Measuring Availability of Mobile Web Services

Kee-Leong Tan, S.M.F.D. Syed Mustapha
Asian Research Centre, British Telecommunications Group,
Cyberview Lodge Office Complex, Hibiscus Block, 1st Floor,
63000 Cyberjaya, Selangor Darul Ehsan, Malaysia.

***Abstract*---Adapting present Web services to the mobile environment has raised new Quality of Service (QoS) challenges for both service provider and requester. In a mobile Web services (MWS) scenario, a successful service invocation will depend on (1) service application (2) condition of mobile network and (3) capabilities of mobile device. Hence, from a client perspective, the availability of a service will depend on all three factors. This paper presents an Availability Checking Model (ACM), to determine the availability status of MWS by processing and referring to end-to-end service QoS parameters. This model introduces availability metrics to quantify service availability checking points. These metrics are represented by a XML schema and published as extension to WSDL. We also discuss mechanisms to aggregate server-side, mobile network and client device metrics to calculate the overall availability information. By knowing the end-to-end availability of a service, mobile client may use that information to make necessary decisions such as to proceed with invocation, to look for alternative service, to retry after predefined time or to stop entirely. Besides, this "check before subscribe" approach will help mobile user to save cost as it prevent unnecessary message transactions. For a service provider, this will improve its overall QoS and can be useful reference for managing Service Level Agreement (SLA).**

***Keywords*---Availability, Mobile, QoS, Web Services, SOA**

I. Introduction

Service Oriented Architecture (SOA) enabled by Web Services is very popular technologies nowadays. Many Web service providers have adopted standards like SOAP, WSDL and UDDI to develop and offer whole range of services to industries such as financial, telecommunication, media and entertainment. The open transport feature of SOAP allows network application to be independent of its underlaying network, platform and language. This ubiquitous advantage made Web services standard an ideal candidate to host and deploy services in the mobile environment as well.

Gebaur and Shaw [17] in their survey suggested the deployment of Web service in mobile environment will contribute to the success of mobile commerce. This trend is complimented with the popularity of more powerful and device-adaptive tools for developing dynamic and rich user experience mobile application such as J2ME Polish [1], JSR-172 [2] and Ajax Opera Platform [3]. One promising technology is Sun kVM (kilo Virtual Machine), which is a stripped down version of the Java Virtual Machine designed to have a small memory footprint of 40 to 80 kilobytes only. Another technology which will have a big impact on mobile Web services adoption is Asynchronous Javascript and XML (AJAX), which allows XML communication between mobile clients with service provider. This will lessen the traditional bandwidth issues. Besides, it also enables transparent updating of information pushed to the mobile client.

A. QoS Issues for Web Services in Mobile Environment

As the number of mobile devices consuming Web services increases, there arises a need for better solutions to manage QoS for mobile Web services. Generally, QoS management for Web services are more complex due to constrains of mobile environment and mobile devices. The basic constraints of mobile environment are limited bandwidth connection, unpredictable response time and high probability of packet loss. The constraints of mobile devices are limited computing power, memory, screen size and battery lifetime.

In addition to these, Web services suffer from unnecessary data overhead due to usage of SOAP/XML to produce human readable text as well as to achieve interoperability. Reference [5] quantifies the average overhead for Web service can reach as high as 400% compared with representing the same data in binary format. The use of universal HTTP and TCP as the underlying SOAP transport protocol too, contribute to extra overhead. In order to send a SOAP message over HTTP, a TCP connection must be established through a three-way handshake. Once the SOAP message is sent, the connection is closed through another handshaking process. The establishing and closing of connections increase the transmission cost and lead to longer delays as discussed in [18].

