A Middleware-Based Approach to Mobile Web Services

Pampa Sadhukhan, Pradip K Das, Rijurekha Sen, Niladrish Chatterjee and Arijit Das
Centre for Mobile Computing and Communication (CMCC),
Jadavpur University, Kolkata- 700032
pampas122001@yahoo.com, pkdas@ieee.org,
rijurekha_s@yahoo.co.in, niladrishchatterjee@gmail.com, ari007_cse@yahoo.co.in

Abstract

With the recent advancements in mobile technology and wireless communication like Bluetooth connectivity and Wi-Fi, accessing web services from mobile devices has become a focal point of research. Due to limitation of resource capabilities in mobile devices, however they are not always capable of consuming web services or web applications that are available on the Internet. This paper presents a middleware-based architecture that could be integrated into GSM network for discovery and invocation of Web Services from heterogeneous mobile devices. Our proposed middleware detects a new mobile device when it enters in its coverage area, authenticates it and then advertises the available services. The mobile device can then invoke any of the discovered services dynamically. This paper shows how the middleware makes different sorts of connection like Bluetooth connection with Bluetooth-enabled device, socket connection with Wi-Fi-enabled device and GPRS connection with a device when it is neither in the Bluetooth nor the Wi-Fi coverage area of the middleware and then pushes the required client application needed to invoke the required service onto the mobile device.

1. Introduction

With the growing use of mobile devices like PDA, mobile phones, etc, with limited computational capabilities, users expect to invoke web services available on the Internet from these devices to solve complex business problems and to get localized information whenever they enter a new location. Availability of Internet on mobile devices does not guarantee that services available on the Internet can be readily invoked by the device. It is a real challenge for the service providers to provide location-based services [1] and web services that are available on the Internet to mobile devices because of less bandwidth of wireless network, less connection stability and higher costs [2, 3].On the part of mobile devices also, there are several limitations. Firstly, mobile devices are limited by system resources like smaller screen size, limited processing power and less file-system capabilities. Secondly, though many mobile devices are J2ME-compliant, majority of them does not have proper API support for consuming Web services. Web Services are software components that provide application logic that can be discovered and accessed over Internet through standard web mechanism and web protocols like UDDI [4], SOAP [5], and HTTP. To invoke web services over the Internet, clients need to have XML-based SOAP message processing capabilities, which are not always available. SOAP (Simple Object Access Protocol) is the protocol devised to transmit XML based data over transport protocols like HTTP, SMTP, FTP, etc. To overcome these problems, we propose a middleware-based architecture where the middleware advertises the services and helps the mobile device to dynamically invoke the required service.

Some researchers work on Mobile Web Service provisioning through automated HTML/XML conversion to WSDL [6] and improved SOAP processing [7], but they do not focus on how mobile devices should discover the services when they reach a new place or how to provide them location-based services. Web Service Description Language(WSDL) is an XML based language for describing a web service, the operations that can be invoked through the web service, the structure of the request message that carry the input parameters and response message that carry the output parameter for each operation and also the binding mechanism. A group of researchers have developed a prototype application that enables mobile devices automatically discover services and consume them [8]. However, they have not considered providing services to heterogeneous mobile devices like Bluetooth-enabled device, Wi-Fi enabled device or devices that possess GPRS [9] connectivity and do not mention how to provide these services anywhere in the world.

To address the above issues, this paper describes the middleware-based architecture that consists of the following components.

- **Middleware:** This is deployed on a computer that is Bluetooth-enabled, Wi-Fi enabled and also connected to GSM network. Because of limited coverage of Bluetooth and Wi-Fi connections, when a mobile device is out of the coverage area of the middleware available on the local server, it should connect to the middleware that is available on the Internet through GPRS connection. A web server is integrated into the middleware system, where some services that provide localized information are deployed. The server is connected to the GSM network for authenticating the mobile devices.
- **Mobile Devices:** Different types of mobile devices like PDA, mobile phones, palmtops etc, can consume the services. The minimum requirement is that these mobile devices need to have Bluetooth, Wi-Fi or GPRS connectivity.

