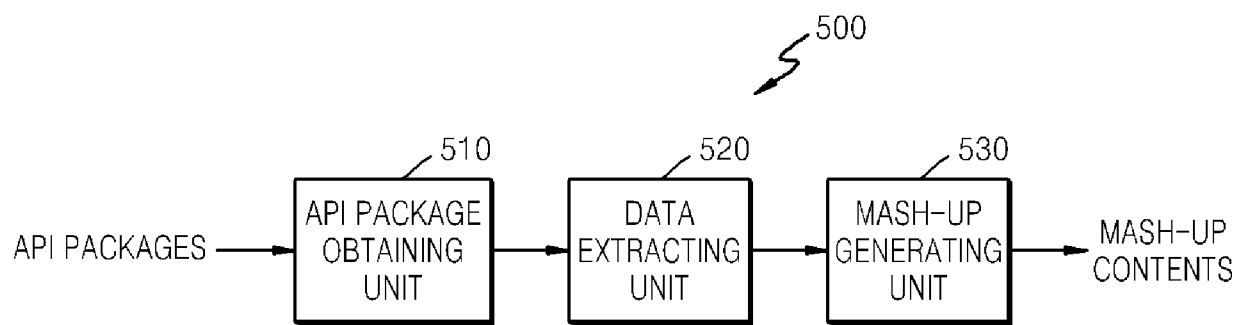
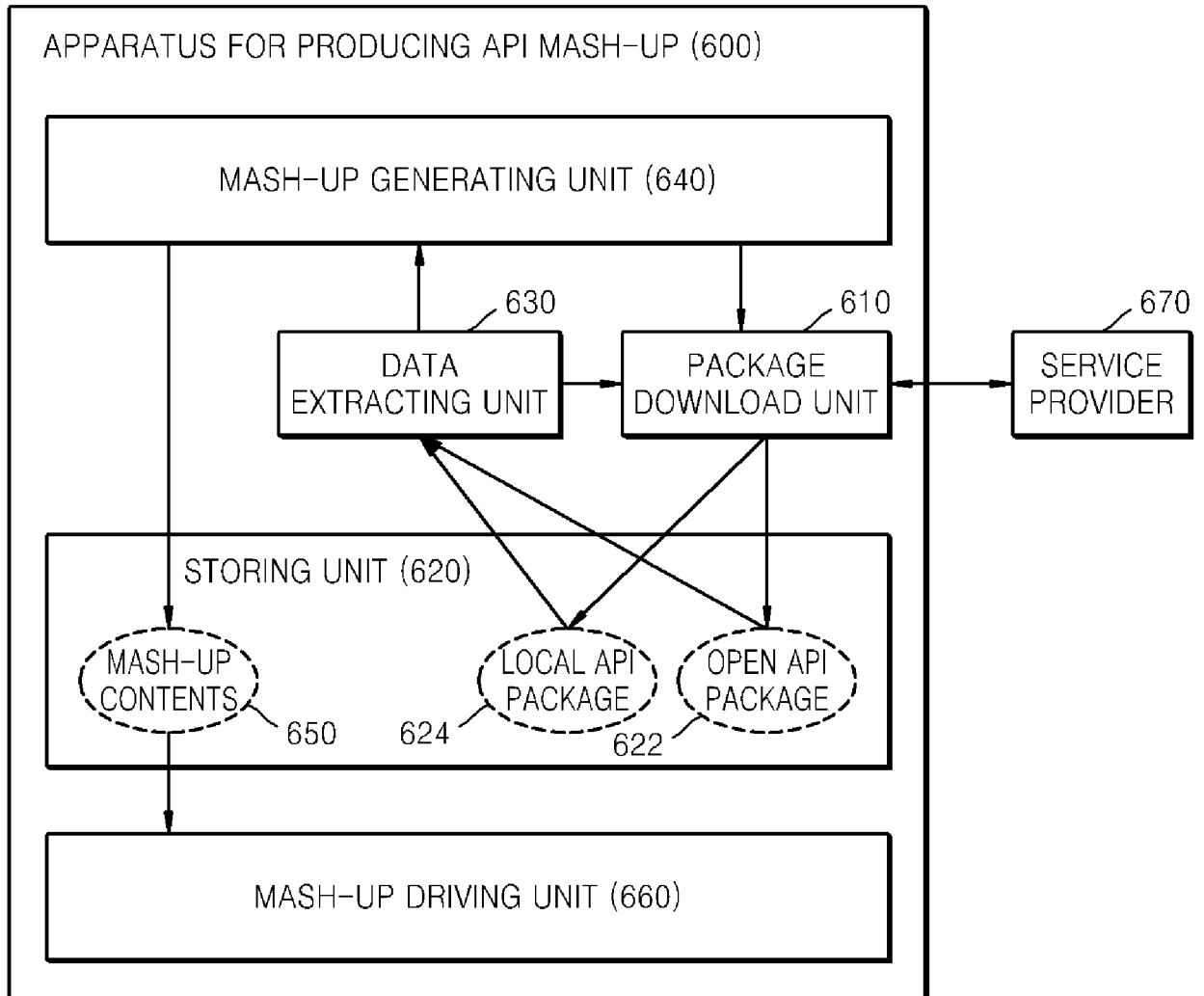