From mobile users' perspective, these MWS QoS issues will pose a negative perception and will affect the adoption rate, as users are often charged by data transmission volume rather than by connection time. One popular approach to address this issue is to consider techniques for compact and more efficient SOAP/XML representation by compressing the data [16],[19]. The trade-off for these compression approaches is the performance loss due to additional CPU processing required. The server provider will need to compress the message and resource-constrained mobile clients will have to decompress the responses. Another approach is to use UDP instead of HTTP for SOAP binding to reduce protocol overhead [18],[20]. However, even these will not guarantee the compressed message will successfully reach its destination, as it still has to travel across an unknown intermediary mobile network, as shown in Figure 1. Besides, the Web service application on the provider network does not have a control or information about the available bandwidth between mobile client and the server.

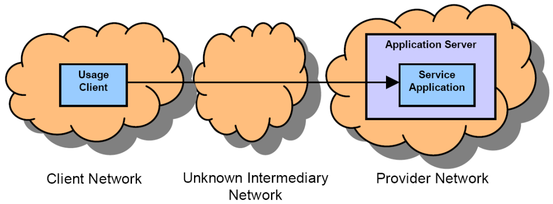

Figure 1: Client access over public network [6]

In this paper we propose a MWS Availability Checking Model (MWS-ACM), an approach to manage, measure and deliver the required QoS metrics in a mobile environment, and formulate the overall availability status to mobile client. This model proposes the integration of *Availability Metrics* into WSDL, allowing a mobile client to retrieve the overall end-to-end metrics, and formulate them into availability information prior to invoking the service. The rest of the paper is outlined as follows: section 2 present some background information on availability issues for mobile Web service. Section 3 further details the QoS metrics used in our availability checking model approach. Section 4 presents our architecture detailing components used to obtain the various availability metrics. After presenting the related work in section 5, we finally conclude with a short summary of the related works and our future QoS research in mobile Web services.

II. QoS and Availability

QoS refers to a level of service that meets user requirements. In Web services, the related term is Service Level Agreement (SLA), which is often used to describe the contracts and relationships between the user and the service as to what constitutes 'satisfactory' [3]. Several quality parameters or attributes like the *availability*, *security*, *response time*, and *throughput* are regarded the criteria for quality driven selection of web services [4] and these normally included in SLA as part of service definition. The focus of this paper is on availability, a QoS attribute defined as the quality aspect of whether the Web service is present or ready for immediate use [5], often represented as the percentage of uptime of a service in a pre-determined period.

From a service provider’s viewpoint, the service availability is dependent on service application and its connectivity with other components on the provider network such as Application Server, Web Server, Database Server, etc. We collectively group these connectivity attributes as *Server Component Metrics*, which values will determine the accessibility of components from the service application. The performance of service application which resides on application server will be affected by the ‘health’ of the system server, dependent on attributes such as CPU utilization, memory consumption, disk usage, network traffic, etc. which are grouped as *Server System Metrics*.

For a mobile user perspective, the factors which could affect end-to-end availability for an arbitrary service over mobile network, has to be extended even further, as service transaction will have to go through an unknown intermediary network (wireless LAN, GPRS, 3G). This is not so much an issue in traditional distributed systems on wired network, as the network and other system environment components are likely to be under control of the same organization.

Mobile network performance varies depending on the route condition between the client and the service. Hence the network QoS measurements taken between one client and some service cannot be used as a reliable indicator of the overall mobile network performance. This will require the network channel characteristics to be a point of availability measurement for every single mobile device. The attributes such as response time, latency, etc. are grouped as *Mobile Network Metrics*. Having this, the value of network data transfer rate can be derived. This value can be used to compute the time taken for a response message of a certain size to reach the mobile client. The message data size can be estimated based on the output parameters and their data types as defined in WSDL file, these data are grouped under *Message Size Metrics*.

To ensure end-to-end service interoperability, the client system capabilities also need to be taken into consideration. Its system attributes a.k.a. *Device Metrics* or device profile such as Mobile Information Device Profile (MIDP) and Connected Limited Device Configuration (CLDC) may be of interest if service application has a minimum capability requirements set. For instance, a service provider who returns a service invocation message with video attachment may require the mobile device to have at least 320 x 240 pixel color screen, and 1 MB available memory. By knowing the client’s *Device Metrics*, service provider will be able to perform requirements matching and, warn the mobile client if the minimum requirements are not met.