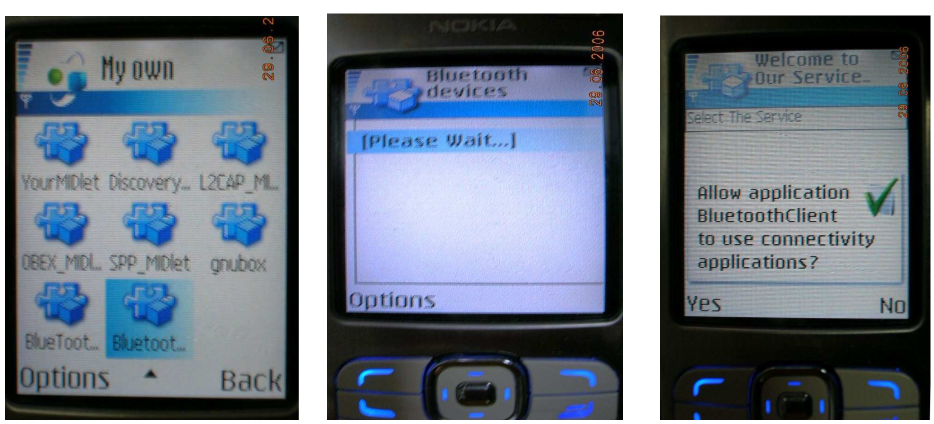
This paper is organized as follows. Section 2 provides the motivation behind the present work. Section 3 provides related works and technologies involved. Then, in Section 4, we discuss our proposed middleware-based architecture and show how it detects the entry of a new mobile device into its vicinity, advertises the services and finally how the mobile device dynamically consumes it. Work flow of the sequence of events has also been presented. Section 5 discusses challenging issues of the proposed architecture. Section 6 indicates scope for further research.

2. Motivation

In today's world, people like to have localized information and location specific services to be displayed on their mobile devices. At the same time, people would like to invoke from their mobile devices the Internet based services that provide solution for complex problem while on the move. Location specific service means that the same consumer with the same portable device will receive different sets of services depending on his current location.

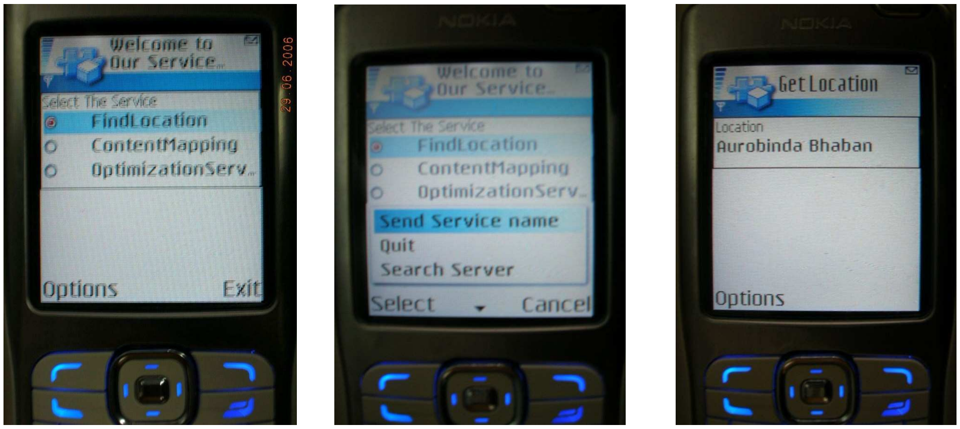
Our proposed architecture provides location-based services [1] and services available on the internet to mobile users, irrespective of the type of the device. The mobile user is also free to invoke services of his choice from a wide array of available services.

To present a typical scenario, suppose a new visitor has entered into Jadavpur University with a mobile phone like Nokia 6600 or Nokia N series or a PDA that is either Bluetooth enabled or Wi-Fi enabled where a client application is already installed. If the visitor starts that client application, after a few seconds, he would get the following messages on the screen (Fig. 1).



1 2 3

A little while after the first screen, a prompt message would appear on the screen asking the mobile user's consent to accept connection from the middleware. Once the user presses the 'yes' button, names of a number of services would appear on the next screen.



4 5 6

Figure 1: Messages that appear on the mobile screen for making connection, service advertisement and invocation of service

Now the user selects one of them (say, Find Location that gives the client the name of location he is currently standing on) and presses the ‘send service name’ button, the corresponding service would be executed by the middleware and the result sent back to the mobile device.

