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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 September 2009 (11.09.2009)

PCT

(10) International Publication Number
WO 2009/111151 A1

(54) Title: DEFINITION FOR SERVICE INTERFACE

Fig. 1

(57) Abstract: Various embodiments provide a model through which service providers can describe offered services using a standardized format. In one or more embodiments, the standardized format is declarative and enables service providers to describe their associated services in a standardized way. In at least some embodiments, the standardized format includes a set of common service properties that are shared across multiple different services. Additionally, service specific properties can be described in addition to the common service properties. In one or more embodiments, existing services can be extended by including, in the standardized format, a description of an extension. In one or more other embodiments, new services can be added for consumption by simply including a description of the new service using the standardized format.

DEFINITION FOR SERVICE INTERFACE

BACKGROUND

[0001]   Today, in order for a service, such as a web service, to integrate with client side applications, many solutions use loose integration "hooks" to enable the service to plug into the application experience. As those in the industry will attest, few truly successful examples of loose integration exist. This is due in large part to the difficulty of delivering a rich user experience when services differ in capacity and, to the related problem of attempting to clearly define an interface between the client and the service in a way that can be easily understood and implemented.
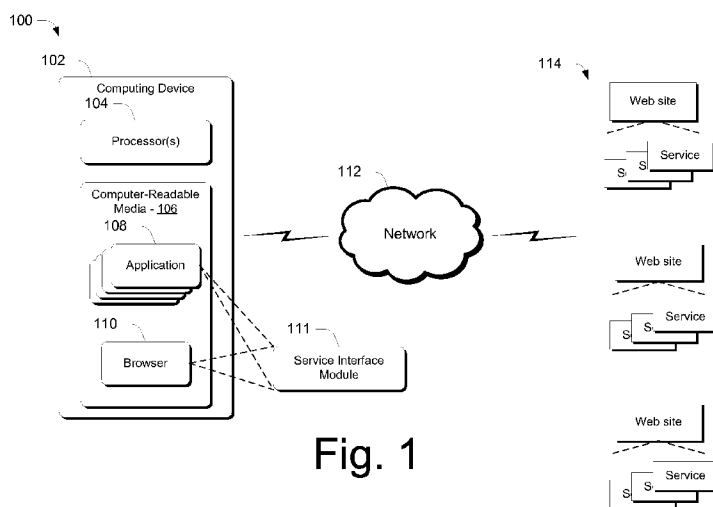
SUMMARY

[0002]   Various embodiments provide a model through which service providers can describe offered services using a standardized format. In one or more embodiments, the standardized format is declarative and enables service providers to describe their associated services in a standardized way. In at least some embodiments, the standardized format includes a set of common service properties that are shared across multiple different services. Additionally, service-specific properties can be described in addition to the common service properties.

[0003]   In one or more embodiments, existing services can be extended by including, in the standardized format, a description of an extension. In one or more other embodiments, new services can be added for consumption by simply including a description of the new service using the standardized format.

[0004]   In at least some embodiments, the standardized format is represented through a declarative, hierarchical tag-based language.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005]   Fig. 1 illustrates a system in which the inventive principles can be employed in accordance with one or more embodiments.

[0006]   Fig. 2 is a flow diagram that describes steps in a method in accordance with one or more embodiments.

[0007]   Fig. 3 is a flow diagram that describes steps in a method in accordance with one or more embodiments.

[0008]   Fig. 4 is a block diagram of a system that can be utilized to implement one or more embodiments.

DETAILED DESCRIPTION

Overview

[0009]   Various embodiments provide a model through which service providers can describe offered services using a standardized format. In one or more embodiments, the standardized format is declarative and enables service providers to describe their associated services in a standardized way. In at least some embodiments, the standardized format includes a set of common service properties that are shared across multiple different services. Additionally, service specific properties can be described in addition to the common service properties. In one or more embodiments, existing services can be extended by including, in the standardized format, a description of an extension. In one or

more other embodiments, new services can be added for consumption by simply including a description of the new service using the standardized format.

[0010] In at least some embodiments, the standardized format is represented through a declarative, hierarchical tag-based language. By way of example and not limitation, a suitable hierarchical text-based language is XML. It is to be appreciated and understood, however, that other methods and ways of representing the standardized format can be utilized without departing from the spirit and scope of the claimed subject matter.

[0011] In the discussion that follows, a section entitled "Operating Environment" describes an operating environment that can be utilized to practice the inventive principles described herein in accordance with one or more embodiments. Following this, a section entitled "Example Standardized Format" is provided and describes but one example of a standardized format in accordance with one or more embodiments. Following this, a section entitled "Common Service Properties-Example" describes some examples of common service properties in accordance with one or more embodiments. Next, a section entitled "Extending an Existing Service" is provided and describes an embodiment in which an existing service can be extended. Following this, a section entitled "Adding a New Service" describes how a new service can be added in accordance with one or more embodiments. Next, a section entitled "Example Methods" is provided and describes methods in accordance with one or more embodiments. Last, a section entitled "Example System" is provided and describes but one system that can be utilized in accordance with one or more embodiments.

