The Proxy-based Web Service Adaptation System for Mobile Clients

Shwe Thinzar Aung, Aung Win

Abstract—The integration of mobile device and web service are becoming popular. This integration introduces new opportunities and challenges. The encoding and decoding of verbose XML-based SOAP messages is the main challenges for mobile devices because of mobile environment's constraints (low processor speed, limited memory, limited battery, and slow intermit wireless connection). The next challenges, the contents (image, text, audio, and video) that return from web services are not convenient for mobile devices as the mobile device's screen is small. In order to solve these challenges, it is important to parse and adapt the SOAP messages to fit the characteristic of mobile devices. In this paper, we present the proxy-based web service adaptation system that provide content adaptation based on rules created for mobile device. This system acts as a gateway between mobile client and service provider. It provides extraction, adaptation and creation of GUI to web service according to the client's device limitations. So the proposed system can minimize computational load on mobile client and enhance the presentation of content to user.

Keywords—content adaptation, rules, SOAP, webservice, XML

I. INTRODUCTION

The web access by people via laptops and smart mobile devices is likely to exceed web access from desktop computers within the next five years. Thus each client uses the internet in different contexts of physical devices as well as a unique environmental and personal context. So web access still suffers from interoperability and usability problems [11]. Web service must offer a solution for interoperability problem because it provides interoperable machine-to-machine interaction over a network. Interoperability of Web Services mainly stems from its Extensible Markup Language (XML) based open standards. Although the integration of mobile computing with Web Services technology will give many advantages to both sides, it will lead to unacceptable performance overhead. First, the overhead of parsing SOAP response does not match with the limited hardware capabilities that mobile devices present nowadays. Second, network bandwidth limitation and the unreliable wireless communication on mobile devices are also decreasing the over-all support for web services consuming on handheld devices [7]. Moreover, the content (images, text, audio and video) that returns from web services is not suitable for mobile devices due to the small physical size of the mobile phone. The entire problems are caused by the limitation of mobile devices.

In order to overcome the terminal and network limitations, we need to support adaptations of XML message from web service. Content adaptation is a process of transforming content to adapt to various devices. This procedure is usually related to mobile devices that require special handling because of their limited computational power, small screen size and constrained keyboard functionality. So in this study, we propose the proxy-based web service adaptation system based on rule engine. The system acts as a gateway between the client and service providers. It also offers the dynamic content adaptation. In dynamic content adaptation, the system needs to select the appropriate adaptation process accordingly the specification of device and the contents. To accomplish this, the system uses rule engine to select optimal adaptation process for contents that meet the constraints of client devices. This proposed system takes care of the burden of the encoding and decoding of verbose XML-based SOAP on behalf of client and offer content adaptation for web service's response according to physical specification of mobile device.

II. RELATED WORK

Several message optimization approaches have been introduced to address web service performance overhead for mobile device. To parse SOAP messages, the researchers introduce two fast parsers: DOM (Document Object Model) and SAX (Simple API for XML). The DOM parser demand more resources both in memory and CPU usages [3]. Although the SAX is faster than DOM, it is significantly more complicated and complex. Applying XML parser to mobile device for SOAP messages parsing may result in performance overhead.

To reduce the communication payload of SOAP enveloping mechanism using compression algorithms, e.g. the free library called SharpZipLib which contains the following compression formats Bzip2, Gzip [4]. The percentage of compression is about 90%, which means that the XML file is decreased about 10 times using compression algorithm. Compression reduces server performance during high demand.

But, the existing proposed solutions in the research area is to enforce the belief that any feasible communication architecture for mobile devices must include a middleware component that exists outside the device and takes the responsibility and heavy load of XML processing needed in...
communicating with the Web service. Those middleware components will act like gateway servers that communicate lightly with the device (thus ensuring a small bandwidth usage – for limited GPRS data communication and little chance for failure in unreliable wireless networks) and take the load of retrieving the response from the Web service [6]. This type of architecture can bring many more opportunities towards ensuring a more reliable communication with the Web service.

The Web Service architecture for mobile clients presented in [7] deal with the evaluation of the RESTful web service for mobile devices against conventional SOAP web services. In this paper, the performance evaluation results show the advantages of using RESTful web services over conventional web services for mobile devices. Advantages include less message sizes and response time. Results of performance comparison between conventional SOAP and RESTful show the obvious high performance RESTful over SOAP. Therefore, RESTful offers a perfectly good solution for the majority of implementations, with higher flexibility and lower overhead.

III. WEB SERVICE

The term web service describes a standardized way of integrating Web based application using the XML, SOAP, WSDL and UDDI over an internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the web services available and its capability and UDDI is used for publishing the web service [9]. Unlike traditional client/server models, the web services do not provide the user with a GUI. Web services share business logic, data and processes through a programmatic interface across a network. Developers can then add the web services to a GUI to offer specific functionality to users. Web services allow different applications from different sources to communicate with each other without time consuming custom coding, because all communication is in XML. Web services are not tied to any operating system or programming language.

