Chapter

1

1. Introduction

1.1. What is WAP:

WAP (the **Wireless Application Protocol**) is a protocol for accessing information and services from wireless devices. WAP is defined and coordinated by the **WAP Forum**, a consortium of industry players who have an interest in extending the kind of information and services that we have become used to accessing over the Internet, to users of mobile devices, including mobile phones. Founded by Phone.com (now OPENWAVE), Ericsson, Nokia and Motorola, the members of the WAP Forum now include most of the leading corporations in the industry, including all the major handset manufacturers, network operators, and software companies. WAP also defines an application development environment by the name Wireless Application Environment (WAE) aimed at enabling operators, manufacturers and content providers to develop different advance services and applications.

The objective for developing WAP was to define a standard application framework that will be universal, and that will allow seamless interoperability of all of the components required for mobile access to network applications.

1.1.1. Problems with Existing Protocols:

A question can be raised that what was the need for a new protocol. There are a number of protocols already available, why can’t they be adopted in the Wireless access to the internet? The answer to this question is that most of the protocols in use today make a set of assumptions about the environment, such as the type of network that will be available (particularly from the point of view of bandwidth and reliability), the types of devices that will be accessing the services, and the types of services that will be accessed. These assumptions do not necessarily hold true in the wireless world. There are a number of differences in terms of the device itself:

- ***Form Factor:*** A mobile device needs to be small enough to move around, and ideally to be able to fit in the palm of your hand or carry in a shirt pocket.
- ***CPU:*** In a mobile device, the CPU is not nearly as powerful as a desktop PC, and is almost certainly of a different architecture.
- ***Memory and storage*:** This is a lot more constrained than on a PC, because handset manufacturers are cost-sensitive, and thus reluctant to add any additional components unless it is really necessary. Also some mobile devices do not have a persistent storage of their own.
- ***Battery*:** Mobile devices are battery powered, and the need to have the device available for long periods of time means that the processing CPU cannot make significant demands on the battery.
- ***Display:*** This is typically limited in size and resolution, and often cannot cope with color.
- ***Input:*** Mobile devices typically do not have keyboards, or if they do they are limited in size. Therefore, input is more challenging than on a typical PC.

A wireless network is considerably different to a fixed-wire network. The bandwidth of the network is typically much smaller, at least at this point in time. Reliability profiles are considerably different, particularly where users move in and out of coverage areas, disappear into tunnels, and so on. Latency may also be an issue in wireless networks. An additional factor is that there are a number of mobile network standards in place across the world, and they do not interoperate seamlessly. Some countries even have incompatible standards in different regions.

Finally, it is important to realize that the market is different where wireless applications are concerned. The types of applications that are suitable for use on mobile devices are not the same as those that are popular on fixed-wire environments. Typical users of mobile applications are likely to be a broader subset of the population than PC users. Even the context in which the applications are going to be used will be different. This highlights the most important aspect of mobile application design, which is to make the application easy to use in the context, and on the device that it will be accessed from. WAP was designed to address the issues that we have discussed above.

1.1.2. Issues handled by WAP:

On the WAP device, the WAP standard defines a **Wireless Application Environment (WAE**), which is suited to the constraints of mobile devices. The WAE includes a **microbrowse**r, which is a **markup language** browser. This browser is less stringent than existing browsers on PCs in terms of specifying exactly how a **User Interface (UI)** element is to be rendered, and concentrates instead on the functionality that is made available through the element. Hints can be provided to the microbrowser, but it is up to the microbrowser to select an appropriate representation for the device. WAP also defines a micro **Virtual Machine (VM)** for the microbrowser's scripting language to execute in, which is suited to the memory and CPU constraints of mobile devices.

1.1.2.1 Limited Resource Issue:

A markup language **Wireless Markup Language (WML)** has been defined keeping in mind the constraints of a mobile device. It is much more suited to the wireless environment than HTML. HTML is fairly strongly oriented towards the visual aspects of document rendering and what the specific user interface elements should be and should look like. While this is fine on devices that are capable of sophisticated rendering and have the capability to both render and allow the user to interact with elements, such as push buttons and framesets, it is not appropriate for most mobile devices, and phones in particular. A smaller, tighter markup language as required that is more appropriate to the wireless environment.