3. Related Work and Involved Technologies

Researchers usually focused on how to provide Internet based web services and web contents to the mobile devices and the mobile device is required to send the request for a service by sending the URL of the service. In [7], authors suggested a HTML/WSDL converter system that would receive the URL for requested service from the mobile client. The converter will generate the WSDL file so that mobile devices can access that web content as a web service. Only those mobile devices that have SOAP message processing capabilities can utilize this. They have not shown how to invoke the services from mobile devices that do not have the SOAP message processing capabilities or how to discover the services. In [8], authors have proposed proxy based distributed architecture where a local access point advertises the services that are available on the local web server. However, they have not highlighted how heterogeneous clients like Bluetooth-enabled device, Wi-Fi enabled device or a device that possesses GPRS connectivity would be able to make connection with local access point and how they would be able to invoke the services when mobile device is out of the coverage area of Wi-Fi connection of the local access point. Another crucial point, overlooked by all previous researchers, is the need to authenticate the user before providing him the services.

Our proposed architecture shows how a mobile device would be able to invoke the services from anywhere through GPRS connection even if there is no middleware system in the vicinity of the mobile device. Here the system where middleware is deployed is called the Base Station(BS). Middleware has been developed in Java and it consists of several modules. All these modules have been described in detail in the following section. A web server (Tomcat 5.0 [10]) resides on the Base Station to deploy the local web services that provide localized information. These local web services are developed by Java Web Service Developer Pack 1.6 (JWSDP 1.6) tool [11] and deployed on Tomcat server. The client program that runs on the mobile device is developed in J2ME [12]. Some of the Base Stations have global IP [13] to make them available on Internet. The middleware that is deployed on this system is termed as web based middleware. The web based middleware has been developed using WML [14] and JSP [14] and is also deployed on Tomcat web server.

4. Architecture

This section provides an overview of the proposed architecture, detailed design of generalized middleware and inter-relationship among different modules of the middleware. We also discuss the web based middleware and mobile client application followed by a discussion on Runtime Interaction between Middleware and Mobile Client.

4.1 Overview

We have mentioned the system as Base Station where middleware is deployed. Base Station is Wi-Fi enabled (802.11x card is inserted into it), Bluetooth enabled (A USB Bluetooth dongle is inserted to its USB port) and another GSM [15] box is connected to it via serial port. The GSM box has been used to authenticate new-coming mobile devices through wireless messaging facility before providing it the services. The area covered by the Wi-Fi connection and/or Bluetooth connection of Base Station is called its own cell. A building can have several Base Stations if cell area of a single Base Station cannot cover it fully. When a mobile device is out of the building, that means it is not within the cell of any of the Base Stations, web based middleware would be accessible from micro browser via GPRS connection. As shown in figure 2, the Base Station where web based middleware is deployed, need not be Wi-Fi enabled or Bluetooth enabled but it is available on the Internet.