IV. CONTENT ADAPTATION

Content adaptation is a process of selection, generation or modification of content (text, images, audio and video) to suit to the user’s computing environment and usage context [2]. It is usually related to mobile devices that require special handling because of their limited computational power, small screen size and constrained keyboard functionality [12]. Content adaptation can be classified into two types: Static Adaptation and Dynamic Adaptation. Static adaptation automatically creates multiple versions on the authored content, at any time after the content has been created [10]. It will make content management more complex and requiring more storage. Dynamic adaptation performs on-the-fly adaptation as each request comes in. The adaptation can cause additional computational load and resource consumption [10]. Based on location of the adaptation process, adaptation approaches are classified into Client-side, Server-side, Proxy-based content adaptation.

In server based approach, the content adaptation process is performed by sever. In this approach better adaptation results could be achieved as it is closed to the content. However clients experience performance degradation due to additional computational load and resource consumption on the server [1].

The client-side approach can be done in two ways; performing transformation by client device or selection of the best representation after receiving the response from the origin server. This approach can be more efficient than server-side or proxy-based. However, all of the clients may not be able to implement content adaptation techniques due to processor, memory and network bandwidth limitation [1].

In proxy based approach, a proxy server analyzes and adapts the content on-the-fly, before the result to the client. Intermediate adaptation can shift computational load from content servers to proxy servers [2].

V. SYSTEM ARCHITECTURE

The proxy based content adaptation system acts as an access point between client and web services. The system consists of two components: service registration for web service and the SOAP message adaptation for mobile client. In order to adapt web service’s message, the web service provider need to register its web service in this system. In service registration, the service provider registers its web service using its WSDL file. The system download and parse WSDL file to get information about the service and store the information in the service repository.

In the SOAP message adaptation, the request of client is sent to this system and it performs a request to the desired web service provider. The request and response of web service are formed as a SOAP message. So the system must parse SOAP message to get information. Usually the content of web service’s response is designed to be suited on desktops. So this contents are adapted for the small screen size devices as the content often looks messy making it difficult for users to read the contents and also certain things like image formats need to be converted since one device may support one format and other may support another. Hence adaptation needs to be carried over. The main task of this system is extraction and adaptation of web service’s content and creation of GUI (jsp or html) to web service according to the client’s device limitations. The proposed system consists of five components: communication module, media and device profiling, rule engine, execution module, response formatter. It also uses three repositories: device, rule and service repository. To improve response time, the system use java cache system that stores the adapted content as an object. The proxy based web service adaptation system has been developed as a java web application. The following figure explains the SAOP message adaptation components.
A. Cache System

The system uses java cache system to improve response time. Java cache system is a distributed caching system written in java. It is intended to speed up application by providing a means to manage cached data of various dynamic natures. Like any caching system, JCS is most useful for high read, low put applications. Latency times drop sharply and bottlenecks move away from the database in an effectively cache system. JCS has a core concept: elements, regions, and auxiliaries. JCS is an object cache. The objects or elements can be put into cache and referenced them via the key, like a hash table. In this system, the adapted response from execution module is stored as an object into the cache. The key of the object is generated using the device model, service id and service’s parameter. When the request is received, the system creates the key and then finds the key into cache. If the key exists, the system retrieve the adapted object and response the client without connection with web services and adaptation of SOAP message. So the system can improve response time.

B. Communication Module

This module is represented as the communication channel between the client and web services. When the client’s request comes in, it extract the client’s device model from the user agent of the http request and generate the key based on the request service, device model and service’s parameter. And then it examines whether the key exists in the cache or not. If it exists, this module will retrieve the object in cache via the key and send to response formatter module. If not, this module converts its request in the form of SOAP request and then requests the web service provider. It also sends the web service’s response and user agent string to the next module to adapt this response. Moreover, it replies the adapted content to the client. By using cache, the system reduces the time of connection with web service and parsing and adaptation of SOAP message.

Algorithm:
Step1: extract the device model from the user agent.
Step2: retrieve the requested service information from the service repository
Step3: generate the key
Step4: If the key is into cache then retrieve the object and pass to the response formatter.
Else
convert the http request to soap request
connect the requested service provider
pass the device model and SOAP reply to the next module

Fig. 2 sample HTTP request from client
<soapenv:Body xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <swa:getBaganInfo xmlns:swa="http://service.baganservice/xsd">
    <swa:Param>BaganView</swa:Param>
  </swa:getBaganInfo>
</soapenv:Body>

Fig. 3 converted SOAP request from HTTP request
http://10.0.2.2:8080/WebServiceAdaptation/communication?
  service="getBaganInfo"&param="BaganView"

C. Rule, Device and Service Repository

Rule repository provides the rules for the content analysis and adaptation. Content adaptation rule define when the adaptation process need to perform over the content depending on the type of device and the content’s specification.