WML has been derived from XML and contains elements that more conveniently map to mobile devices than HTML elements. For example, WML defines an <*option*> element, which the microbrowser can render in any appropriate way that is semantically equivalent to the HTML

<*button*> element. There is also a scripting language **WMLScript**, which is derived from the standard ECMAScript. Again, compatibility has been maintained wherever

possible, and much of the semantics uses existing HTTP 1.1. WML and WMLScripts are discussed in detail in the later sections.

1.1.2.2. Network Issues:

The network issues are addressed largely through the protocol stack that was designed to take into account bandwidth limitations and reliability issues. To maintain compatibility and use existing standards where possible, it operates over IP networks, and uses User Datagram Protocol (UDP) over IP wherever possible. However, because the existing mobile networks are not packet switched, it is capable of operating over non-IP networks as well.

The WAP content types and protocols have been optimized for mass market, hand-held wireless devices. WAP utilizes proxy technology to connect between the wireless domain and the WWW. The WAP proxy typically is comprised of the following functionality:

- ***Protocol (WAP) Gateway:*** The protocol gateway translates requests from the WAP protocol stack (WSP, WTP, WTLS, and WDP) to the WWW protocol stack (HTTP and TCP/IP).
- ***Content Encoders and Decoders:*** The content encoders translate WAP content into compact encoded formats to reduce the size of data over the network. This also helps in addressing the Band Width issue.

This infrastructure ensures that mobile terminal users can browse a wide variety of WAP content and applications, and that the application author is able to build content services and applications that run on a large base of mobile terminals. The WAP proxy allows content and applications to be hosted on standard WWW servers and to be developed using proven WWW technologies such as CGI scripting, Java Servlets, ASP etc.

1.1.3. Basic WAP Architecture:

The basic WAP configuration consists of a WAP server (a normal Web server with a few modifications), a WAP Gateway and a WAP Client, but the WAP architecture

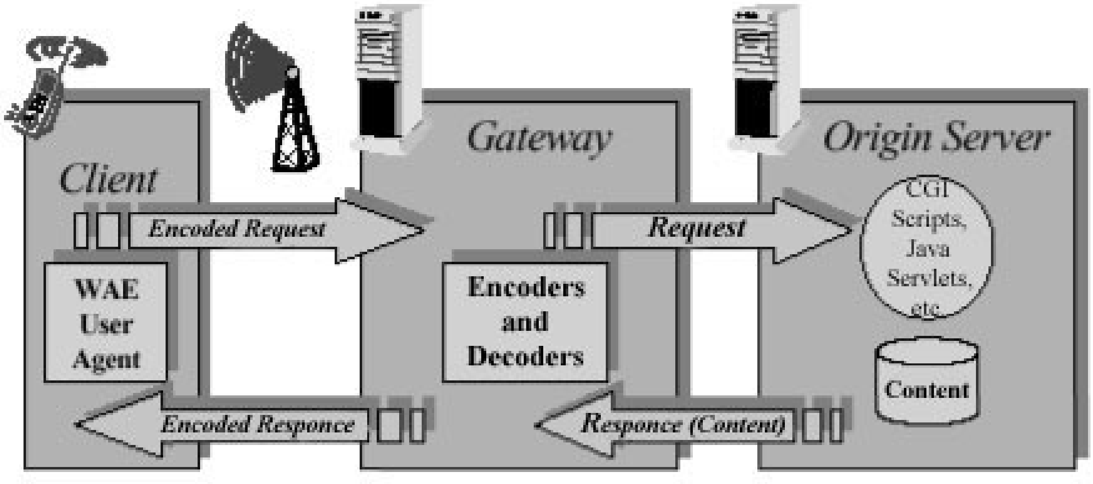

Figure 1. WAP Programming Model

can easily be change to support other configurations. For example it is possible to have a WAP server that is also a WAP Gateway. The WAP gateway communicates with an ordinary web server, using HTTP. What happens from the web server through to the application is no different from Internet access from fixed-wire devices, so the content could be static WML pages, or dynamic content generated using Servlets, ASP, CGI, or any other server-side web technology. All content is referenced using standard URLs. Figure 1 shows a typical WAP programming Model. WAP architecture will be explained in detain in later chapters.

