

Research on Function Enhancement of WAP Gateway

MING-ZHI CHEN* WEI-JIE CHEN♦ SHUI-LI CHEN□ LUN YU*

*.College of Physics and Information Engineering, Fuzhou University, Fuzhou, 350002, CHINA

♦.Xiamen Development Center of China Construction Bank, Xiamen,361000, CHINA

□.School of Science, Jimei University, Xiamen, Fujian 361021, CHINA

E-mail: donres@fzu.edu.cn, chenweijie.zh@ccb.cn, shuilichen@vip.sina.com, yulun@fzu.edu.cn

Abstract: - In this paper, some disadvantages of the WAP mobile devices, such as bad arithmetic capacity, small memory and worm-like speed of network transmission, could be bypassed, while the functions of traditional WAP (Wireless Application Protocol) gateway are enhanced to automatically create the WEB services client in terms of the requests of WSDL document from the mobile internet user. And then, by the WAP gateway, the client WML documents are compressed and transmitted to WAP mobile device for displaying. Those achieve the fast coordination of dynamic E-business and facilitate the general mobile users' enjoying the WEB services.

Key-Words: - WAP Gateway, WEB services, WSDL, SOAP

1 Introduction

The emergence of the technology of WEB services has made the gradual transfer from the traditional electronic commerce (EC) to dynamic EC. The dynamic EC refers to a new electronic commerce flow and relevant system that could be dynamically adapted to everchangeful operation in company, which reflects the concept of dynamic integration of application on the condition of real time and realizes the real value of EC. So the technology foundation of dynamic EC involves the WEB services.

Along with the prevalence of the WAP mobile device, the descent of expense of WAP on line and the speed-up of the network, more and more general users have been WAP mobile users. Those ignite the increasing request of WAP users for the network function of WAP mobile, while the users could not settle for browsing static WML page by WAP mobile network; in addition, they thirst for as much diversified services through mobile network as wired network, especially, enjoying the convenience brought from the WEB services on their own mobile terminal equipment. On this condition, the best solution to this problem is to enhance the function of the WAP mobile network, that is, to enhance the function of the WAP gateways---enable them to support the WEB services.

2. Web Services and Relevant Technology

2.1 WEB Services

The architecture of WEB services is based on the interaction of three roles, services provider, services registration center and services requestor. And the interaction involves such operation as release, retrieval and binding. Generally, services provider provides the accessible software module through network (namely, a performance of WEB services) and describes the content of the WEB services. And then provider issues them to the services requestors or services registration center. Services requestor retrieves the services description from local or registration center by means of the retrieval operation and then binds services provider together with the description of services to call the WEB services to interact with provider. There is an interchangeable logical relationship between the services provider and requestor. The following figure shows it.

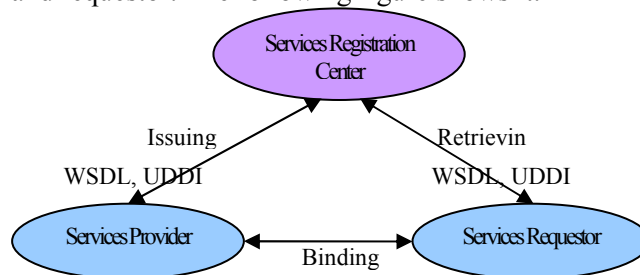


Figure1. WEB services model

2.2 Simple Object Access Protocol (SOAP)

The main arm of Simple Object Access Protocol (SOAP) is to call the remote COM objects by the protocol of HTTP to break the limit of the network

and firewall so as to empower the COM. The SOAP presents a simple, lightweight mechanism to peer-to-peer exchange structured, typical information within loosely coupled distributed environment in the form of XML. So a message of SOAP is a simple XML document, which consists of such elements as SOAP envelope, SOAP Encoding and RPC Presentation.

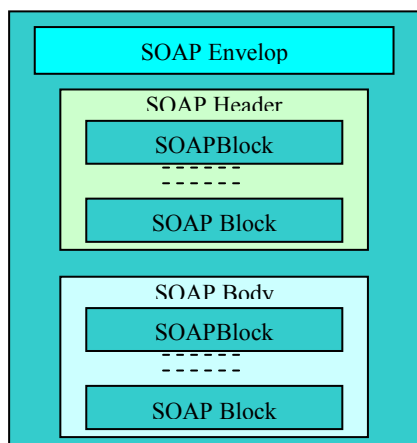


Figure2. SOAP message format

2.3 Description Language of WEB Services

Web Services Description Language (WSDL) is a XML document format to describe the WEB services and to define WEB services as a communication endpoint set to exchange messages. The outstanding point of WSDL is to separate the abstract definition of services from concrete implementation.