Device description repository (DDR) are databases that store a huge amount of information concerning mobile phones, tablets, interactive TVs, set top boxes and any device having a Web browser. The system can retrieve the device capability that match device model from the Device Description Repository (DDR)[8]. So the repository can support the information about the client’s device.

Service repository stores the information about the registered web service by parsing its WSDL. This information provides the address of web service to communicate.

D. Content and Device Profiling
To make dynamic selection of adaptation process, the system needs to identify the capabilities of devices and the characteristic of content. To accomplish this process, this module uses the user agent string and SOAP response as the input. This module defines the device’s capabilities using DDR. It also specifies the characteristic of content form the SOAP message. Then it passes the DeviceProfile object and ContentProfile object to the rule engine.

**Algorithm:**

Step 1: load and get the RuleServiceProvider object.

Step 2: load the RuleAdministrator object from RuleServiceProvider

Step 3: Create the device’s information as a java object

  (DeviceProfile)

Step 4: parse the content from the SOAP message

  Check whether the content is text or image

Step 5: If it is image then download the image and specify the characteristic of image.

  Create the characteristic of image as a java object

  (ContentProfile)

**E. Rule Engine**

The rule engine is the main component of the system. It focuses on making decision to select the suitable adaptation processes for the contents that need to meet the constraints of client’s device. So this module must compare the DeviceProfile object with ContentProfile object and provide the list of adaptation process with its parameter to adjust the contents until the constraints are met.

To accomplish this, the java rule engine JRule engine is created. A JRule engine may be viewed as a sophisticated if/then statement interpreter. The if/then statements that are interpreted are called rules[13]. A rule is typically composed of two parts: a condition and an action. When the condition is met, the action is executed. Rules can be loaded in two modes: from a XML file in rule repository and from LocalRuleExecutionSetProvider.createRuleExecutionSet method. JRule engine is based on forward chaining algorithm.

**Algorithms:**

Step 1: load and get the RuleServiceProvider object.

Step 2: load the RuleAdministrator object from RuleServiceProvider

Step 3: Get an input stream to a test XML rule set in rule repository and from LocalRuleExecutionSetProvider.createRuleExecutionSet method. JRule engine is based on forward chaining algorithm.

**Fig. 4** sample rule for format conversion process in rule repository

**F. Execution Module and Response Formatter**

In execution module, it executes adaptation processes according to the ProcessList. It collects the adapted content that returns from the process and stores it to the cache and then passes it to the response formatter.

The response formatter focuses on the creating the presentation or view for the client. It arranges the adapted content in a jsp or html file according to the device’s form factor. Then it passes this file to the communication module.

**VI. RESULT**

Ultimate goal of the proposed system is to provide the capability to adapt web service’s contents to meet the specific needs of the client’s device and to develop a GUI of web services for the client. This will dynamically identify the type of device from where the request is coming, parse the SOAP response and adapt the content of response. Thus it provides user friendly content to the mobile device users.

The resulted GUI (jsp) for adapted web service’s content given below (Fig 5) represents the original SOAP response message from web service. Fig 6 represents the web page without adapting the content of web service in mobile which leads to a messy look. Fig 7 represents the web page with adapted content of web service for Nexus One emulator.

**Fig 5** Original web service’s SOAP response
In this study, the proposed proxy based web service adaptation system process the input the web service’s SOAP response to parse the content (data) and adapt the content dynamically based on the type of content and limitation of device and provide the GUI for client. The proxy-based web service adaptation system not only eliminates XML processing on mobile devices but also adapts the content from web service to be more suitable for mobile environment. So the proposed system can minimize computational load on client and server that lead to improve response time. Moreover, the system can remove SOAP-message transferring over wireless links that can lessen the transferred data rate over connection.

It also provide GUI for client that lead to improve the user’s satisfaction. In this system, the rule of adaptation process is represented as a XML file in rule repository. So the proposed infrastructure is flexible and open as we can easily add new adaptation process with its rule and update the existing rule without disrupting the other component. For secured or proprietarily encoded content, the organization deploying the proxy will need to coordinate with the service or content provider in order to access the content for performing adaptation. So this system can not provide the security concern.

ACKNOWLEDGMENT

I would like to give my special thanks to my supervisor, Dr. Aung Win, Professor of our university, UTYCC, (University of Technology, Yatanarpon Cyber City) for his kind heartedly aids. I would like to express my deepest thanks to all my teachers and colleagues who lend a hand directly or indirectly during the laborious process of completing this dissertation successfully. I also show appreciation to my parents for their mental supporting.

REFERENCES

[1] Girma Berhe, Lionel Brunie, Jean-Marc Pierson,” Realization of distributed content adaptation with service-based approach for pervasive systems”