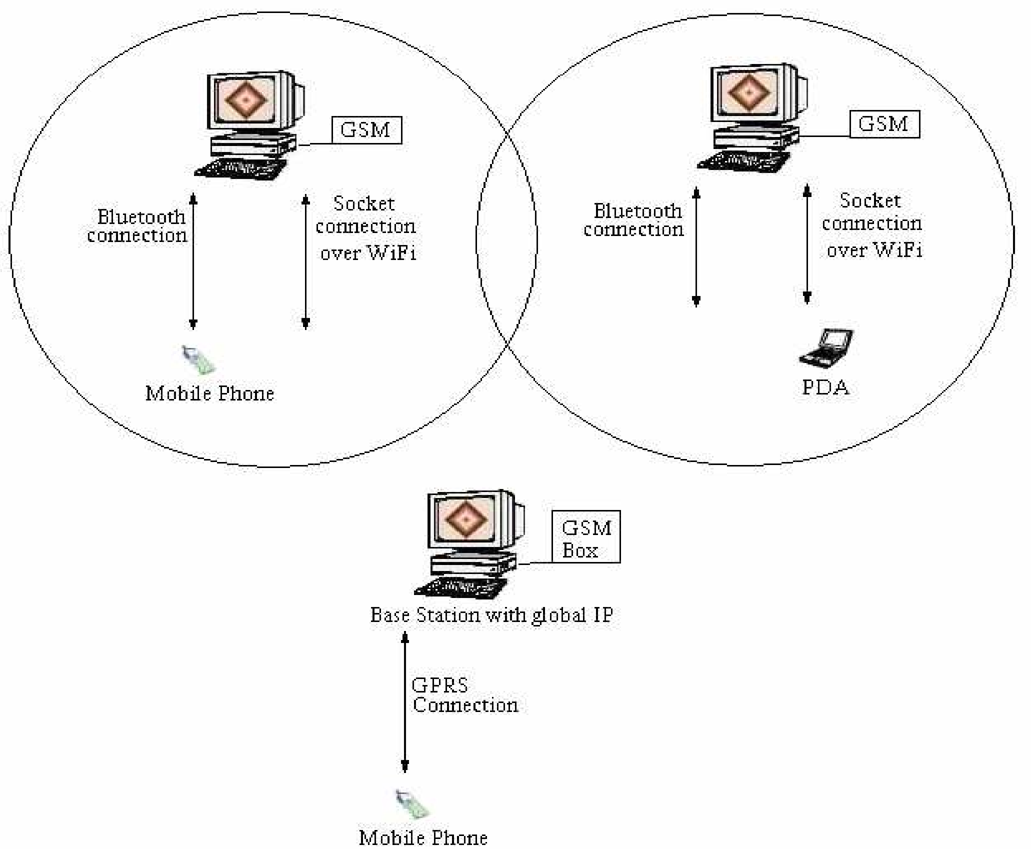


Figure 2: Architecture of the system

4.2 Detailed Design

4.2.1 Middleware

It consists of several modules (Fig. 3). In general, the middleware communicates in between the web server, where the services are deployed, and the mobile client, to facilitate the invocation of services by mobile clients dynamically. In this section we have discussed how all these modules work together and how they interact with the mobile client running on mobile device.

Main Module (MM)
It simultaneously invokes Bluetooth Connection Module and Wi-Fi connection module and continues to do this every 30 seconds.

Bluetooth Connection Module (BTCM)
It polls Bluetooth enabled mobile remote devices. Once a new remote device is discovered, it creates a new thread to search the service on that device that accepts connection from the Base Station. It can find out that service only on those mobile devices where mobile client is running, as pointed out by event (1a) in Fig. 3. Once the connection is established, it asks for the phone number, if it is a cellular phone or PDA, as shown by event (2a) in Fig 3. After it receives the number, it invokes the Authentication Module as shown by event (3a) in Fig 3 and waits for the security code to be returned by the mobile device over Bluetooth connection. After receiving the correct code from that mobile device over the established connection, it creates another thread to insert a record for that device into the local database as depicted by event (6) in Fig 3 and starts Service Advertising & Invocation Module that is shown by event (7) in Fig 3.

Wi-Fi Connection Module(WiFiCM)
It waits for the incoming client connections, that is, if a Wi-Fi enabled mobile device on which client program is running comes within its vicinity, the module detects that new device and establishes socket connection by creating a new thread, as shown by event (1b) in Fig. 3. It then asks for the phone number, if it is a cellular phone or PDA, as shown by event (2b) in Fig. 3. After receiving the number, it invokes the Authentication Module as shown by event (3b) in Fig.3 and waits for security code to be returned by the mobile device over Wi-Fi connection.