2.4 Universal Description, Discovery and Integration (UDDI)

Universal Description, Discovery and Integration (UDDI) are a set of standard specification based on WEB to provide info registration center to WEB services. It is also a set of access protocols to enable enterprises to provide WEB services registration available for others. UDDI presents a method of business registration center based on distribution.

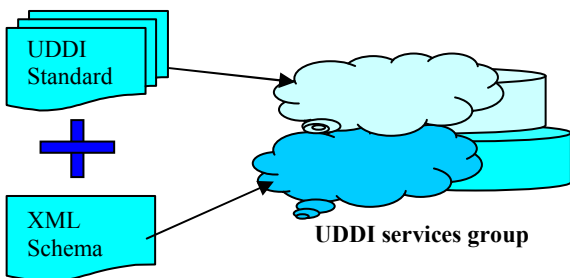


Figure3. Relationship of UDDI standard, XML Schema and UDDI services group

3. Introduction to WAP

Wireless Application Protocol (WAP), which combines Internet technology and wireless network technology, is a cluster of wireless application protocols based on Internet. The WAP defines an open global wireless application framework and a series of network protocols so as to import the application and services based on Internet into wireless terminal like mobile phone. The most fascinating thing of WAP is to make anyone speedily get information whenever and wherever, in addition, portable and easy-use features of WAP terminal. The electronic commerce is the best place for application of WAP such as stock trading by handset, banking by handset.

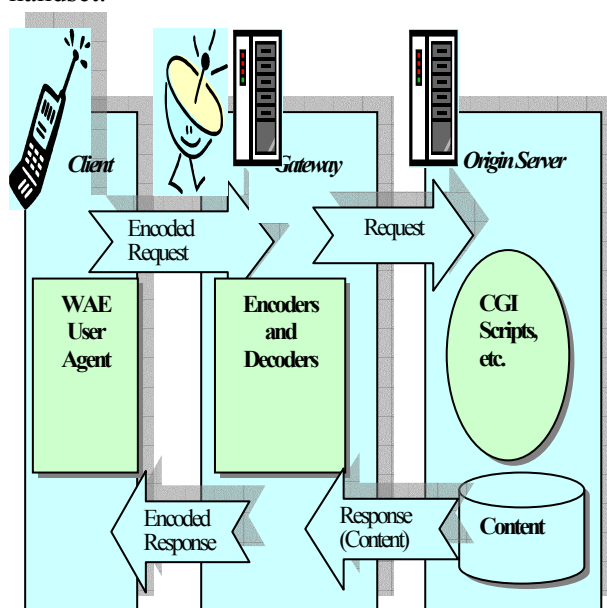


Figure4. WAP architecture

WAP gateway is the core product of application of WAP. WAP gateway mainly builds a bridge between mobile device and Internet server to enable interchange between WAP protocol stack and traditional stack of TCP/IP so that the wireless terminals with function of WAP browse can access the resources on Internet.

In terms of the two factors, open of source codes and extensibility, the Kannel gateway is selected to do an enhancement of function. The overall design of Kannel gateway strictly conforms to the protocol standards of WAP, which consists of three modules (namely, management module, load-bearing module and module of WAP protocol stack processing).

4. Enhancement of WAP Gateway

4.1 WAP Gateway with Supporting WEB Services

In order to want WAP mobile users to conveniently use current resources of WEB services, it is necessary to enhance the function of WAP gateway, which enable WAP gateway to automatically analyze the WSDL document, WEB services interface, and then generate the client of WEB services. Certainly, it is requested that such new additive function be imported in the form of module and be compatibly combined with original functional module of Kannel WAP gateway. By means of enhancement of function of traditional Kannel WAP gateway, we can speed up the progress of using WEB services for the users of wireless terminal and also save plentiful amount of work for coding WAP client of WEB services.

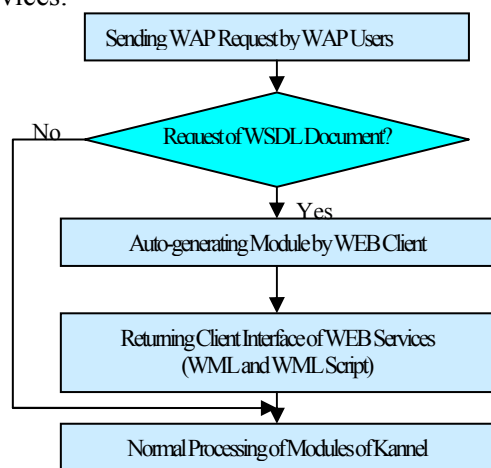


Figure5. Structure of expanded Kannel gateway
The point of difference the expanded WAP gateway from the traditional one lies in that the former need to judge the type of WAP request from WAP mobile users. If the type of WAP request is of WSDL document, the WEB services client could be activated to automatically generate a module to process those requests.