1.2. Benefits of WAP:

The main benefit of WAP is Mobility. Mobility is the new buzzword in the business world and over time, expectations have risen about exactly what this means. In the late eighties and early nineties, mobility was associated with the ever-reachable salesman and his mobile phone. This concept expanded (mainly across Europe and Asia) with the advent of Global System for Mobile communications (GSM) in 1991. It is also possible to connect your laptop to a phone, whether by cable, IR port or, in the near future, the much-anticipated Bluetooth.

> ***Bluetooth is a new technology that is designed to provide a common way to connect mobile devices, such as PDAs, laptops and mobile phones. It was developed by a consortium including Ericsson, Nokia Intel, IBM, Toshiba, Motorola and Palm (3 Com), and its final goal is to take the place of cables and IR, providing faster connection speeds.***

In today’s business world a much more appropriate definition of mobility is:

> ***Mobility is the ability to access information and services any time, anyhow, anywhere***

This information may be email or any other personal data. The services may include banking applications, online shopping and checking stock quotes. The basic idea is extending the office to include any location the in which worker might be – at a Conference, traveling, and so on.

The increase in expectations of the mobile public over recent years has been driven by the rapid development of wireless technology. From mobile phones to PDAs and handheld computers, the devices being developed have become smaller, more powerful, and – as consumer demand increases – cheaper. This in turn drives the market forward. New technologies spread much faster than they did in the past, giving everyone the chance to experience new services. There is no longer a neat division between different categories of people. Technology available to businessmen is now equally available to teenagers. Although the markets for different categories of people are very different, they can all benefit from new and attractive services.

1.3. Potential Applications of WAP

WAP is not just accessing Internet for a mobile device. Rather it can be used in a number of ways. Few of the main categories of WAP are as follows:

1.3.1. Wireless access to Personal Information:

Wireless device users can access their e-mail, calendars, contacts, and other personal information

1.3.2. Wireless access to Internet content:

Internet content can be adapted to and leveraged for WAP devices to provide consumers immediate interactive access to information. ISPs can create mobile channels for their existing services, as well as totally new services for their mobile customers.

1.3.3. Wireless access to Corporate Information Systems:

Corporations can leverage the WAP infrastructure to deliver appropriate corporate IT systems to mobile users via WAP. Corporate e-mail access will begin this process

1.3.4. M-Commerce:

Mobile Commerce is the adaptation of the much popular *eCommerce* structure for WAP. The purpose is that the users can buy and sell things no matter where they are. An interesting point is that M-Commerce is the most popular application of WAP and other wireless Internet services all over the world.

1.3.5. Location Information Services:

A new service, made available with **WAP**, is the **Location Information Service**. This can supply the position of a network subscriber to the **WAP** applications that uses it. It is made possible by the many antennas network operators have distributed around the country to communicate with the mobile phones. The network operator always knows which antenna is receiving the signal from a given mobile phone, and of course of operator also knows where each antenna is located.

Location services provide a way of delivering location dependent information and advertising to subscribers. For example, it will allow us to find out where the nearest bank or the nearest travel agency is. These services, even though included in the **WAP** 1.1 specifications, are not yet implemented in the operator networks. When available they will provide many benefits.

Chapter
2

2. Project Specification

The project encompasses the study of different factions able to form WAP. It deals with the analysis, design and implementation of a site viewable on the mobile phone.

2.1 Aim of the Project:

The aim of the project was:

1. Setting up a WAP enabled site (Site able to be visited on a WAP enabled mobile phone)
   - To master the different languages introduced to make websites appear on WAP enabled phones.