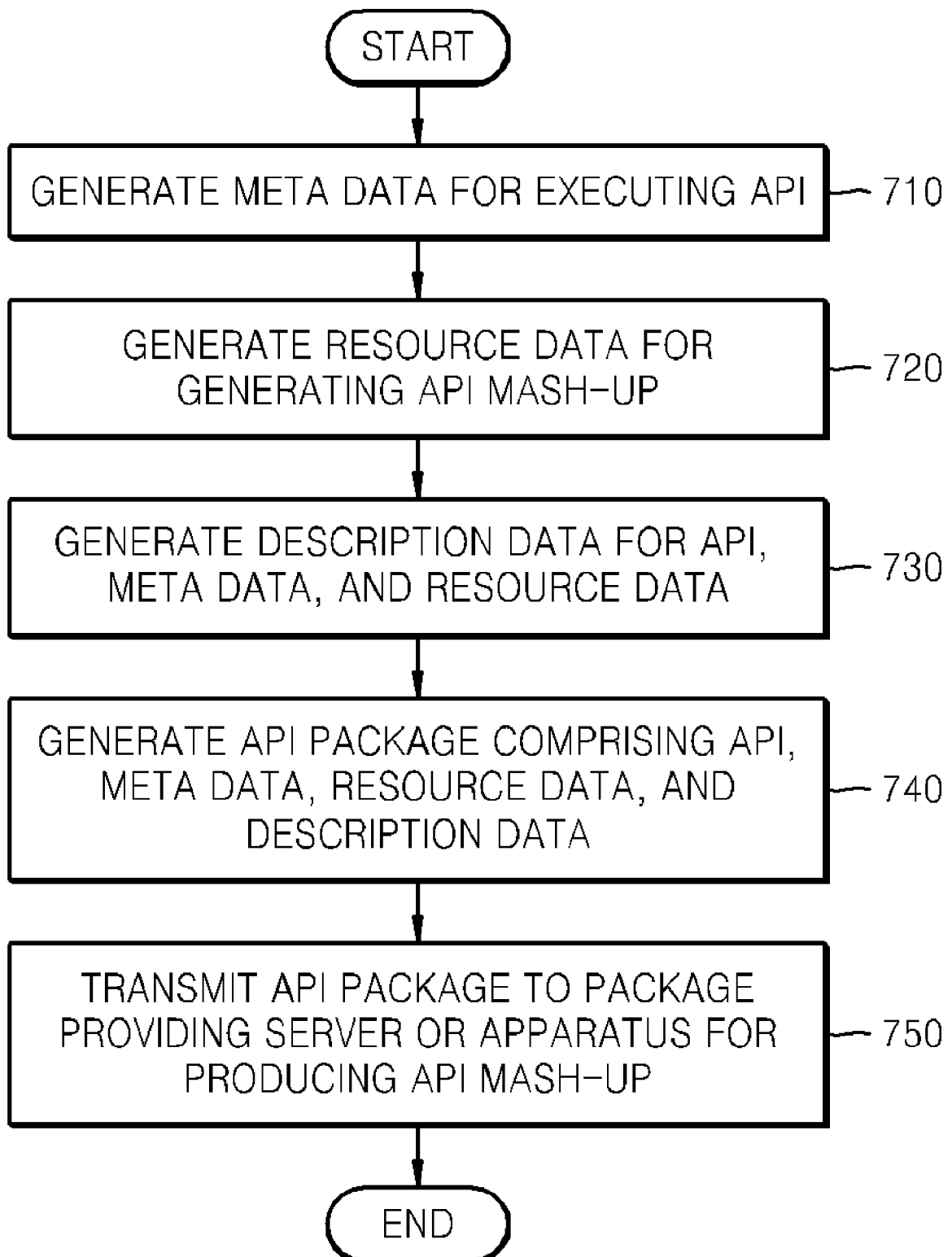