4.2 Automatic Generation of Client of WEB Services

4.2.1 Principle of Auto-generation of Client of WEB Services

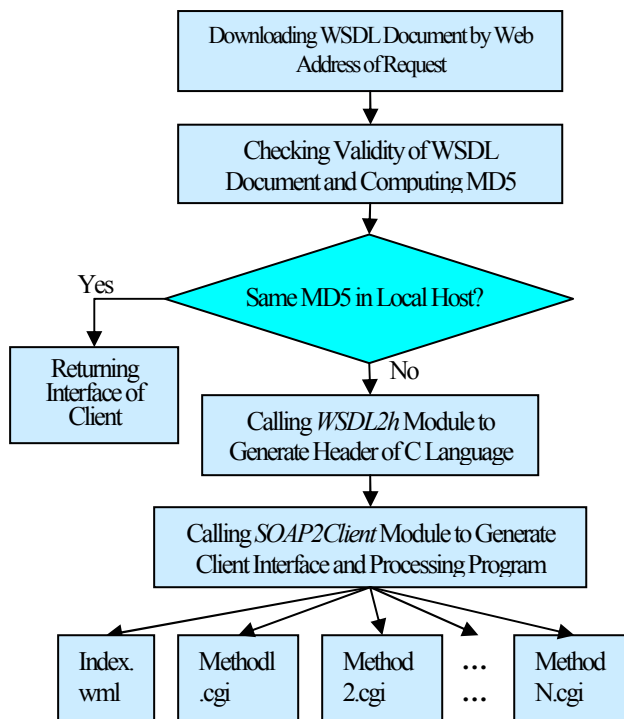


Figure6. Flow chart of auto-generation of WEB services client

The file, *index.wml*, consists of a main page and some pages of WEB method. Each WEB services method correspond to certain file of *Method?.cgi*, which is to process the input of mobile users and to generate the files of WML and *WMLScript* containing the result of calling of WEB services.

4.2.2 Processing for WSDL Document

WSDL document plays an important role in WEB services. The user's interface of auto-generator of WEB services client can automatically adjust himself by means of the content of WSDL document. Before that, the elements in WSDL document should be correctly analyzed in advance.

There are cross-referential, cross-inclusive relationships among the some elements within WSDL document.

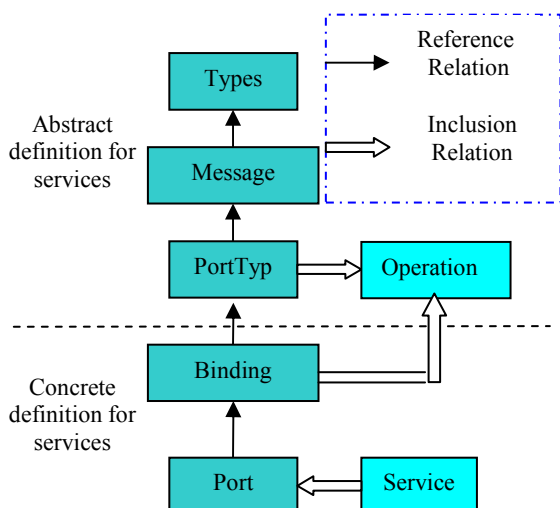


Figure7. Graph of relation of elements in WSDL document

The WSDL document is split into two parts: abstract definition of services and concrete definition of services. The Message refers to the abstract description for the exchange data. And the PortType be said to be the abstract set of operation. The analysis for WSDL document is equivalent to describing the content of WSDL document in programming language, which can be archived by adopting C++ classes.

WSDL document could correspond to the following data structure when the gSOAP is adopted to analyzed the WSDL document.

```

class wsdl_definitions
{
public:
std::vector<wsdl_import>    import;
//member variable of types corresponding to the
//element in WSDL document.
wsdl_types                *types;
//member variable of message corresponding
//message's element in WSDL document.
std::vector<wsdl_message>  message;
//member variable corresponding to the portType's e
//in WSDL document.
std::vector<wsdl_portType>  portType;
std::vector<wsdl_binding>   binding;
std::vector<wsdl_service>   service;
.....
}
    
```

Providing the net address of WSDL document or local-storage address is given, the traversing for WSDL document could be gotten by virtue of gSOAP.