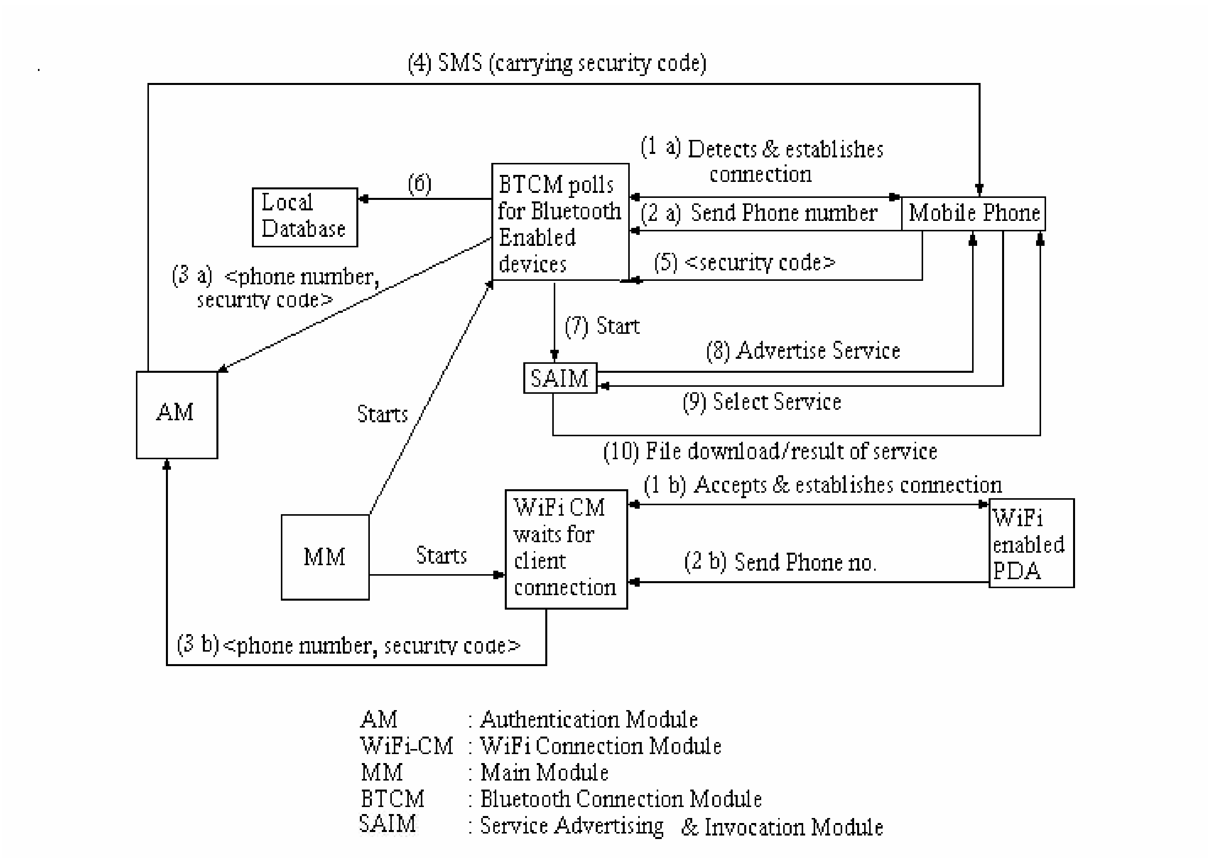


Figure 3: Modules of Generalized Middleware and their Inter-relationship

Authentication Module (AM)
This module has been developed for authenticating the newly entered mobile device before providing any service to it. This module is invoked by BTCM and Wi-FiCM, who provide this module with a security code and the phone number of the newly entered mobile device. AM then sends an SMS to the mobile device, containing the security code and an instruction to send back the code to the middleware over the established connection.
Once the middleware has received the correct security code from a newly entered mobile device, BTCM or Wi-FiCM invokes Service Advertising & Invocation Module as shown by event (7) in Fig 3.

Service Advertising & Invocation Module (SAIM)
This module advertises the services to the mobile device as shown by event (8) in Fig 3. After getting the choice from the user as shown by event (9) in Fig 3, it executes that service and sends the result to the mobile client through the appropriate connection. How it executes the service, is pictorially presented in the following section. Two things can happen here that is depicted by event (10) in Fig 3. If the service requested by mobile client does not need either any input data to be sent by the client or any other application to be run on the client side (for example user requests the service 'Find Location') middleware simply sends the result after necessary computations. On the other hand, if execution of service needs some input data from the client or any new client-side application to be run on the mobile device (for example content mapping that gives information about how to reach the destination), middleware pushes the required applications to the device. Instructions for running the received applications to execute that service properly are also sent. All these modules have been developed using Java.

Execution of Service by SAIM module
Clients have to generate stub/proxy files [17] that send the SOAP request message to web service and receive SOAP response message from it before consuming the web service. The middleware generates the stub files for a service before it advertises that service to the client.