Operating Environment

[0012] Fig. 1 illustrates an operating environment in accordance with one or more embodiments, generally at 100. Environment 100 includes a computing device 102 having

one or more processors 104, one or more computer-readable media 106 and one or more

applications 108 that reside on the computer-readable media and which are executable by

the processor(s). Applications 108 can include any suitable type of application such as, by

way of example and not limitation, reader applications, e-mail applications, instant

messaging applications, and a variety of other applications. In addition, a Web browser

110 is provided and includes a service interface module 111 that is configured to operate

as described above and below. Specifically, the service interface module 111 is

configured to operate using a model that provides standardized format through which

various types of services can be described. Service descriptions using the standardized

format can describe not only a common set of properties shared among various services,

but also service-specific properties that describe nuances associated with how a service is

to be consumed, as will become apparent below.

[0013]   It is to be appreciated and understood that while service interface module 111 is

shown as comprising part of Web browser 110, the service interface module can manifest

itself as a stand-alone component that is used by the Web browser. In addition, the service

interface can be used by other applications, services, and entities including, by way of

example and not limitation, other service providers and Web-based applications.

[0014]   The computer-readable media can include, by way of example and not

limitation, all forms of volatile and non-volatile memory and/or storage media that are

typically associated with a computing device. Such media can include ROM, RAM, flash

memory, hard disk, removable media and the like. One specific example of a computing

device is shown and described below in Fig. 4.

[0015]   In addition, environment 100 includes a network 112, such as the Internet, and

one or more web sites 114 from and to which content can be received and sent. Websites

114 can offer a variety of services that can be consumed by applications 108 and Web

browser 110, as will become apparent below. In addition, in at least some embodiments, services can be offered locally on a client computing device.

[0016] Computing device 102 can be embodied as any suitable computing device such as, by way of example and not limitation, a desktop computer, a portable computer, a handheld computer such as a personal digital assistant (PDA), cell phone, and the like.

Example Standardized Format

[0017] The discussion below describes an example standardized format that can be used to describe various different types of services. It is to be appreciated and understood that the description below constitutes an example only and is not to be used to limit application of the claimed subject matter to the specific example format described.

[0018] In the model described below, service providers can describe offered services using a standardized format which, in this particular example, employs XML. The standardized format is declarative in the sense that it defines or otherwise declares properties associated with various services that are offered by service providers. In addition, in at least some embodiments, consumers of the standardized format, such as applications and other entities, can select portions of the format to support. For example, some applications may support the full format while other applications might support a subset of the format. In the discussion that follows, a set of common service properties are first described. In at least some embodiments, the set of common service properties are shared across multiple different services. Following this, a discussion of service-specific properties is provided. The service-specific properties are used to describe specific attributes or properties associated with a particular service. These service-specific

properties can be used to describe a service being offered, a service that has been extended, and/or a new service.

Common Service Properties-Example

[0019]   In one or more embodiments, the standardized format describes a common set of service properties that are employed by one or more services. In at least some embodiments, many of the services that can be consumed an application, such as a Web browser, use the standardized format to describe the common set of service properties.

[0020]   In one or more embodiments, the common set of service properties includes, by way of example and not limitation, a property associated with a homepage URL of the service and a display, such as a visually-renderable display, associated with the service. As but one example of how these common service properties can be represented using the standardized format, consider the following:

```
<OpenServiceDescription>
<homepageUrl />
<display />
</OpenServiceDescription>
```