2. Carrying out a detailed study of the Protocol (includes):
   - Study of the 5 layers of WAP Protocol Stack.
   - Study of other protocols necessary for the WAP framework to be complete i.e. TCP/IP
   - Study of the network topology/topologies that are setup in order to create a WAP framework capable of carrying out Internet across the mobile.
     This would eventually lead to the WAP Gateway

3. Constructing an M-commerce site.

Chapter 3

3. Overview of Wireless Internet World

3.1. Wireless Internet Protocols and Services:

Before we discuss the different Wireless Internet Protocols and Services it is better that we first take a look at where in the world are all the users of these services and what services are most popular. The major population of these users is in Japan. Almost 81% (35 million) of the total number of Wireless Internet users are in Japan. And this number is growing day by day. Figure 3.1 gives the breakdown of the wireless internet users in terms of their location and the service that is used by them.

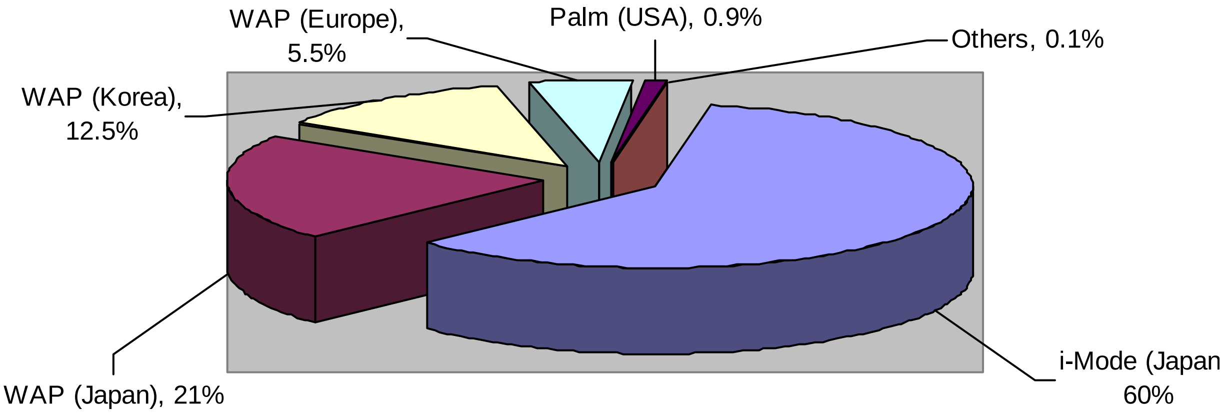

Figure 3.2 Wireless Internet Users (Source: Eurotechnology Japan K.K)

As it is clear from the figure that i-Mode is the most popular service in Japan. Wireless internet users use the following systems:

- i-Mode: 60% of the world's wireless internet users

- WAP: 39% of the world's wireless internet users
- Palm.net: 0.9% of the world's wireless internet users
- Others: 0.1% of the world's wireless internet users

Lets us now look at these services in detail

3.2. i-Mode:

i-Mode is a service developed by Japan's largest ISP, NTT DoCoMo that enables users to access Internet services via their wireless phones. In some ways, i-Mode is equivalent to AOL: both are a brand representing a service or family of services. Until now, DoCoMo's advertising focused more on entertainment than on business applications. As is the case for the wired Web based mostly on PCs, i-Mode killer app is email, comprising nearly half of the total traffic.

Technically i-Mode is an overlay over NTT-DoCoMo's ordinary mobile voice system. While the voice system is "circuit-switched" (i.e. you need to dial-up), i-Mode is "packet-switched". This means that i-Mode is in principle "always on", provided you are in an area where the i-Mode signal can reach you. When you select an i-Mode item on the handset menu, the data are usually immediately downloaded. There is no delay for dialing to set up the connection. However, there is a delay for the data to reach you. For now this service is available in Japan only but DoCoMo is planning to expand it to Europe and North America and it is investing large amounts of money in this regards.