In summary, to obtain a more accurate availability status of a Web service in mobile environment, five QoS parameters have to be considered. Say $A_{component}$ denotes *Server Component Metrics*, A_{system} denotes *Server System Metrics*, $A_{msgsize}$ denotes *Message Size Metrics*, $A_{network}$ denotes *Mobile Network Metrics* and A_{device} denotes *Device Metrics*, then the overall availability for a mobile Web service, A_{MWS} can be defined as follows:

$$A_{MWS} = A_{component} + A_{system} + A_{datasize} + A_{network} + A_{device}$$

For instance imagine a mobile client wish to check *availability* of a service which performs a calculation of a statistical data and returns the result to the client. If time to complete is of importance to the client then not only the service's speed of operation (*Server System Metrics*) is relevant - but it must take into consideration the network latency, bandwidth and error rate (*Mobile Network Metrics)*. And should it choose to display the result in graphical representation, a prior check of mobile client capabilities to support graphical image is required (*Device Metrics*).

III. MODELING SERVICE AVAILABILITY METRICS

Table 1 further describes the attributes for all availability metrics discussed in section 2.0. The attributes are quantified variables whose values will be used to determine the overall availability status for a mobile Web service, A_{MWS}. The values are computed by executing certain software functions to capture and measure factors affecting each availability metrics.

Sequence diagram in Figure 2 shows where availability metrics originate and the flow of the metrics data during availability checking usage scenario. Except for *Mobile Network Metric*, all the metrics' values are computed at the server-side, and subsequently transferred to mobile user. This is to save the CPU and memory resource of mobile device. As for *Mobile Network Metrics*, *latency_timer* is used to measure the time it takes to transfer the message from client to the server over the mobile network [16].

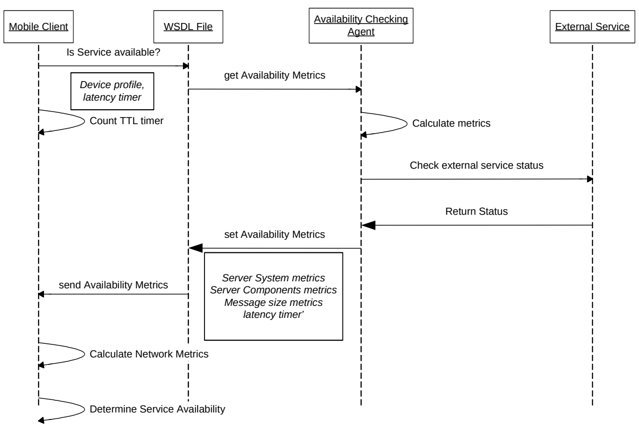

Figure 2: Flow Sequence of Availability Metrics. The final availability status will be determined on the mobile client.

TABLE 1
ATTRIBUTES FOR AVAILABILITY METRICS

Metrics	Attributes	Description
Server Component	*Server_connectivity*	Status response from server binding element, TRUE=Status ON, FALSE=No status.
Server System	*CPU_usage*	Percentage of server CPU utilization, 0 = 0% usage (totally free), 100=100% usage (server has reach maximum capacity)
	Memory_usage	Percentage of server memory consumption, 0=0% usage, 100=100% usage.
	Network_traffic	Bandwidth consumed by network input/ output activities, in bits per second.
	Data_binary	To determine the message will contain binary data, 0 = text only. 1=binary (or combination of text and binary)
	Data_size	Estimation of total data size (in KB) to be transmitted during service invocation
Mobile Network	*Delay*	Time (in second) taken for a message to be transmitted.
	Response_time	Round-trip time (in second) from mobile client request transmission to reply acknowledgement from server.
	Jitter	Variation in delay or response time
Device	*Device_profile*	Information describing characteristics and features of the mobile device. This includes the type of device, model number, display size, input and output methods, etc.