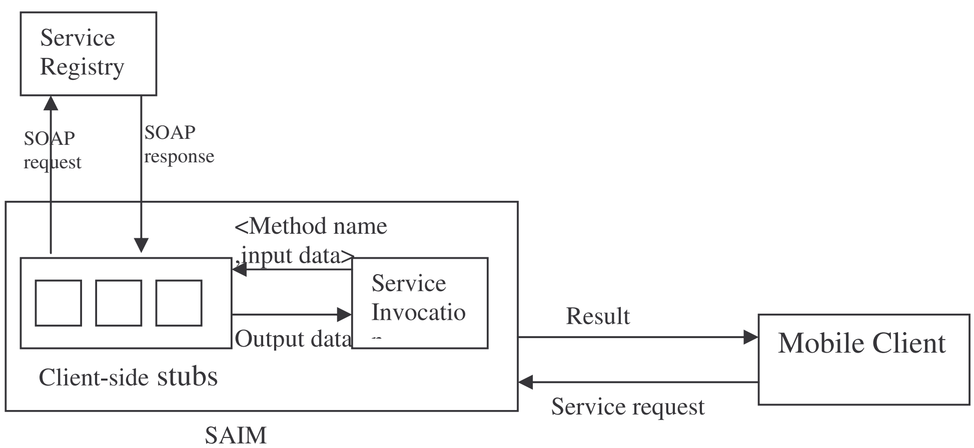


Figure 4: Execution of service by SAIM

When the client running on mobile device sends the request for a service execution with input data, Service Invocation component invokes appropriate stub files generated to communicate with the service that is available on Service Registry or available on Internet as shown in figure 4. In the latter case, service registry contains the URL of the WSDL file of that web service. When stub receives the SOAP response message that carries the result of service execution, it sends the output data to Service Invocation component and then SAIM module returns the result back to the client.

4.2.2 Web Based Middleware

It consists of three modules – Main Module, Authentication Module and Service Advertising & Invocation Module. Interaction between these modules and interaction of these modules with the client program is identical to what is shown in figure 3.

If the mobile client cannot find any Wi-Fi or Bluetooth enabled Base Station in its vicinity, client application displays a well known URL for the web based middleware and instructs the user to type in the URL on micro web browser to connect to the middleware.

For example, the URL can be

http://250.68.66.147:8080/middleware/main_module.wml

As we have already pointed out this middleware has been developed using WML and JSP. Once the Main Module has received the phone number from the mobile device, it invokes AM for ensuring that phone number is correct and makes an entry into the local database. It then invokes the SAIM to provide the service to the mobile device.

4.2.3 Mobile Client Application

It is developed using J2ME (Java 2 Mobile Edition). To successfully run the client application on a mobile device inside the cell of a Base station, the device should have the support of Bluetooth API (JSR 82) and Wireless Messaging API (JSR 120). If a device does not have support for Bluetooth API or Wireless Messaging API, client program displays the URL of a WML web page to be run on the micro web browser.

Runtime Interaction between Middleware and Mobile Client:

In this section we show the set of events and exchange of messages that takes place between the middleware and the mobile client and exchange of messages between different modules of the middleware during invocation of service by the mobile client.

