A Novel Approch for Mobile Devices in Accessing Web Services

Dinesh Krishna Andhalkar
Sidhhanth College of Engineering
Savitribai Phule Pune University, Pune,
India

R. W. Deshpande
Sidhhanth College of Engineering
Savitribai Phule Pune University, Pune,
India

S. A. Nalawade
Sidhhanth College of Engineering
Savitribai Phule Pune University, Pune,
India

ABSTRACT

In Recent years Mobile devices & Web Services have become highly popular . It is a challenging task to host the web services in mobile devices over a wireless network, Due to different limitations of mobile devices like limited battery, memory space, and low processor speed. So it's becoming necessary to integrate web services into applications that run on these mobile devices. We are introducing an efficient proxy based approach which overwhelms technical challenges involved in the dynamic discovery of web services & generates client side proxies at runtime. We describe an effective architecture which shows how mobile devices dynamically discovers & invocate web services by limited intervention with network, service client just need to enter search phrase & values for method parameters on the dynamically generated GUI.

Keywords

Mobile Devices, Mobile computing, Proxy Server, Web Services.

1. INTRODUCTION

In today's era smart mobile phones are not just a small device used for communication, but also it is used for accessing web services from anywhere, anytime. Different applications are communicating over a wireless network to the base of web services. Accessing web services [1] from mobile devices, the mobile devices perform the role of a service consumer as well as a service provider. In terms of more portability & location transparency, it is necessary to integrate both mobile computing & web service technology into an application that run on mobile devices.

1.1 FRAMEWORK FOR MOBILE WEB SERVICE

Basically, this framework having three components: a Service Requestor (Client), Service Provider (Server) and Service Registry (UDDI). The service requestor accesses the described Web Service using SOAP [6], the service requestor Searches the UDDI registry for the services [7], and the UDDI compatible service registry refers the respective WSDL [8], The service provider publishes its Web Services with the service registry as shown in above fig.

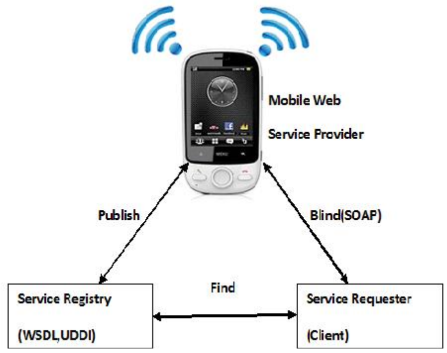

Fig 1.1 Architectural setup of Mobile Host

There are two solutions while accessing web services from mobile devices, first is, parsing of WSDL file by the mobile device application that will directly interact with web services & Second is build proxy that is compiled assembly from a WSDL file at runtime, use it while interacting with services. For parsing WSDL file from a mobile device, we need a compiler. But there is no such commercial compiler has been developed for such devices.

As compared to desktops, Mobile devices have different limitations like lower processing power, limited bandwidth, less memory, and finite battery power. Sometime during the discovery of web services, wireless network may have failures due to which there is an impediment in the completion of user request. Due to this difficulty user not uses mobile devices frequently for accessing web services.

All of the above mentioned issues & technical challenges led to the architectural configuration proposed in this paper which represent a proxy server called mobile web servers who act as a gateway between mobile devices & web server. This server will do the parsing and build proxy for the service requested by the user. This process of building proxy by the local server and send it to the mobile device looks like a local process and it is available for the mobile whenever it is needed. Most of the workload moves to the proxy server, due to which mobile devices are released from energy & time consuming tasks such as communicating with internet server & parsing of WSDL file.

2. RELATED WORK

Mainly there are two approach’s for accessing web services from mobile devices SOAP & the REST approach [4]. In [4] SOAP-based approach, the client side proxy application makes a procedure call that resembles a local call. As an outcome, clients can invoke Web services by method call that is in the form of the SOAP message, this message sent to a web server in which requested web service is available. The outcome of the execution of a procedure call is sent in the form of SOAP message to the requested client side proxy. The proxy is generated by producing a source file from a WSDL, which is present in UDDI registries. Then generated source file is compiled into a proxy class. Finally, this proxy class is registered with the client application.