IV. PUBLISHING THE AVAILABILITY METRICS

To execute a Web service, a service requester has to reference WSDL file and SOAP endpoint address. Before a Web service is invoked, requester needs info such as message data format, message exchange pattern and service provider's network address. A WSDL description for a Web service will provide details such as the inputs, outputs, the data types of those inputs and outputs, and how the Web service should be invoked. Presently, a common practice is to have WSDL posted at an accessible URL which is made available to requester by some out-of-band methods such as web site listing, email attachment or printout.

Since WSDL is the de facto method to describe and publish a Web service, MWS-ACM will provide the availability information to the mobile clients by extending the WSDL to support the availability metrics as discussed in section 3.0.

To ensure interoperability across multiple devices, platforms, Operating Systems, versions and languages, we have adopted the guidelines and best practice for designing interoperable Web Service [9]. However the challenge remains in defining data schema to ensure the conformant of all value types data values when it moves between a service client and service endpoint. For example, mobile Web services does not support Java Collection types, which means the mobile Web services clients will probably fail to generate stub files from a well-formed WSDL file. To address this issue, we adopt those proposed rules suggested by S. Fang Rui, in [7] by using only the preferred data types for attributes which are most relevant to availability measuring.

```
<definitions xmlns:mwsA="http://www.arc.bt.com/wsdl/mwsAvailability/" ...>
   <binding ...>
      <operation name="ExampleRpcEnc">
                    <mwsA:MobileNetwork Response_Time="130"/>
             <soap:operation  soapAction="http://abc.com/ExampleRpcEnc" style="rpc"/>
            <input>
                  <soap:body use="encoded"  namespace="http://abc.com/"
                        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </input>
            <output>
                  <soap:body use="encoded"  namespace=http://abc.com/
                        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </output>
      </operation>
   </binding>
</definitions>
```

Figure 3: WSDL extensibility element for Mobile Network Metrics

She also further refines other modelling design considerations for mobile Web services.

In MWS-ACM, the XML type definitions used will be stored as a separate schema file, *mwsAvailability.xsd* for better readability as well as reusability of WSDL documents. The schema file can be referred by any service provider who wishes to provide availability measuring feature as part of the service. This can be done by using the WSDL *<import>* to import those availability elements into the WSDL document.

On the client side, the availability metrics obtained will have to be mapped to corresponding elements for further calculation to support decision making, or as service parameter for SLA. This is possible by defining some WSDL extensibility elements that belong to a new XML namespace called *mwsAvailability* (eg: http://www.arc.bt.com/wsdl/mwsavailability/). These extensibility elements are designed to extend parts of WSDL by providing more information specific to availability checking mechanism. They can be included within relevant parts of the WSDL document as suggested in [10].

Figure 3 shows an example of using a WSDL extensibility element called *mwsA:MobileNetwork* for *Mobile Network Metrics*. Clients may use the attribute value of *Response_time* to determine if the mobile network bandwidth is sufficient enough for their needs. If it is below the required value, the client may look for another service than can provide the same functionality with a faster guaranteed response time.

V. AVAILABILITY CHECKING ARCHITECTURE

The architecture of the MWS-ACM typically consists of three main components: the mobile client, service provider and *Availability Checking Agent* (ACA), as shown in Figure 4. According to this architecture, a service provider publishes its service description and interface within a WSDL file, with an availability extension. Upon discovery of the service it wishes to invoke, mobile client will initiate a request to check the availability status of the Web service. This request will be intercepted by ACA, which then initiate availability status measurement, calculate the metrics and return their values to a mobile client.