[0021]   In one or more embodiments, the common set of service properties represents a set of properties that services embody. By way of example and not limitation, these properties can include: a name property, an icon property, and a description property. Service-specific functionality is described within its own node in the standardized format. Using the standardized format can allow applications to reliably support service functionality and enable a consistent user experience. As an embellishment of the XML representation shown above, consider the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<openServiceDescription
    xmlns="http://www.microsoft.com/schemas/openservicedescription/1.0">
<homepageUrl>http://encarta.msn.com </homepageUrl>
    <display>
        <name>Define with Encarta</name>
        <icon>http://encarta.msn.com/favicon.ico </icon>
        <description>Encarta provides up-to-date encyclopedia, dictionary,
    and atlas </description>
    </display>
```

[0022]   The first thing to notice about the representation above is that, in one or more

embodiments, a specific namespace is defined for the described services. This is useful, as

will be appreciated by the skilled artisan, because it helps to fully qualify a service and,

accordingly, avoid collisions with other similarly-named services which are not qualified

by the same namespace.

[0023]   Further, in this example, a <homepageUrl> tag describes a homepage

associated with the particular service. In this particular example, the homepage URL is

"encarta.msn.com". In one or more embodiments, a service has an ID which, in this

example, is a combination of the homepageURL and the type of service. Other service

IDs can, of course, be used. Additionally, the <display> tag describes three properties

associated with the particular service. In this particular example, these properties include,

by way of example and not limitation, a name, an icon, and a description. Notice that the

icon property contains a link to an icon associated with the service.

[0024]   Having now discussed an example of a common set of service properties that

are shared across multiple different services, consider now a discussion of two specific

types of services which serve as an illustration of how a particular service can be

represented using the standardized format. It is to be appreciated and understood that

these specific services are provided as an example and are not intended to limit application of the claimed subject matter to these particular services. Rather, other services can be described using the standardized format without departing from the spirit and scope of the claimed subject matter.

[0025]   In the example that follows, services in the form of an activity service and an e-mail service are described.

[0026]   With regard to an activity service, such can be considered as a specific type of service that is used to send content to a webpage. The description of the activity service uses the standardized format and describes the type of content a service can handle, what to display on a preview of a selected service, and where to navigate the user on execution of a particular activity. As an example of a representation of an activity service using the standardized format, consider the following:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<openServiceDescription
        xmlns="http://www.microsoft.com/schemas/openservicedescription/1.0
">
        <homepageUrl>http://maps.yahoo.com </homepageUrl>
        <display>
            <name>Map with Yahoo!</name>
            <icon>http://www.yahoo.com/favicon.ico </icon>
        </display>
        <activity category="map">
            <activityAction context="selection" >
                <preview action =
"http://api.maps.yahoo.com/livewords/livewords.php?entity={selection}"/>
                <execute
                    method="get"
                    action="http://maps.yahoo.com/">
                    <parameter name="csz" value="{selection}" type="text" />
                </execute>
            </activityAction>
```

```
                </activity>
            </openServiceDescription>
```

[0027] In the illustrated and described embodiment, the standardized format can be utilized to describe the category of a particular activity service. For example, in the representation above, an <activity> tag includes a category property. In this particular example, the category property describes a "map" which defines this activity service as a mapping service. Other types of categories can, however, be used. These types of other categories include, by way of example and not limitation, e-mail activities, define activities, translate activities, and the like.

[0028] The nature of an activity service definition is such that it can define different actions that a user can take and what actions to perform responsive to a user action. In the illustrated and described embodiment, the <activityAction> tag is used to describe what happens when the user makes a selection or when the user right clicks on a link. In this particular example, when the user makes a selection, the user invokes an activity such as a mapping activity. So, for example, when the user selects an address on a webpage, the selection can be used to implement a preview action. To implement a preview action, the user's selection is provided into the URL listed under the <preview> tag. This URL is sent to an associated service provider that then processes the user's selection and returns an associated visual preview for the user to see. If, after seeing the preview, the user wishes to have the full mapping function executed, they can simply click on the displayed preview to invoke the functionality described in the <execute> tag. Additionally, services can obtain properties of a document and a user's selection as variables. These variables can be expressed as part of the action URL or through form-based parameters. In accordance with one or more embodiments, the following variables can be supported:

- documentUrl
- documentTitle
- documentDomain
- selection
- link
- linkTitle
- linkRel
- linkType
- linkDomain

[0029] In the above example, variables are enclosure in brackets {}. A variable can be specified as optional by using a "?" after the variable name. If the value of an optional variable is empty, an empty text string in used.

[0030] Inline parameters can be used to specify inputs to a service through a URI template. As an example, consider the following:

```
<execute method="get"
action="http://foo.com/done.aspx?url={documentUrl}&title={documentTitle?
}" />
```

[0031] In this example, documentUrl is a required variable and documentTitle is an optional variable. If the value of documentTitle is empty, then an empty string is used. If the value of documentUrl is empty, then the Activity errors.

[0032] Form-based parameters can be used to specify inputs to a service through name-value pairs, similar to HTTP form submissions. This is useful for making HTTP post requests or if the HTTP get request could be longer than the URL character limit.

[0033] The parameter element is used to list name-value pairs. This uses the same example as the inline parameters above:

```
<execute method="post" action="http://foo.com/done.aspx>
```

```
        <parameter name="url" value="{documentUrl}" />
        <parameter name="title" value="{documentTitle?}" />
    </execute>
```

[0034]   Here, if the variable is indicated as optional and the value is empty, then the value of the optional variable is an empty string. If the variable is required and the value of the variable is empty then the entire name-value pair is not used as part of the form submission.