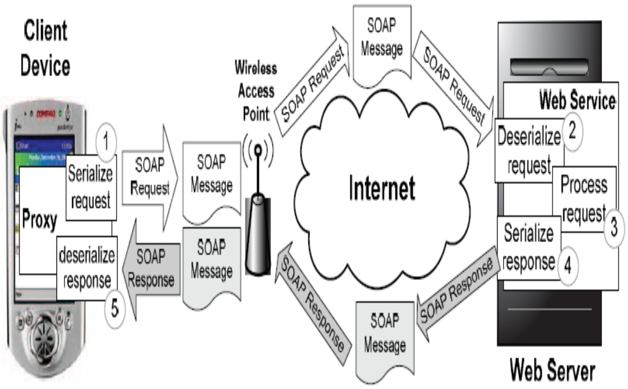

Fig 2.1 A SOAP based architecture for mobile in accessing web services

In REST [4] based approach, REST stands for Representational State Transfer is relying on a single application protocol (HTTP), universal resource indicators (URI) and standardized data formats, through XML. It employs established HTTP methods such as GET and POST to direct applications. So instead of creating a standard, machine-readable way for applications to discover and use application components on remote systems - the way SOAP is being used for Web services - REST developers use URIs to create a common ground so applications can use HTTP and XML to share data.

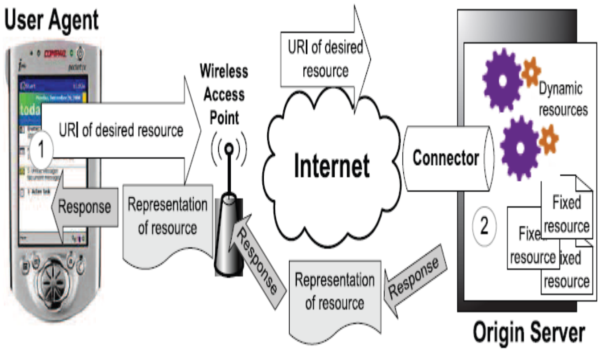

Fig 2.1 A REST based architecture for mobile in accessing web services

Several researchers have developed different frameworks & architectures for dynamic discovery & invocation of web services in mobile devices. Power et al. [9] Discuss concisely different mobile service provisioning approaches & categories it into Peer-to-Peer, asymmetric infrastructure and proxy server based. In a P2P-based approach, mobile devices act as a Web service provider, whereas any Web client can consume these services either it is a mobile or a stationary client. P2P provisioning, P2P network advertising mechanism is used to publish and discover Web service & it is handled using JXTA [2] as shown in fig. For the dynamic discovery of Web services, Web clients having peer ID for that it must join the P2P network. By using underlying JXTA protocol mapping between the client IP and the Peer ID is done. After successfully connected to the JXTA network, Clients can query/search the network for the required services. Once the required service is located, through JXTA pipes client communicates with the service provider in order to send and receive messages.

Finally, in the work where intermediate server or proxy server is used in between the requestor and the service provider which does the maximum work of generation of proxy class by the process of text matching with the cached web service descriptions that are sent periodically from the UDDI registries by periodical requests from the proxy server. If the requested service is not present in the cached files then the server will send request for URLs to the web server and the web server will send the WSDL files. With these files the matching file is selected and the source code is generated. Then the source code is compiled by using libraries and build the client side proxy and then it is shipped to the mobile device. This client side proxy is responsible for generating the GUI dynamically which provides the service results requested by user in the mobile device.

3. PROPOSED WORK

In the proposed architecture, WAP & WML [14] [15] two essential standards for mobile web services are used. WAP [14] is the open standard, defined & coordinated by WAP forum that enables mobile users to easily access and interact with information and services from their wireless devices. It is compatible with any operating system and with most well-known wireless networks. The main purpose of WML [15] is to provide content and user interface for specific wireless devices that have different constraint like small displays, limited user input capabilities, bandwidth restrictions, limited memory & computational resources, limited browsing capabilities etc. [1] [3] [9].

In the proposed architecture, we have service client requestor, proxy server & web server. In the next section we discuss process execution at each of the components.