A. Mobile Client

The client is responsible to initiate Web service availability checking request. The *Availability Status Requestor* will initiate the service availability status request, which includes latency timer and device profile attributes. Device profile will be used to determine if the device meets service requirements. Referring to Figure 2, *latency_timer* is used to clock the message response time in a mobile network, while *ttl_timer* is a predefined counter used to set the timeout period. If message response received before *ttl_timer* expires, the *Service Availability Calculator* will aggregate all computed and non-computed metrics, and quantify the Web services availability status employing weighted sum mechanism. This is to allocate more weight for certain metrics under specific conditions. An example is for service which requires high CPU and memory resource, more weighting is given to *Device Metrics*, as it constitutes a significant percentage of overall impact factor for availability.

B. Availability Checking Agent

The *Server-side Availability Calculator* is responsible for checking the status as well as connectivity between Web services and external resources, such as connectivity of database, application server and web server. The calculator will quantify all attributes as discussed in section 4.0 and produce server system and component metrics.

From a WSDL file, the number of input/output parameters and their data types can be known. These data will be used by *Message Data Calculator* to estimate the size of message that will be sent or received by service provider. The values quantified (*Message Data Metrics*) will be used together with *Mobile Network Metrics* to obtain a more precise duration of message arrival time.

The ACA is also equipped with *Device Compliance*

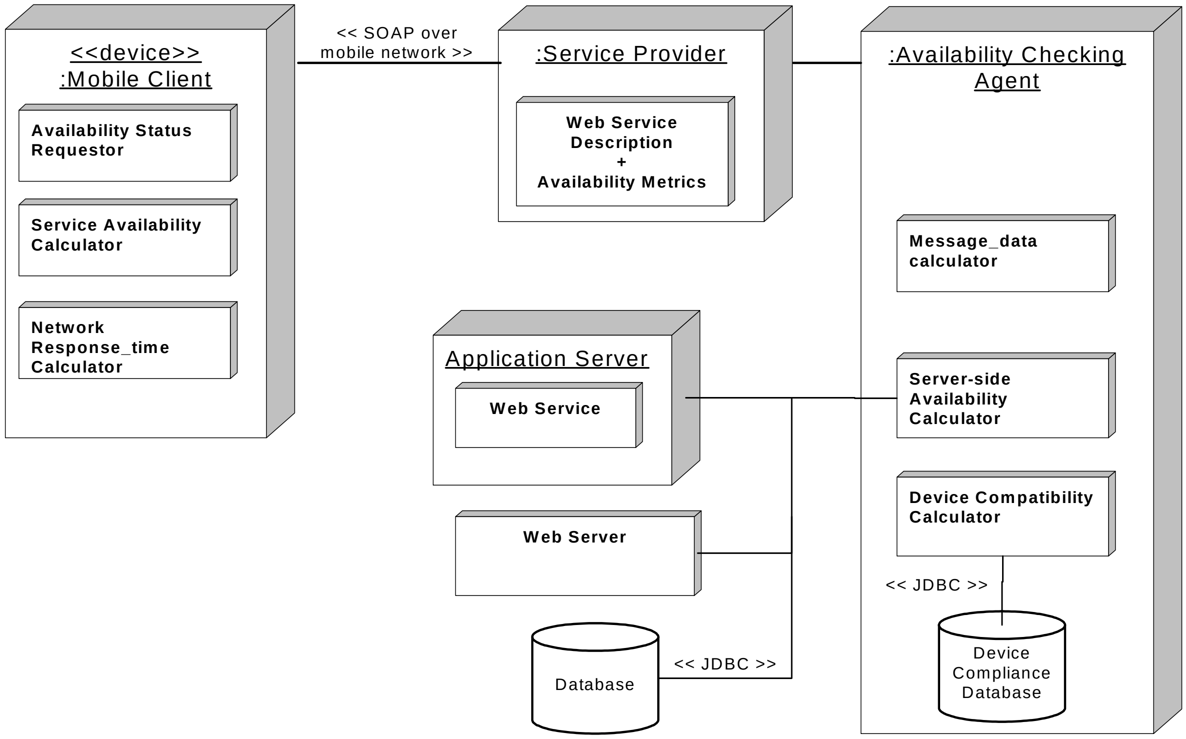

Figure 4: Architecture of Mobile Web Service Availability Checking Model

Database, where records of all supported mobile device information based on WURFL and OMA will be constantly updated. The mobile client supplies its device profile (*Device metrics)* at the beginning of service availability checking, the *Device Compatibility Calculator* will retrieve the device profile and perform a query to search for matching result from the *Device Compliance Database*. The result value could be either 1 or 0. Value of 1 indicates the mobile device meets the minimum requirements of the Web service, while value of 0 indicates it is not supported.