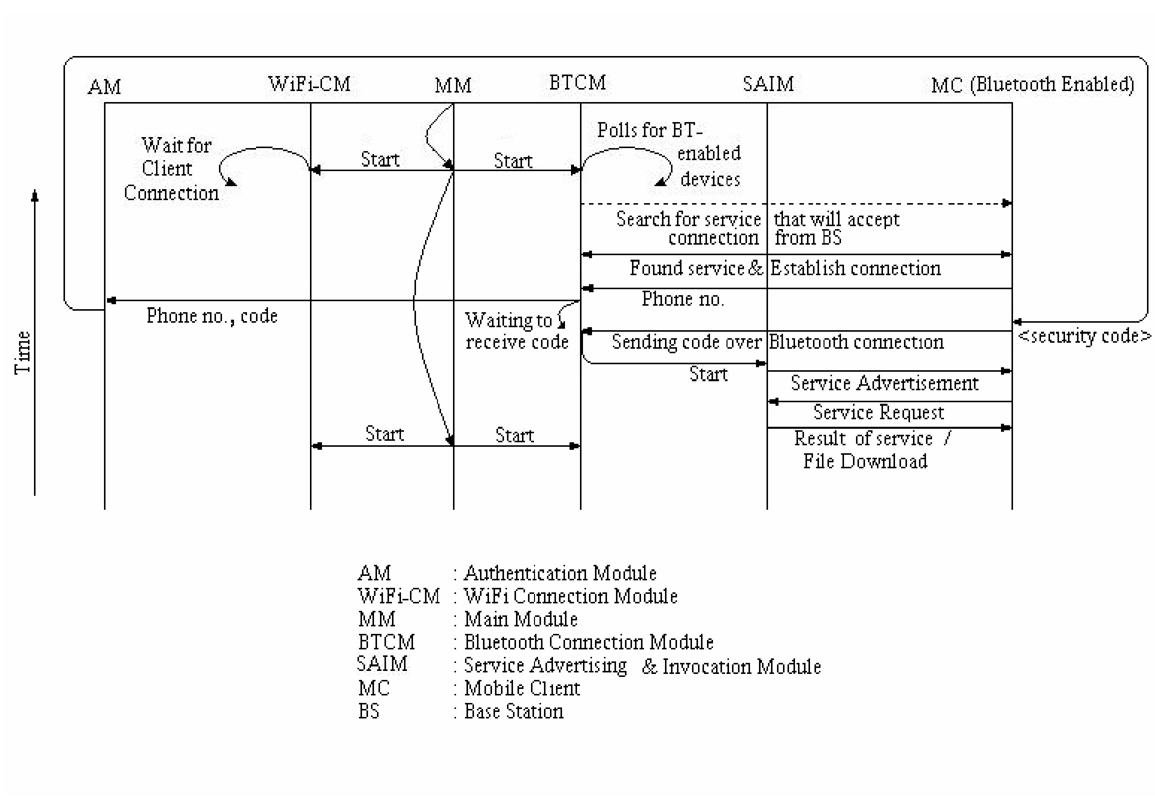


Figure 5**:** Time chart showing interaction between the client and the middleware

As shown in figure 5, the Main Module creates two new threads to invoke BTCM and Wi-FiCM in parallel. BTCM polls for Bluetooth enabled devices in the cell and Wi-FiCM waits to accept connection request from Wi-Fi enabled devices. When, a Bluetooth enabled device enters into the cell, BTCM finds it out and creates a new thread to search for a service that accepts connection from BS on that new device. Since client application is running, BTCM can find out the service and connection gets established. After receiving the phone number, it generates a unique security code for the new device, creates another thread to invoke authentication module that is provided the security code and phone number of the mobile device and it waits to receive the security code over Bluetooth Connection. Once AM has finished the task of sending the SMS carrying security code to mobile device, it exits and the thread is terminated. If BTCM receives the matching security code from the mobile device within a specified time period, it makes an entry for that authenticated mobile device into local database and invokes SAIM. As we have already pointed out depending on type of service requested by mobile client, SAIM sends either the result of service or pushes another client application onto the mobile device.

5. Some open issues

Some of the open issues in the design of the system are as follows:
i) If a mobile device is Bluetooth and Wi-Fi enabled, what sort of connection will the BS establish with the mobile device?
ii) If a Bluetooth or Wi-Fi enabled device is in the intersection region of two cells, how will that device invoke services offered by only one middleware?
To properly handle the above mentioned challenges the mobile client should consist of several components that work together as follows (as shown in figure 6).