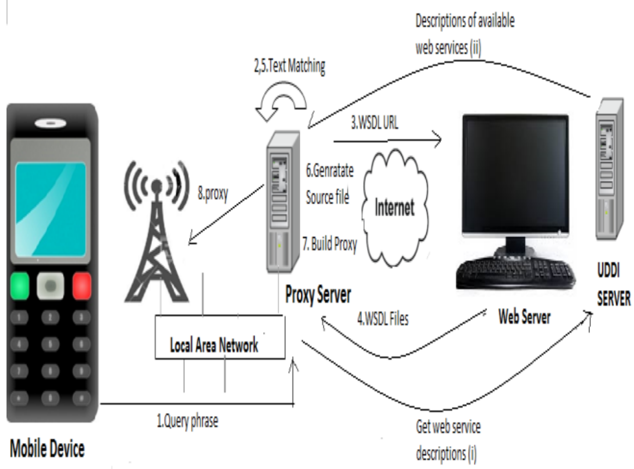

Fig 3. An Overview of proxy server architecture for mobile in accessing web services

3.1 Service Client Process

Service clients are mobile phones-oriented users interested in diverse services such as newspaper reports, weather forecast, airport services, mobile shopping, mobile banking and m-government services. In this process service client or mobile device is connected to the proxy server instead of directly connecting to the internet. This process will reduce the disadvantages of mobile accessing the internet. Here user sends request or query phrase to the proxy server. If the requested service is present in the cache of a proxy server, it immediately gives response to service client process, if requested service is not present in the cached files, then the proxy server will send request for URLs to the web server and in response the web server will send the WSDL files. With these files the matching file is selected and the source code is generated. Then the source code is compiled by using libraries and build the client side proxy and then it is pushed to the mobile device. This client side proxy is used in the process of generation of dynamic GUI which gives the service results requested by users in the mobile device.

3.2 Proxy Server Process

Mainly Proxy server does three processes such as Text matching, Generating source files, and Build proxy. Text matching has two purposes, first to generate a short list of candidate web services based on matching their cached short descriptions with the user's supplied search string, and second, to identify the most appropriate web service among those short listed based on matching the method's description found in their downloaded WSDL files with the user's string. In the second process is the WSDL File Downloader which takes URIs of the shorted-listed services as input, and downloads the files if they are not in the cache from UDDI server. Finally, the third main process is the Proxy Builder, which generates a client-side proxy class and sends it to the mobile device through FTP.

3.3 Web Server Process

At the web server, it receives a WSDL URI from a proxy server in response web server sends the WSDL file.

3.4 UDDI Server

UDDI server has service directory can be used by mobile users to locate new services. Discovery is computed at runtime by the proxy server, once the user has sent their request of new services at UDDI server.

4. MODULES AND DESCRIPTION

The proposed system contains two modules:

4.1 Discovering Web Methods

There have been several UDDI registries available on the Internet in order to offer APIs for locating desired services. For evaluation, we restricted ourselves to the private XMethods registry, which contains methods that allow for service summary structures for all active services listed at X methods. In the local Proxy server background process calls this method and caches the discovers service descriptions locally in a simple SQL SERVER database table with these service descriptions the proxy server will generate a proxy class.

The mobile application binds to a proxy class that interfaces to the proxy server's developed web service. In this paper, we have a certain type of services such as information regarding barcode services, mail services. The service descriptions regarding these services are retrieved from the internet and they are stored in the database table. Whenever the user requests regarding barcode or mail services, the proxy server will retrieve information according to the request search string provided by the user. If the information is not present, the user is asked to specify the search phrase little more accurately. It is through this "remote" method that the user submits his search query. At the proxy server, the query is matched against the cached descriptions of the web services using the Boyer-Moore algorithm, which is a fast string searching algorithm. This algorithm is suitable for this type of applications as it works the fastest when the alphabet is moderately sized and the pattern is relatively long. The outcome of this search is one web method that best matches the input phrase.

In the searching process, if the WSDL file is not present in the cache of a request for the service is searched will forward to the web server. After that the WSDL files are parsed in order to get the methods of the web service and their corresponding documentation. Initially the XML file is searched for the operation name (name of the web method) and the documentation. This web method is used for further process of proxy generation by the proxy server.