[0035]   In one or more embodiments, one can also use the action URL and parameters as a way to track the usage of a service. This can be done by adding a special parameter with a special value indicating that it is from a browser, such as Internet Explorer 8. As an example, consider the following:

```
    <execute method="post" action="http://foo.com/done.aspx>
        <parameter name="url" value="{documentUrl}" />
        <parameter name="source" value="IE8" />
    </execute>
```

[0036]   Returning to the example above, the functionality described in the <execute> tag describes an HTTP "get" method which uses the associated URL described by the "action" property in conjunction with the data provided by the <parameter> tag. In operation, the user can be navigated to the listed URL and have their particular selection mapped by the webpage associated with the URL.

[0037]   With respect to an e-mail service, such can be considered as a specific type of service that describes the functionality of web-based e-mail. In operation, the standardized format can be used to describe an URL to an inbox, an URL to get updates to the inbox, and how to send content to a new compose message.

[0038]   As of example representation of an e-mail service using the standardized

format, consider the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<openServiceDescription
     xmlns="http://www.microsoft.com/schemas/openservicedescriptio
n/1.0">
     <homepageUrl>http://hotmail.live.com </homepageUrl>
     <display>
        <name>Hotmail</name>
     </display>
     <email>
        <compose action=" http://www.hotmail.com/secure/start">
            <parameter     name="subject"     value="{subject?}
        type="text" />
            <parameter  name="body"  value="{body?}  type="html"
        />
        </compose>
        <composePreview
     action="http://www.hotmail.com/Recipients.htm" />
        <inbox
           open=" http:// mail.live.com/mail/"
           monitor="http://mail.live.com/feed.xml" >
        </inbox>
     </email>
</openServiceDescription>
```

[0039]   In the above example, after the description of the common service properties, a

separate <email> tag is provided. The <compose> tag contains a description of

information associated with composing a message. The illustrated URL describes where

information is sent when the user composes an e-mail message. The various parameters

within this tag pertain to information that is to be included in the message. Specifically,

there is a subject field and a body field.

[0040]   In addition, a user can also receive a preview by virtue of the preview URL that

is described in the <composePreview> tag. When this is invoked, a webpage referenced

by the URL is fetched and surfaced for the user.

[0041]   The final part of the XML representation above describes an inbox

representation. This representation describes information associated with allowing a user

to check their inbox to ascertain if there are any updates. In operation, the browser can

create a button in the chrome of the user interface which allows the user to check their e-

mail. In the illustrated and described embodiment, the XML representation describes an

URL associated with the e-mail updates. In this particular example, the < monitor> tag

describes a feed to which a browser can subscribe to receive e-mail updates.

[0042]   Having discussed some example services, consider now a discussion of how an

existing service can be extended and how a new service can be added in accordance with

one or more embodiments.

Extending an Existing Service

[0043]   In one or more embodiments, the standardized format provides a mechanism by

which existing services can be extended. To extend an existing service, one simply

includes a description of that service extension using the standardized format. In at least

some embodiments, an existing service can be extended by defining a new namespace for

the service extension, and then describing the extension parameters relative to the newly-

defined namespace. As an example, consider the following XML representation. In this

example, an existing activity service is extended with thesaurus functionality.


```
<?xml version="1.0" encoding="UTF-8"?>
<openServiceDescription
```

```
            xmlns="http://www.microsoft.com/schemas/openservicedescrip
tion/1.0"
            xmlns:word="http://word.service.org/schemas/wordservice/1.
0" >
            <homepageUrl>http://encarta.msn.com </homepageUrl>
            <display>
                <name>Define with Encarta</name>
                <icon> http://encarta.msn.com/favicon.ico</icon>
            </display>
            <activity type="define">
                <activityAction context="selection" >
                    <execute

action="http://encarta.msn.com/dictionary_/{selection}.html"/>
                        <word:getThesaursus
                        url="
            http://thesaurus.reference.com/{selection}.html"/>
                </activityAction>
            </activity>
        </openServiceDescription>
```

[0044]   Notice in this example, a new namespace definition is provided near the top of

the XML representation, i.e. the second occurrence of the "xmlns" string. Further in the

XML representation, the thesaurus functionality is described in the <activityAction> tag.

[0045]   In this manner, existing services can be extended by describing the extensions

using the standardized format. Alternately or additionally, new services can be added as

described just below.

Adding a New Service

[0046]   In one or more embodiments, a new service can be added by creating a new

node as a child of the root node in the standardized format. In the example just below, a

new service is created by creating a namespace, i.e. the second occurrence of the "xmlns"

and then describing the new service as indicated. In this particular example, the service is described as a link service which allows a user to synchronize their bookmarks with a client machine.

[0047]   In the XML representation shown just below, notice that the common set of service properties is provided and everything after that in the representation pertains to the new service, i.e. the <link> tag.