VI. RELATED WORKS

QoS issues in Web services technology, with its composite, distributed and heterogeneous nature still presents a formidable challenge to both industrial and academic research. Among the major industrial and academic efforts working towards the specification and management of QoS for Web services are the WS-Agreement by Global Grid Forum [21], Web Service Level Agreement (WSLA) developed by IBM [11], the Web Service Offering Language (WSOL) developed at Carleton University, Canada [12], SLAng developed at University College London, UK [13]. However these efforts are mainly focussing on Web services in wired network and do not pay much attention to constrains in mobile network. With regards to QoS issues for mobile Web services, present research works can be categorised into two general areas: (1) Evaluation of mobile Web service performance and (2) Improving QoS by reducing SOAP/XML overhead.

There have been some significant findings related to mobile Web services performance as published by M.Tian [15] who evaluates the performance of Web service performance in mobile environment and proposes a scheme that supports a server in the decision whether to compress Web service response. The experiment conducted also shown mobile clients with poor connectivity benefit from compression but compression also reduces server performance during high demand. Mikko L. experiment results [16] show the network performance is the dominant factor in the whole Web service invocation process in slow mobile network. They too found XML compression technique is well-justified in mobile Web service as message size is inversely proportional to the transfer time.

As a corollary to the above, most recent efforts try to improve the QoS performance by applying prolific compression techniques to SOAP/XML data. Chistian W. et al introduced SOAP compression approach based on differential encoding [19] where it transmit only the difference of an ordinary SOAP and a skeleton SOAP message generated from WSDL. References [19][20] study the use of UDP as an alternative binding protocol for SOAP, and discovered it reduces protocol overhead significantly and gives higher troughput as compared to SOAP-over-HTTP.

In the related works quoted above, the availability of MWS is often only treated as one of the QoS parameters. Our argument articulates availability as the most significant and crucial QoS parameter prior to service invocation. We see the need to further refine and improve its description, checking mechanism and measurements for MWS context. Its description should cover end-to-end dependencies such as on the mobile client, network and service provider; while checking mechanism for each dependency has to look for the most accurate points of measurements. By improving the availability checking and measurement, the *availability metrics* values obtained will reflect the actual status of the end-to-end dependencies, thus will contribute to the accuracy and precision of the overall MWS QoS.

VII. Conclusion And Future Work

This is our effort as one of the answer to the call by [6] which read "*Innovative solutions are requested to be able to bundle QoS properties of a service application with the properties of the provider's network, the client's network and the network connecting both*". We propose MWS-ACM to measure the end-to-end availability status of a Web service in mobile environment. The model defines metrics to *bundle* the QoS properties of service provider, mobile network and mobile client. A server-side component, *ACA*, functions as a service broker to invoke processes to aggregate and measure the metrics' values. The interface to get these values can be embedded into the WSDL file of service application. Once the mobile client obtains all the metrics' values, it will dynamically formulate them into the overall availability status of that particular mobile Web service. This final availability status can be used either as service parameter for SLA, or as a condition to support further decision making. We are currently working on implementing a prototype based on ACM, formalizing the calculation of the metrics and the overall availability status A_{MWS}, using weighted sum mechanism. As part of our future work, we intend to evaluate the performance of MWS-ACM in a real mobile network environment.