3.3. Palm.Net:

Palm.Net is a wireless communication service that provides internet access to the Palm VII and Palm VIIx hand held device. Palm.Net service offers coverage in over 260 metropolitan areas in the U.S. It has a transfer rate of up to 9.6 Kbps. Users are charge according to the amount of data that is transferred. It utilizes web clipping and not web browsing unlike its counterpart OmniSky.

3.4. OmniSky:

OmniSky is full featured integrated wireless service, which provides internet access to Palm V users. In order to do this the Palm device is hooked to a wireless OmniSky Modem called "Mistrel V". The OmniSky software is then run on the Palm V, Which makes it capable of browsing the web. AT&T Wireless is the preferred back bone of OmniSky wireless service. However OmniSky is also striving to be provider independent, also offering services with Bell Atlantic, GTE and Ametitech1.

Unlike most wireless service providers, OmniSky charges a flat monthly rate for its CDPD (Cellular Digital Packet Data) network service. It works as a two way radio section offering connection speeds of up to 19.2Kbps. In comparison to the Palm.Net it is much faster. And unlike Palm.Net it supports full web browsing and access to other POP email accounts.

3.5. WAP

Unlike i-Mode, Palm.Net and OmniSky which are basically services WAP is a true wireless protocol. It is based on the OSI Reference Mode. It works on any digital bearer like GSM, GPRS, CDMA, TDMA etc. It has a data transfer rate depends on the bearer being used. On GSM networks it has a communication speed of 10 Kbps where as on GPRS networks, the through put is 175Kbps. Details about WAP will be discussed in later chapters.

3.6 Comparison:

Let us now take a look at the major differences between the technologies describes above.

- The first major difference is the programming language used. WAP uses the markup language WML (Wireless Markup Language) while i-Mode uses CHTML (Compact HTML). Incase of OmniSky the normal HTML pages on the internet a made available after conversion to a suitable format that is viewable on the handheld device. While Palm.Net uses Web Clipping meaning it edits the HTML content and just displays a small part of it.

- Another difference is that WAP is bearer independent. It can work on any digital cellular network. While the other technologies are network dependent e.g. i-Mode only works on NTT DoCoMo’s PDC-P network while OmniSky uses CDPD network.
- A major difference between WAP and i-Mode is that i-Mode is an “Always on” connection. This mean that you do not have to dial up to access a site and email is instantly sent to your phone. But in case of WAP the is a dial up connection if a GSM network is used. If a GPRS network is used then WAP also has a “Always on” connection.
- There is a marked difference in the case of billing procedure. I-Mode and Palm.Net all charge according to the amount of data transferred. OmniSky a flat amount is charged per month. While in case of WAP the billing depends on the network being used. For example if a GSM network is the bearer then the charging is done on the bases of time but if a GPRS network is used the billing done according to the data transferred.

Chapter

4

WAP Architecture

4.1 The World-Wide Web Model

The Internet World-Wide Web (WWW) architecture provides a very flexible and powerful programming model (Figure 4.1). Applications and content are presented in standard data formats, and are *browsed* by applications known as *web browser*s. The web browser is a networked application, i.e., it sends requests for named data objects to a network server and the network server responds with the data encoded using the standard formats.