4.2 Proxy Class Generation and Compilation

After discovering the web methods, the proxy (dll) is built from its WSDL file. This task involves generating a class (source code), compiling it, and then sending it to the mobile device where it is used for direct method invocation. All the involved code runs on the proxy server, obviously except the mobile application and the client proxy after being sent to the device. Basically the URL of the discovered web service by the proxy server is used to get the service details. More specially the server retrieves the service description & schemas & saves them in a list, & then performs a set of steps to generate instances of classes through which the code in the service's WSDL file is programmatically generated. After generating the C# class, the assembly file (compiled proxy) is created through a set of additional steps that uses compiler parameters and assemblies specific to mobile devices.

5. CONCLUSION

Our architecture provides web service discovery services to personal applications running on mobile devices. As a future work, a proxy server can be programmed with the intelligence which identify a set of web service entities that performs computationally intensive tasks, as in multimedia processing, bioinformatics & data mining along with this how to connect and coordinate them is also a challenging task that is not suited for mobile devices. Extending the participation in the wireless network, the implemented solution overcomes technical limitations, and also saves device battery power. The presented design makes it possible for mobile device users to dynamically invoke web service methods.

6. REFERENCES

[1] B.R. Javani., S. L. Bajeja, V. C. Gondalia April 2012, "An Architectural Model For Mobile Based Web Services," International Journal Of Computer Application Issue2, Volume 2

[2] Srirama S. 2006, "Publishing and Discovery of Mobile Web Services in Peer to Peer Networks". International Workshop on Mobile Services and Personalized Environments (MSPE'06), Aachen, Germany.

[3] Alin COBÂRZAN: “Consuming Web Services on Mobile Platforms”, Informatica Economică vol. 14, no. 3/2010.

[4] AlShahwan F.,Moessner K. May 2010, "Providing SOAP Web Services and RESTful Web Services from Mobile Hosts", International Conference on Internet and Web Applications and Services, pp.174 - 179, 9.

[5] C.Weyer “DynwLib”, 2010, Available at,: <http://weblogs.thinktecture.com/cweyer /2006/11/from-dynwslib-to-proxyfactory. html>, [Accessed 25 January 2014].

[6] ”Simple Object Access Protocol” (SOAP).2010, Available at, <http:// www .w3.org/TR/soap12 part1/> [Accessed 10 September 2013].

[7] ”Universal Description, Discovery and Integration.”(UDDI), 2010, Available at <http: //uddi. org/ pubs/ uddi-v3.0.2-20041019 .htm> [Accessed 10 September 2013]

[8] “Web Services Description Language” (WSDL), 2012, Available at,<http:// www.w3.org /TR/wsdl20/>[Accessed 10 September 2013]

[9] Khalid Elgazzar, Patrick Martin, Hossam S. Hassanein 2012, “Enabling Mobile Web Service Provisioning”, Technical Report 598

[10] “Xforms-The Next Generation of webforms”, 2012 Available at <http://www. w3. com/ Mark Up/Form> [Accessed 20 January 2014].

[11] ”VoiceXML”, 2012, Available at <http: //www. vxml.org/>, [Accessed 20 January 2014].

[12] R. Steele, K. Khankan, and T. Dillon, 2005,“Mobile Web Services Discovery and Invocation through Auto-Generation of Abstract Multimodal Interface,” Proc. Int’l Conf. Information Technology: Coding and Computing (ITCC ’05), vol. 2, pp. 35-41.

[13] Yeon-Seok Kim and Kyong-Ho Lee. 2007, “A Light-weight Framework for Hosting, Web Services on Mobile Devices”, IEEE European Conference on Web Services, pp. 255-263.

[14] “Wireless Application Protocol” White Paper, 2010, available at :< http:// www .Wapforum.org/what/ WAP_white_pages. pdf.> [Accessed 17 January 2014].

[15] ”Wireless Markup language” Specification Version1.1, (WML), 2011, available at <http: // www. Wapforum.org/ what/ technical.htm> [Accessed 12 December 2013].