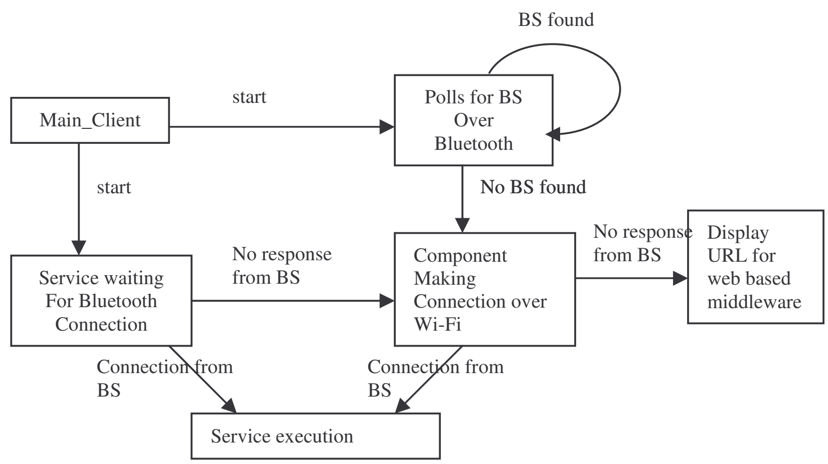


Figure 6: Different components of clients and their inter-

Usually, Wi-Fi connection range is wider than Bluetooth connection range. As shown in figure 5, if mobile device is in the Bluetooth connectivity range of middleware, it establishes the Bluetooth connection with the middleware. When the mobile device is in the Wi-Fi connection range of middleware, but not in the Bluetooth connection range, it establishes Wi-Fi connection with the middleware because Wi-Fi connection range exceeds the Bluetooth connection range.
Now suppose, a mobile device has sent a request for a service over Bluetooth connection and it has moved out of the Bluetooth connection range, a message would appear on the screen that “Connection failed. Please wait”. If the mobile device is in Wi-Fi connection range of middleware, socket connection over Wi-Fi is established and mobile device can invoke the service execution.
When a mobile device is in the intersection region of two cells, the Base Station that detects it first makes the connection with it and the mobile device can then invoke the services provided by that Base Station. If this mobile device is now discovered by the other Base Station, request for connection

establishment with this BS will not be fulfilled because mobile client can establish connection with only one Base Station at a time.

6. Conclusion & Future Work

This paper proposed to use the bandwidth of Bluetooth and Wi-Fi connection for providing adaptive location-based services [16] and application to heterogeneous mobile devices. Invoking Internet based services over GPRS connection and discovering them in UDDI registry is not a feasible solution for mobile devices. Our architecture shows how a mobile device can dynamically bind to an Internet based service by sending the request for the service to the nearest Base Station that pushes onto the mobile device the client application with stubs required to bind to the service. After completion of the service execution or after leaving the cell, the mobile device can remove the downloaded files to save its memory space. This paper also shows how a mobile device would be able to invoke the web services with the help of middleware even if it does not have proper API support for consuming the web services.

In this proposed architecture there are some open issues like if a mobile device has sent request for a service over Bluetooth connection and before receiving the result it has moved out of Bluetooth range of middleware, then service result gets lost. It can establish the Wi-Fi connection with the middleware if it is still in the Wi-Fi connection range of the middleware, to again send the same service request to the middleware. It suffers a delay of few seconds to get the result from that requested web service. We need to integrate another module into the middleware, that will keep track of a mobile device that has sent a service request over Bluetooth connection, so that, if it is not reachable via Bluetooth connection, but is reachable via Wi-Fi connection then after making the connection with the middleware only the service result will be sent.

7. References

[1] Pampa Sadhukhan and P K Das, "Location-aware Services in Mobile Environments," Proceeding of the 4th Asian International Conference on Mobile Computing, January 4-7, 2006, pp. 152-156.
[2] Siau, K., E. P. Lim, and Z. Shen, Mobile commerce: promises, challenges and research agenda. Journal of Database Management, 2001. vol. 12, no. 3: p. 4-13.
[3] Kim, H., et al. An Empirical Study of the Use Context and Usability Problems in Mobile Internet. In 35th Annual Hawaii International Conference on System Science(HICSS'02), 2002 .
[4] http://xml.coverpages.org/uddi.html
[5] http://www.w3.org/TR/soap/
[6] http://www.w3.org/TR/wsdl
[7] Park, Kim, Bae, Kim Sok, Kang. An Automated WSDL Generation and Enhanced SOAP Message Processing System for Mobile Web Services- Proceeding of Third International Conference on Information Technology: New Generation (ITNG'06).
[8] Steele Robert, Khankan Khaled, Dillon Tharam. Mobile Web Services Discovery and Invocation Through Auto- Generation of Abstract Multimodal Interface- Proceeding of Third International Conference on Information Technology: Coding and Computing (ITCC'05).
[9] http://en.wikipedia.org/wiki/GPRS
[10] http://jakarta.apache.org/tomcat
[11] http://java.sun.com/webservices/downloads/webservicespack.html
[12] http://java.sun.com/products/sjwtoolkit/
[13] **Data Communications and Networking:** Behrouz A. Forouzan
[14] www.developershome.com/wap/wml/
[15] http://ccnga.uwaterloo.ca/~jscouria/GSM/gsmreport.html
[16] Peter Ibach, Gerrit Tamm, Matthias Horbank: Dynamic Value Webs in Mobile Environments Using Adaptive Location-Based. Services. Proceeding of the 38th Hawaii International Conference on System Sciences – 2005
[17] http://developers.sun.com/techtopics/mobility/midp/articles/webservices/