4.2.3 Implementation of WEB Services' Method

The calling for WEB services is equivalent to constructing and sending of SOAP message; and to return result of WEB services is to receive and analyze the SOAP message. In this paper, according to the description of WSDL document, the WEB service has tow WEB methods: *AddMethod* and *SubMethod*. The following request message of SOAP is generated by calling WEB services method, *AddMethod*.

```

POST /WebService1/Service1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: actual length of SOAP message
SOAPAction: "http://tempuri.org/AddMethod"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema
a-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema
a"
xmlns:soap="http://schemas.xmlsoap.org/soap/en
velope/">
  <soap:Body>
    <AddMethod xmlns="http://tempuri.org/">
      <num1>parameter1</num1>
      <num2>parameter2</num2>
    </AddMethod>
  </soap:Body>
</soap:Envelope>
    
```

```

The SOAP message returned from the WEB services
method (AddMethod)
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: actual length of SOAP message
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema
"
xmlns:soap="http://schemas.xmlsoap.org/soap/env
elope/">
  <soap:Body>
    <AddMethodResponse
xmlns="http://tempuri.org/">
      <AddMethodResult> result </AddMethodResult>
    </AddMethodResponse>
  </soap:Body>
</soap:Envelope>
    
```

4.2.4 Generation of the Client Interface

In order to facilitate the mobile users to use WEB services, a friendly interface to display the WEB services method is of an absolute necessary. The development tool of gSOAP, however, only provides

such functions as analysis of WSDL document, binding of compound data type and code implementation of WEB services method without any generation tool for interface. So we program WML file(Deck) for auto-generating the client interface of WEB services in terms of WSDL document and CGI file for processing call of WEB services method and return of result etc..

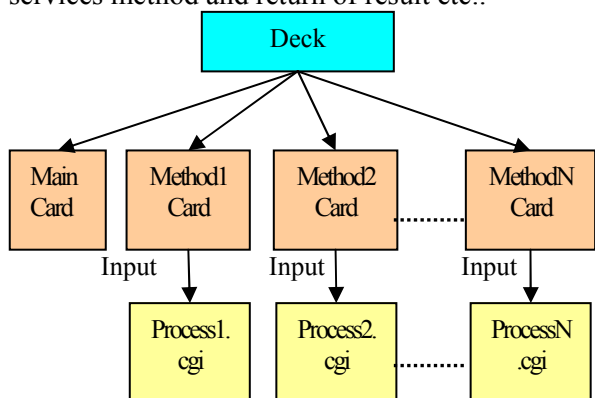


Figure8. Basic documents contained in client of WEB services

The WML file (Deck) has some cards as follows.
Main Card: to display the interface of all WEB services methods defined by WSDL document;
Method? Cards: to display the interface containing input parameters and calling buttons of corresponding WEB services methods;
Process?.cgi, a series of files, to process the users' input parameters passed from corresponding Method? Card, which calls the methods of WEB services and then waiting the results returned from WEB services method to display them to wireless users.



Figure9. Interface of client of WEB services

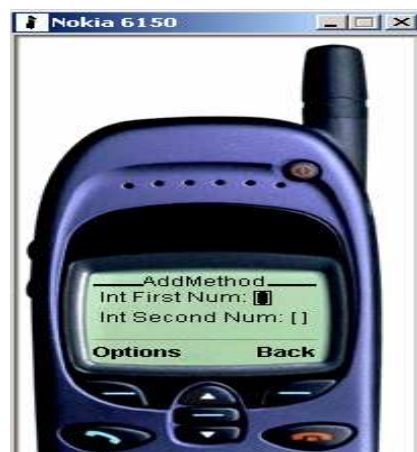


Figure10. Operation interface of methods of WEB services

4.2.5 IO Handler for WAP Users

In this paper, CGI(Common Gateway Interface) is adopted to dynamically generate WML pages because that the script program of CGI and WEB services method generated by gSOAP all are carried out in C/C++ language, which can make CGI program seamlessly call the method of WEB services. The procedure of input data processing of WAP users and result returning is showed as follows.