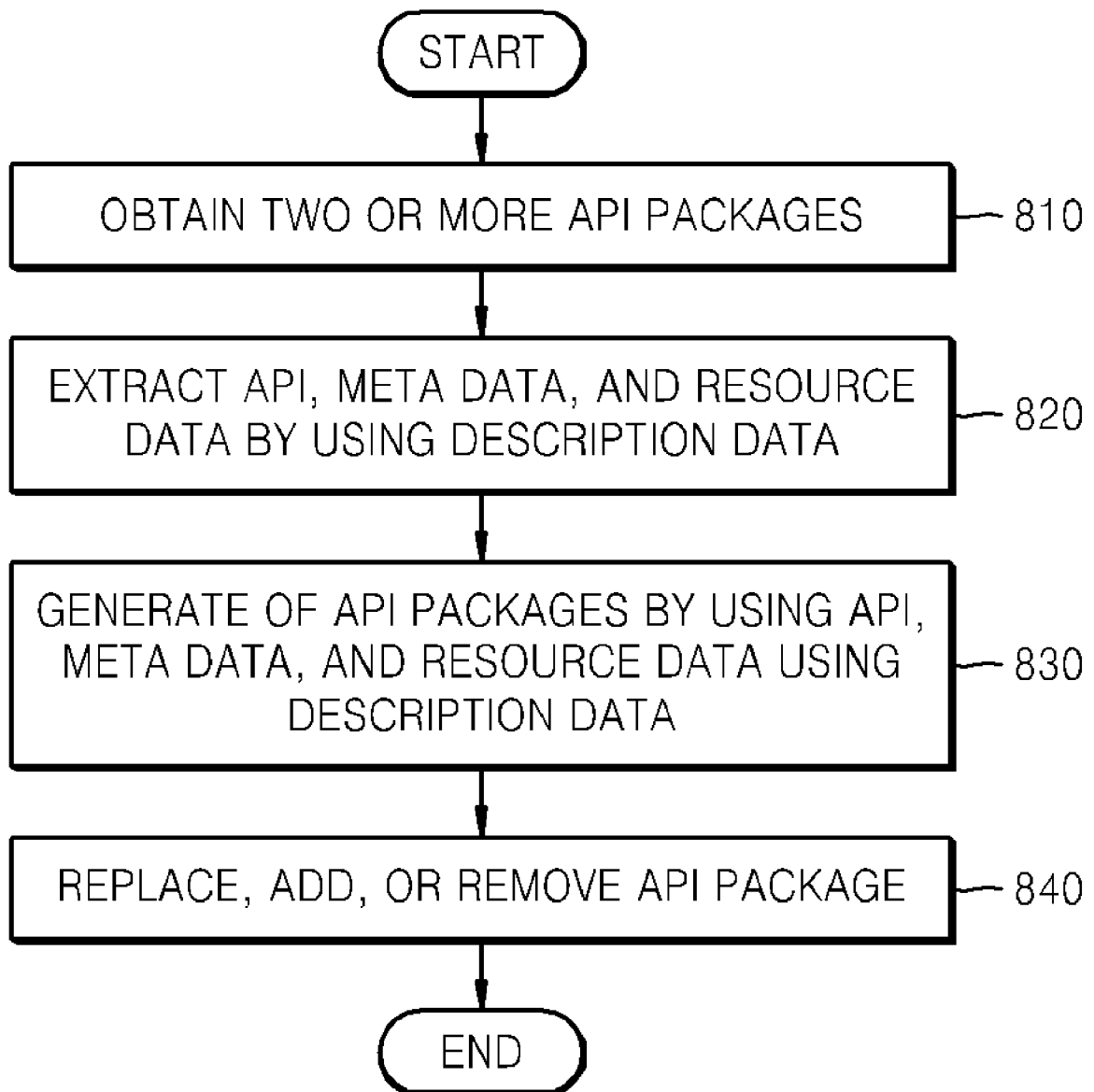
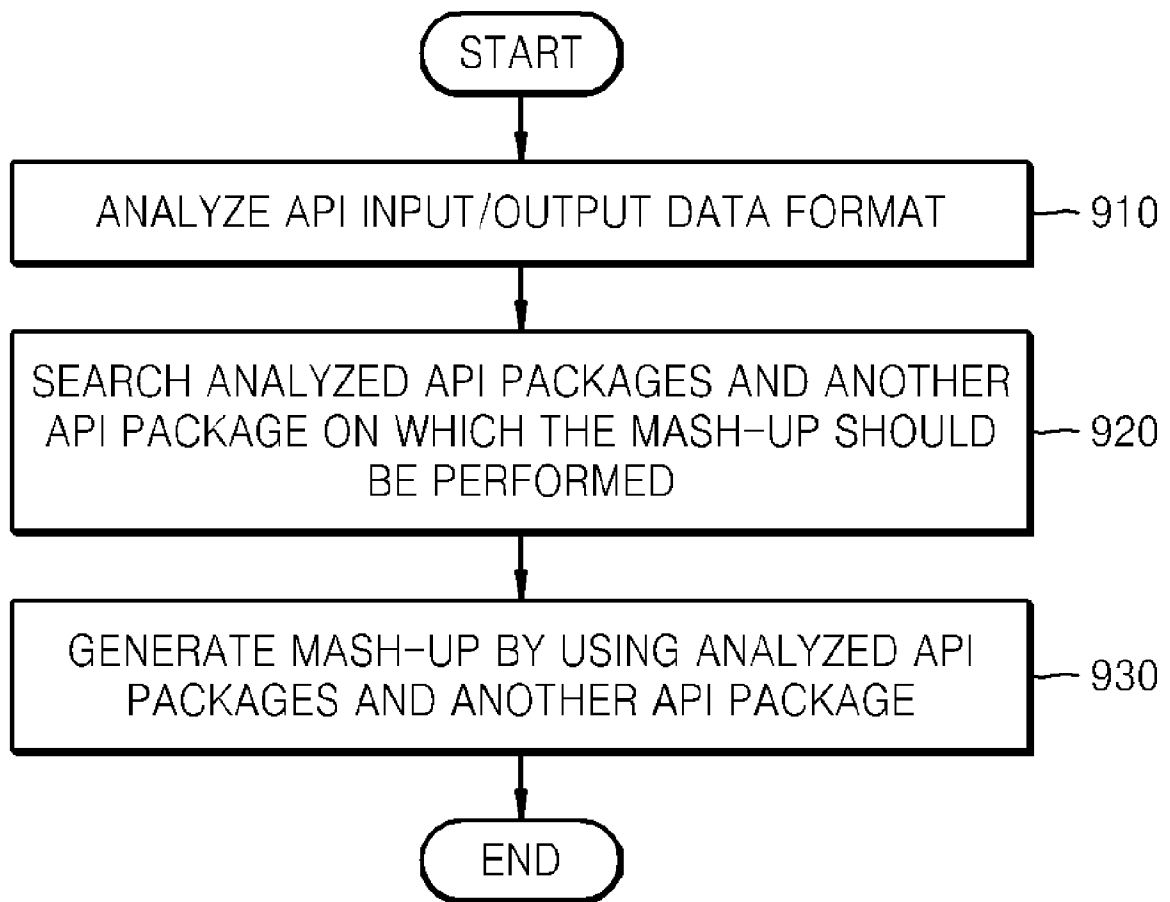