```
<?xml version="1.0" encoding="UTF-8"?>
  <openServiceDescription
    xmlns="http://www.microsoft.com/schemas/openservicedesc
ription/1.0"
    xmlns:ext="http://link.service.org/schemas/linkservice/
1.0" >
    <homepageUrl>http://del.icio.us </homepageUrl>
    <display>
      <name>Add to Del.icio.us</name>
      <icon>http://http://del.icio.us/favicon.ico </icon>
      <description>Access your bookmarks from any computer!
Sync bookmarks with your computer</description>
    </display>
    <ext:link>
      <ext:tags>
        <ext:retrieve
url="https://api.del.icio.us/v1/tags/get"/>
        <ext:rename
url="https://api.del.icio.us/v1/tags/rename?"/>
      </ext:tags>
      <ext:retrieveBookmarks>
        <ext:retrieve                           type="all"
url="https://api.del.icio.us/v1/posts/all?">
          <ext:argument name="tag?"/>
        </ext:retrieve>
      </ext:retreiveBookmarks>
      <ext:editBookmarks>
        <ext:add
url="https://api.del.icio.us/v1/posts/add?">
          <ext:argument name="url"/>
          <ext:argument name="description"/>
```

```
                <ext:argument name="tags?"/>
            </ext:add>
            <ext:delete
    url="https://api.del.icio.us/v1/posts/delete?{url}"/>
            </ext:editBookmarks>
        </ext:link>
    </openServiceDescription>
```

Example Methods

[0048]   Figs. 2 and 3 are flow diagrams that describe steps in a method in accordance with one or more embodiments. The methods can be performed in connection with any suitable hardware, software, firmware, or combination thereof. In at least some embodiments, aspects of the methods can be implemented in connection with a Web browser. In the discussion that follows, the method described in Fig. 2 pertains to a method of describing services; and, the method described in Fig. 3 pertains to a method of discovering and providing services.

[0049]   Referring to Fig. 2, step 200 describes one or more services using common service properties. This step can be performed in any suitable way. For example, a service provider can provide a description of one or more of their services using a standardized format. An example of common service properties as well as a standardized format is provided above. Step 202 describes service-specific functionality for one or more services. The step can be performed in any suitable way. For example, a service provider can provide service-specific functionality descriptions for one or more services that they offer. Examples of how this can be done are provided above. Step 204 makes service descriptions available for one or more applications. The step can be performed in any suitable way. For example, once the service descriptions are built, the service descriptions can be provided into one or more files that can be downloaded to a client device. So, for example, when the user wishes to subscribe to a particular service, they

may click a particular user interface instrumentality to indicate that they wish to consume a service. Responsive to indicating an interest in a service, a service description such as that described above can be sent to the user's client device. Now, the user's application has a declarative definition of the service and can take steps, such as those described above, to implement the service's functionality. It is to be appreciated and understood that in at least some embodiments, receiving a service description such as those described above, and executing associated functionality can be performed independent of, and without downloading associated executable code.