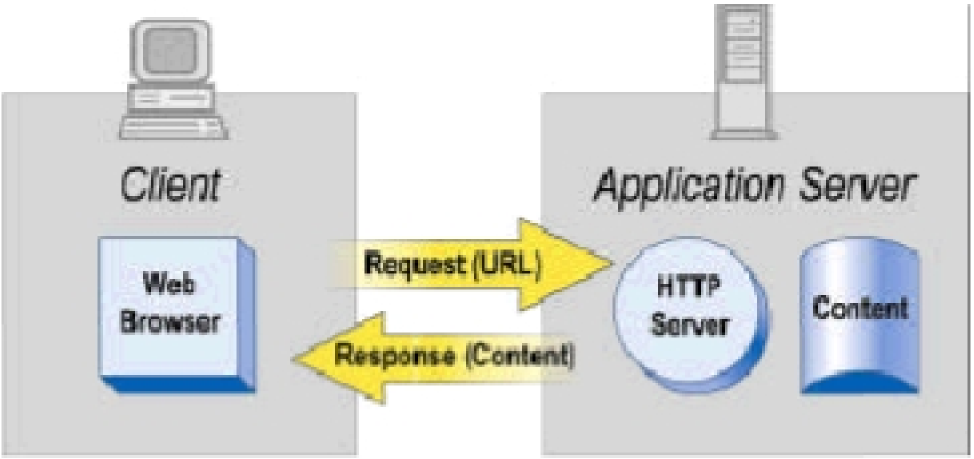

Figure 4.1 World Wide Web Programming Model

4.2 The WAP Model

The WAP programming model (Figure 4.2) is the WWW programming model with a few enhancements. Adopting the WWW programming model provides several benefits to the application developer community, including a familiar programming model, a proven architecture, and the ability to leverage existing tools (e.g., Web servers, XML tools, etc.). Optimizations and extensions have been made in order to match the characteristics of the wireless environment. Wherever possible, existing standards have been adopted or have been used as the starting point for the WAP technology.

□

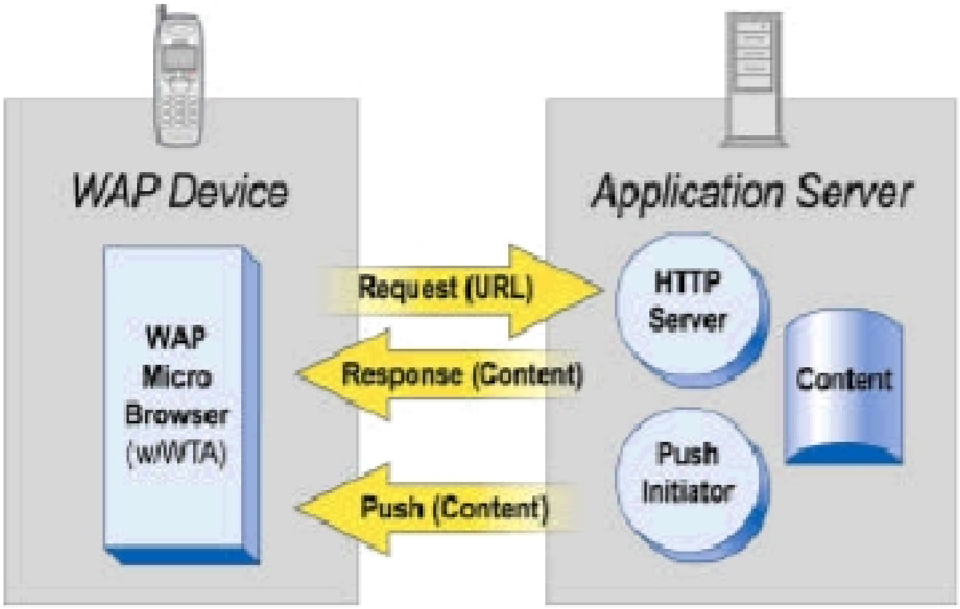

Figure 4.2. WAP Programming Model

The WAP protocols were designed with the web protocols in mind. The goal of WAP was to use the underlying web structure, but to render communication between content providers and mobile devices more efficient and less time consuming than if the web protocols themselves were used. In this section, we will start introducing the elements involved in mobile communications, and their role in the whole picture.

There are basically two ways in which information can be accessed on the internet using a WAP device.