References

[1] J2ME Polish http://www.j2mepolish.org/docs/features.html

[2] Sun JSR 172 Micro Edition Web Services Specification http://java.sun.com/j2me/docs/j2me_jsr172.pdf

[3] Judith M. Myerson, Guarantee your Web Service with an SLA: Introduction, architecture and testing mechanisms, http://www-106.ibm.com/developerworks/webservices/library/swsla.html, April 2002.

[4] Daniel A. Menasce. Qos issues in web services. In IEEE Internet Computing, pages 72–75. IEEE Press, November-December 2002.

[5] Anbazhagan Mani and Arun Nagarajan, Understanding Quality of Service for Web Services, http://www-106.ibm.com/developerworks/library/wsquality.html

[6] H. Ludwig, "Web Services QoS: External SLAs and Internal Policies: Or, How Do We Deliver What We Promise?", Proc. 4th IEEE Int'l Conf. Web Information Systems Eng. Workshops, IEEE CS Press, 2003, pp. 115–120.

[7] S. Fang Rui, Designing mobile Web services, Jan 2006, http://www-128.ibm.com/developerworks/wireless/library/wi-websvc/

[8] Wireless Universal Resource FiLe. (WURFL), http://wurfl.sourceforge.net/index.php

[9] Wangming Ye, Web services programming tips and tricks: Improve interoperability between J2EE technology and .NET, Part 3, Feb 2005, http://www-128.ibm.com/developerworks/webservices/library/ws-tip-j2eenet3/

[10] WSDL Appendix 3, where we can place our own extensibility elements?

[11] A. Keller, H. Ludwig (IBM), The WSLA Framework: Specifying and Monitoring of Service Level Agreements for Web Services, IBM research report RC22456, 2002.

[12] V. Tosic, B. Pagurek, K. Patel, B. Esfandiari, and W. Ma, Management Applications of the Web Service Offerings Language (WSOL), The 15th International Conference on Advanced Information Systems Engineering(CAiSE'03), Velden, Austria, June 2003, http://www.sce.carleton.ca/~vladimir/publications.html

[13] D. D. Lamanna, J. Skene, and W. Emmerich, SLAng: A language for defining Service Level Agreements, The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03), San Juan, Puerto Rico, May 2003, http://csdl.computer.org/comp/proceedings/ftdcs/2003/1910/00/19100100abs.htm

[14] Z. Chen, C. Liang-Tien, B. Silverajan, and L. Bu-Sung, UX – An Architecture Providing QoS-Aware and Federated Support for UDDI, The 2003 International Conference on Web Services (ICWS'03), Las Vegas, Nevada, USA, June 2003, http://www.ntu.edu.sg /home5/ PG04878518/Articles/ICWS03_Paper.pdf

[15] M. Tian, T. Voigt, T. Naumowicz, H. Ritter, and J. Schiller. Performance Considerations For Mobile Web Services. Elsevier Computer Communications Journal, 27(11):1097–1105, July 2004.

[16] Mikko Laukkanen and Heikki Helin, Web Services in Wireless Network – What Happened to the Performance? The 2003 International Conference on Web Services (ICWS'03), Las Vegas, Nevada, USA, June 2003.

[17] J. Gebauer and M. Shaw. Success factors and impacts of mobile business applications: Results from a mobile eprocurement study. International Journal of Electronic Commerce, pages 19–41, Spring 2004.

[18] Kwong Yuen Lai, Thi Khoi Anh Phan, Zahir Tari, Efficient SOAP Binding for Mobile Web Services, Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05).

[19] Christian Werner, Carsten Buschmana, Stefan Fischer, Compressing SOAP Messages by using Differential Encoding, The 2004 International Conference on Web Services (ICWS'04), Las Vegas, Nevada, USA.

[20] Christian Werner, Carsten Buschmana, Stefan Fischer, Enhanced Transport Bindings for Efficient SOAP Messaging, The 2005 International Conference on Web Services (ICWS'05), Las Vegas, Nevada, USA.

[21] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, Web services agreement specification (WSAgreement) (draft). https://forge.gridforum.org/projects/graap-wg, 2006.