[0050]   Referring to Fig. 3, step 300 receives one or more service descriptions. Service descriptions can be received in any suitable way. For example, a file that contains one or more service descriptions can be received. Examples of service descriptions are provided above. This step can be performed by any suitable application or other entity, examples of which include a Web browser. Step 302 ascertains, from the service descriptions, common service properties for one or more services. Examples of common service properties are provided above. Step 304 ascertains, from the service descriptions, service-specific functionality for one or more services. Examples of how this can be done are provided above. Step 306 provides one or more services based upon the service descriptions.

[0051]   In the illustrated and described embodiments, provision of services based upon the service descriptions can be provided by ascertaining user actions described in the service descriptions, and then responsively taking an action, also described in the service descriptions, based upon the user's action.

[0052]   Having described various embodiments above, consider now an example system that can be utilized to implement one or more of the embodiments.

Example System

[0053]   Fig. 4 illustrates an example computing device 400 that can implement the embodiments described above. Computing device 400 can be, for example, computing device 102 of Fig. 1 or any other suitable computing device.

[0054]   Computing device 400 includes one or more processors or processing units 402, one or more memory and/or storage components 404, one or more input/output (I/O) devices 406, and a bus 408 that allows the various components and devices to communicate with one another. Bus 408 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. Bus 408 can include wired and/or wireless buses.

[0055]   Memory/storage component 404 represents one or more computer storage media. Component 404 can include volatile media (such as random access memory (RAM)) and/or nonvolatile media (such as read only memory (ROM), Flash memory, optical disks, magnetic disks, and so forth). Component 404 can include fixed media (e.g., RAM, ROM, a fixed hard drive, etc.) as well as removable media (e.g., a Flash memory drive, a removable hard drive, an optical disk, and so forth).

[0056]   One or more input/output devices 406 allow a user to enter commands and information to computing device 400, and also allow information to be presented to the user and/or other components or devices. Examples of input devices include a keyboard, a cursor control device (e.g., a mouse), a microphone, a scanner, and so forth. Examples of output devices include a display device (e.g., a monitor or projector), speakers, a printer, a network card, and so forth.

[0057]   Various techniques may be described herein in the general context of software or program modules. Generally, software includes routines, programs, objects,

components, data structures, and so forth that perform particular tasks or implement

particular abstract data types. An implementation of these modules and techniques may be

stored on or transmitted across some form of computer readable media. Computer

readable media can be any available medium or media that can be accessed by a

computing device. By way of example, and not limitation, computer readable media may

comprise "computer storage media" and.

[0058]   "Computer storage media" include volatile and non-volatile, removable and

non-removable media implemented in any method or technology for storage of

information such as computer readable instructions, data structures, program modules, or

other data. Computer storage media include, but are not limited to, RAM, ROM,

EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks

(DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage

or other magnetic storage devices, or any other medium which can be used to store the

desired information and which can be accessed by a computer.

Conclusion

[0059]   Various embodiments provide a model through which service providers can

describe offered services using a standardized format. In one or more embodiments, the