FIG. 6



**FIG. 7**

**FIG. 8**

**FIG. 9**

**A. CLASSIFICATION OF SUBJECT MATTER****G06F 17/00(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

IPC G06Q 50/00C3, G06F 9/06, G06F 17/00

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

Korean Utility models and applications for Utility models since 1975

Japanese Utility models and applications for Utility models since 1975

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKIPASS (KIPO internal) &amp; keyword : "open API, API,API package, mash-up, mashup, extract,meta data, resource data"

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	MI-JI LEE, WAN-SOO IM, JUNE-HWAN KOH, YUN-SOO CHOI. A study on the Web Portal Site Development Model encouraging Public Participation in Smart Growth Planning Processes by Utilizing Publicly Available GIS Tools. Journal of the Korea Open Geographic Information Systems Association. June 15 2007, pages 430 and 435.	21-23
A	KR 10-0700626 B1 (NEOMTEL CO., LTD.) 28 March 2007 See the abstract; claims 1-11; figures 1-6.	21-23
A	SOO-JEONG SHIN, MYUNG-SUK YUN. A strategy to introduce Open-API in e-Government. e-Government Focus. October 2007, pages 21 to 29.	21-23
A	KR 10-2005-0034824 A (SK TELECOM CO., LTD.) 15 April 2005 See the abstract; claims 1-2; figure 1.	21-23

 Further documents are listed in the continuation of Box C. See patent family annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&amp;" document member of the same patent family

Date of the actual completion of the international search

21 MAY 2009 (21.05.2009)

Date of mailing of the international search report

**22 MAY 2009 (22.05.2009)**

Name and mailing address of the ISA/KR

Korean Intellectual Property Office  
Government Complex-Daejeon, 139 Seonsa-ro, Seo-  
gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

PARK Mi Jeong

Telephone No. 82-42-481-8379



**Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)**

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1.  Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
  
2.  Claims Nos.: 1-20,24-28  
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:  
  
The claims are too broad to make meaningful search.
  
3.  Claims Nos.:  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

**Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)**

This International Searching Authority found multiple inventions in this international application, as follows:

1.  As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2.  As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3.  As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
  
4.  No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

**Remark on Protest**

- The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- No protest accompanied the payment of additional search fees.

**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.

**PCT/KR2008/006031**

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
KR 10-0700626 B1	28.03.2007	None	
KR 10-2005-0034824 A	15.04.2005	None	