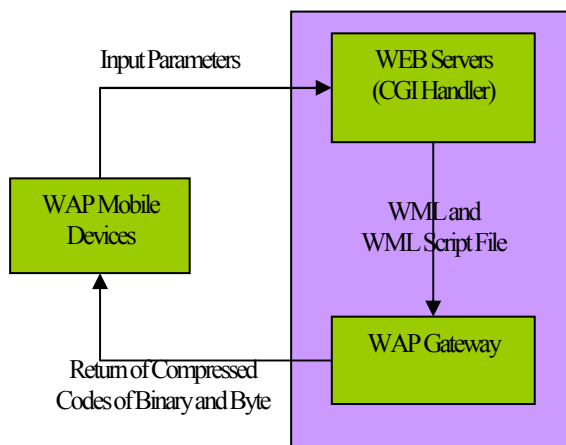


Figure11. Flow chart of data IO process

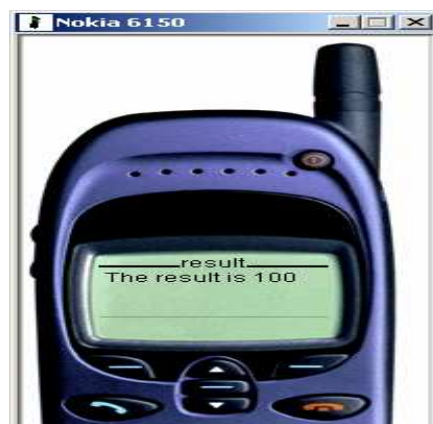


Figure12. Display of result of WEB services

5. Conclusion

In this paper, the model of WAP gateway with supporting WEB services (auto-generation of client-side) was presented. And then the detailed designs of this model were carried out. Finally, some results of experiment show that this model can run well. Due to some factors, however, this model is inadequate sophisticated and has some defects such as display of Chinese, a little long time for auto-generation of client of WEB services, which is mostly spent in analyzing for WSDL document; although the computation and comparison of MD5 are put out to reduce the auto-generation of client of WEB services. Maybe the only way to solve this is to speed up the analysis for WSDL document by means of better algorithm.

Acknowledgement

Corresponding Author □ Prof. Lun Yu

The authors are grateful to the anonymous referees for their valuable comments concerning the way the paper could be improved.

The project was supported by the National Natural Science Foundation of China (No. 60372070), the Natural Science Foundation of Fujian Province of China (A0510023)

References:

- [1] Chai Xiaolu. *School and Gest—Nes strategy of Web Services*. http://tech.ccidnet.com/art/289/20030417/43926_1.html. 2003.04
- [2] Chai Xiaolu, Lian Yuqi. *Technology, Framework and Application of Web Services*. Beijing: Electronic Industry Press, Jan, 2003
- [3] Elite Science and Technology. *Exposing Secrete of Technology of WAP and WML*. China Electrical Power Publisher, Apr. 2001
- [4] Arehart. *Manual of WAP Programming*. Electronic Industry Press, Jan. 2001
- [5] Hou Chunping. *Wireless Application Protocol*. Engineering Industry Publisher, Jan. 2000
- [6] Liu Wei. *Overall Contact with CGI Technology*. Beijing: Tsing Hua University Press, Jan. 2001
- [7] Guo Mingzhuang. *Research and implementation of Model of Auto-Generator of WEB Services Client*. Master Thesis of Fuzhou University, Dec. 2003
- [8] Chai Xiaolu. *WSDL: Description of Your Web Services*.

- <http://www-900.ibm.com/developerWorks/cn/webservices/ws-wsdl/index.shtml>
- [9] Robert van Engelen. *gSOAP 2.6.2 User Guide*. <http://www.cs.fsu.edu/~engelen/soap.html>
- [10] Andreas Fink. *Kannel 1.3.2 User's Guide*. <http://www.kannel.org/download/1.3.2/userguide-1.3.2/userguide.html>
- [11] Lars Wirzenius. *Kannel Architecture and Design*. <http://www.kannel.org/download/1.3.2/arch-1.3.2/arch.html>
- [12] Ryan Kanno, Aaron Klish. *Investigating the Architecture of a WAP / SMS Gateway Server*. <http://www.localkinegrinds.com/kannel/index.html>
- [13] Dan Gisolfi. *Web services architect*. http://www-900.ibm.com/developerWorks/cn/webservices/ws-arc/part3/index_eng.shtml
- [14] W3C. *Web Services Description Language (WSDL) 1.1*. <http://www.w3.org/TR/wsdl>
- [15] W3C. *Web Services Description Language (WSDL) Version 1.2 Part 3: Bindings*. <http://www.w3.org/TR/wsdl12-bindings>
- [16] W3C. *XML Schema Part 2: Datatypes*. <http://www.w3.org/TR/xmlschema-2/>
- [17] *WAP Forum Releases*. <http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html#wap20>
- [18] WEB Services Development center. <http://www-136.ibm.com/developerworks/webse rvices/> http://tech.ccidnet.com/pub/article/c291_a43926_p1.html
- [19] WAP Forum. *Wireless Application Environment Specification*. <http://www.openmobilealliance.org/tech/affiliates/wap/wap-236-waespec-20020207-a.pdf>