standardized format is declarative and enables service providers to describe their

associated services in a standardized way. In at least some embodiments, the standardized

format includes a set of common service properties that are shared across multiple

different services. Additionally, service specific properties can be described in addition to

the common service properties. In one or more embodiments, existing services can be

extended by including, in the standardized format, a description of an extension. In one or

more other embodiments, new services can be added for consumption by simply including a description of the new service using the standardized format.

[0060]   In at least some embodiments, the standardized format is represented through a declarative, hierarchical tag-based language. By way of example and not limitation, a suitable hierarchical text-based language is XML. It is to be appreciated and understood, however, that other methods and ways of representing the standardized format can be utilized without departing from the spirit and scope of the claimed subject matter.

[0061]   Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed invention.

CLAIMS

1.   A computer-implemented method comprising:

describing (200), using a standardized format, common service properties associated with multiple different services;

describing (202), using the standardized format, specific functionality associated with one or more services; and

making (204) service descriptions available for one or more applications or other entities.

2.   The method of claim 1, wherein the standardized format comprises XML.

3.   The method of claim 1, wherein the act of making is performed by making the service descriptions available for a Web browser.

4.   The method of claim 1, wherein the common service properties are shared across multiple different services.

5.   The method of claim 1, wherein the common service properties include a property associated with a homepage URL and a property associated with a display for a service.

6.   The method of claim 1, wherein the common service properties include a property associated with a homepage URL and a property associated with a display for a service, wherein the property associated with the display for a service comprises properties associated with a name, an icon, and a description.

7.   The method of claim 1, wherein the standardized format uses at least one namespace to describe said one or more services.

8.   The method of claim 1, wherein said standardized format is used to extend an existing service.

9.   The method of claim 1, wherein said standardized format is used to add a new service.

10.   A computer-implemented method comprising:

receiving (300) one or more service descriptions, the service descriptions being represented in a standardized format;

ascertaining (302), from the service descriptions, common service properties for one or more services;

ascertaining (304), from the service descriptions, service-specific functionality for one or more services; and

providing (306) one or more services based upon the service descriptions.

11.   The method of claim 10, wherein the standardized format comprises XML.

12. The method of claim 10, wherein the standardized format uses at least one namespace to describe said one or more services.

13. The method of claim 10, wherein said acts of receiving one or more service descriptions, ascertaining common service properties, ascertaining service-specific functionality, and providing one or more services are performed by a Web browser.

14. The method of claim 10, wherein the common service properties are shared across multiple different services.

15. The method of claim 10, wherein the common service properties include a property associated with a homepage URL and a property associated with a display for the service, wherein the homepage URL is used to identify a service.

16. The method of claim 10, wherein the common service properties include a property associated with a homepage URL and a property associated with a display for the service, wherein the property associated with the display for a service comprises properties associated with a name, an icon, and a description.

17. The method of claim 10, wherein said standardized format is used to extend an existing service or add a new service.

18. The method of claim 10 further comprising supporting only a portion of the standardized format.

24

19.   A system comprising:

one or more computer readable storage media (106);

computer-readable instructions on the one or more computer readable storage media which, when executed, provide a Web browser (110) configured to:

receive one or more service descriptions, the service descriptions being represented in a standardized format expressed in XML;

ascertain, from the service descriptions, common service properties for one or more services, wherein the common service properties include a property associated with a homepage URL and a property associated with a display for a service, wherein the property associated with the display for a service comprises properties associated with a name, an icon, and a description;

ascertain, from the service descriptions, service-specific functionality for one or more services; and

provide one or more services based upon the service descriptions;

wherein said standardized format can be used to extend an existing service and/or to add a new service; and

wherein the standardized format uses at least one namespace to describe said one or more services.

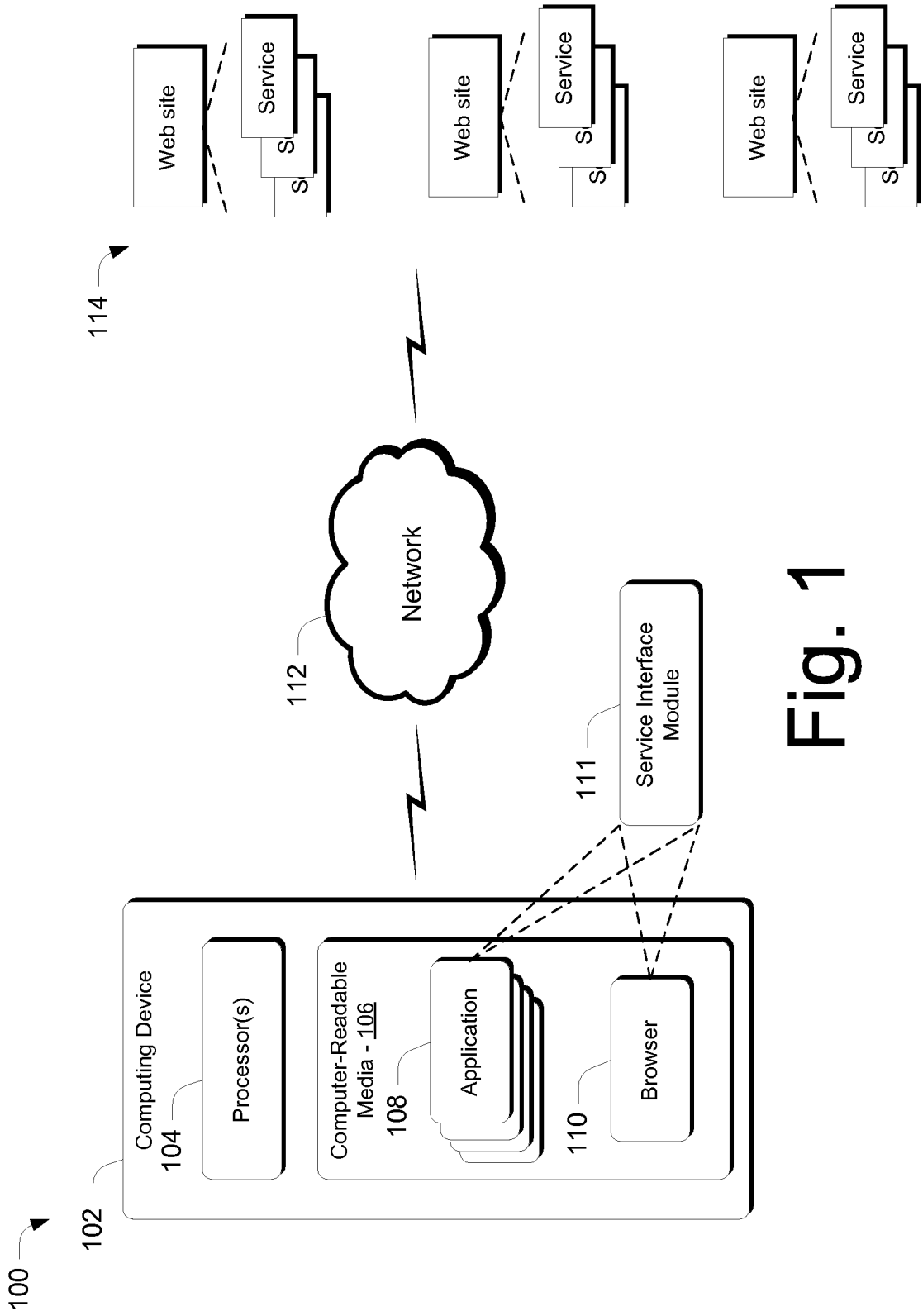20.   The method of claim 19, wherein the common service properties are shared across multiple different services.
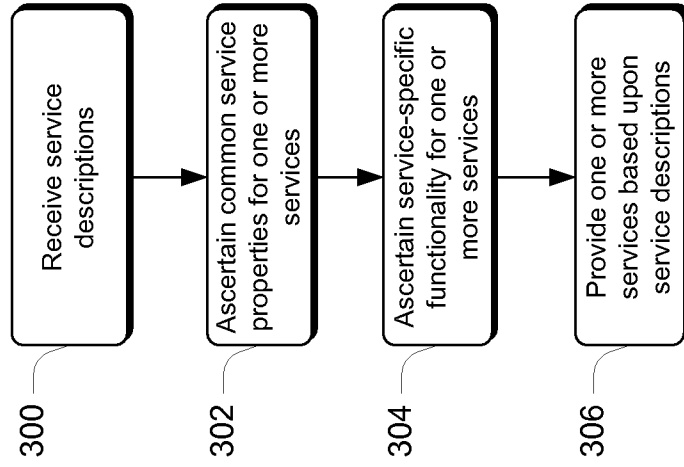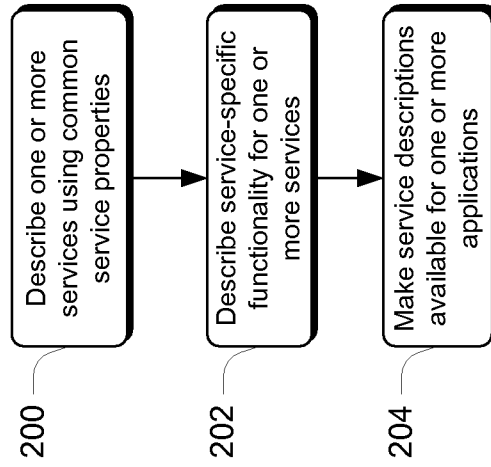
Fig. 1

Fig. 3

300 — Receive service descriptions

302 — Ascertain common service properties for one or more services

304 — Ascertain service-specific functionality for one or more services

306 — Provide one or more services based upon service descriptions

Fig. 2

200 — Describe one or more services using common service properties

202 — Describe service-specific functionality for one or more services

204 — Make service descriptions available for one or more applications
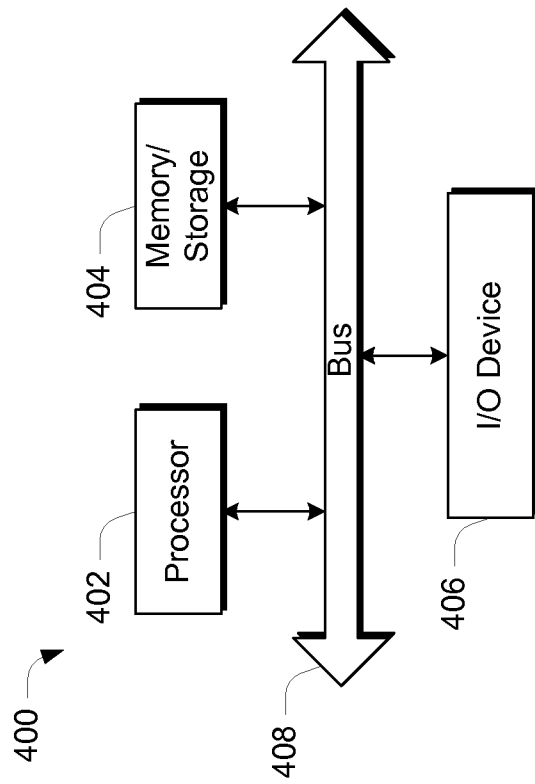
Fig. 4

A.   CLASSIFICATION OF SUBJECT MATTER

G06F 17/00(2006.01)i, G06F 17/21(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 8 : G06F 13/00, G06F 15/00, 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)
  eKOMPASS(KIPO internal) : "(web or internet) and (XML or markup) and extensible"

C.   DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X ------------ A	US 2004-0068554 A1 (Christopher E. Bales et al.) 2004.04.08. see abstract ; paragraphs [0038]~[0042], [0074] ; figs.4~7, figs.14~15.	1~2,5~7,10~12,15~16 ------------------------- 3~4,8~9,13~14,17~20
A	US 2007-0011605 A1 (Donald Dumitru et al.) 2007.01.11. see abstract ; paragraphs [0029]~[0041] ; figs.2~4.	1~20
A	US 2007-0174143 A1 (Shrage Smilowitz et al.) 2007.07.26. see abstract ; paragraphs [0019]~[0021], [0064]~[0066] ; figs.5~6.	1~20

☐ 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 "&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 AUGUST 2009 (17.08.2009)	**18 AUGUST 2009 (18.08.2009)**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office Government Complex-Daejeon, 139 Seonsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea	KIM, SANG CHEOL
Facsimile No. 82-42-472-7140	Telephone No. 82-42-481-8521

Form PCT/ISA/210 (second sheet) (July 2008)

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004-0068554 A1	08.04.2004	WO 2003-093964 A1	13.11.2003
US 2007-0011605 A1	11.01.1007	None	
US 2007-0174143 A1	26.07.2007	US 2007-180055 A1 WO 2007-087264 A2	02.08.2007 02.08.2007