- **WAP** used to access the Internet
- **WAP** used to access an intranet

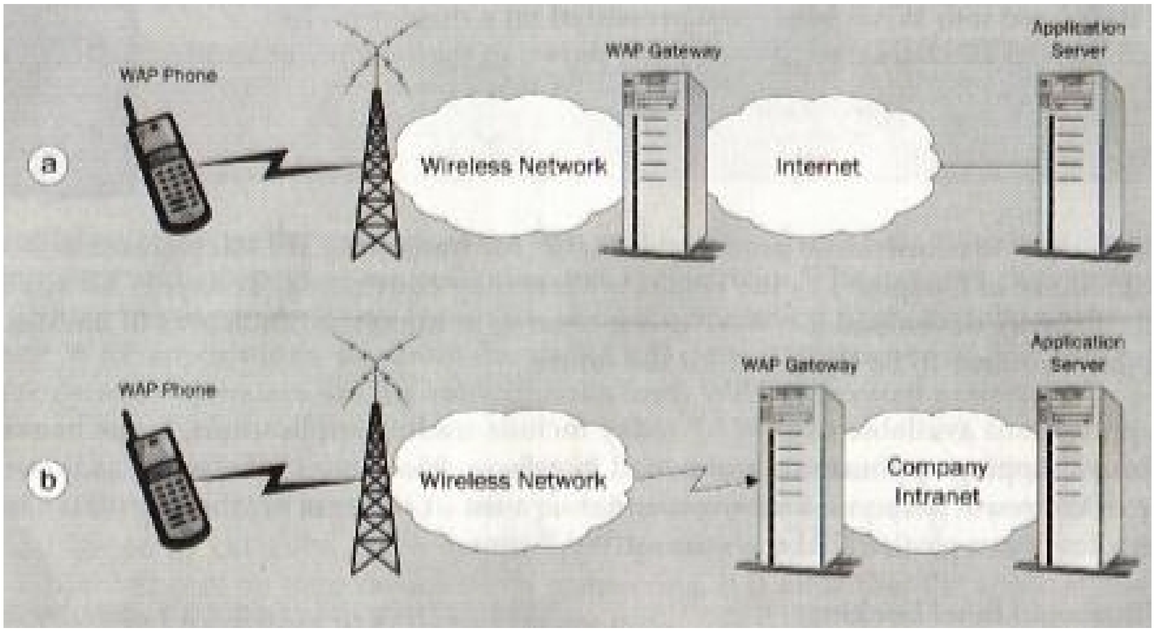

Figure 4.3

4.3 WAP Components:

Before continuing we should first need to understand the following terms which are essential for the working of WAP.

- **WAP Device:** This term indicates the physical device that you use to access WAP applications and content. It doesn’t necessarily have to be a mobile phone; it might be PDA or a handheld computer. More generally, it is every WAP compliant device.

- **WAP Client:** In a network environment, a client is typically the logical entity that is operated by the user and communicates with the ‘server entity’. In the WAP world, the client is the entity that receives content from the Internet via a WAP Gateway. This is usually (but not necessarily) the WAP browser. Commonly, ‘WAP client’ and ‘WAP browser’ are often used interchangeably. WAP Client is discussed in detail in the section 4.5.

- **WAP Browser:** This is software running on the WAP device. It receives content arriving from the Internet and decides how to display it on the screen of the WAP device. WAP browsers are available for all WAP devices, and are frequently referred to as Microbrewers. There are also emulators available for some browsers, which run on PCs.

- **User Agent:** An agent is normally the software that deals with protocols, and WAP is no exception to this. The WAP client contains two different agents: the WAE User Agent and the WTA User Agent.

- **WAP Gateway:** This is the element that sits (Logically) between the WAP device and the origin server. It acts as an 'interpreter' between the two, enabling them to communicate. It usually resides within the operator network, but you can also install your own gateway, as we will see later. Unless otherwise stated, when a gateway is discussed, we mean a gateway residing in the operator network, since this is the more common situation that one encounters. Discussed in detail in section 4.6

- **Network Operator:** This is the company or organization that provides carrier services to its subscribers. As an example, the company you are paying your telephone bills to is your network operator. A network operator enables you to make calls to other phones from your telephone and, in addition, provides you with different services, such as voice mail, call diversion etc.

- **Bearer Services:** These are the different ways that a mobile phone can communicate with the wireless network. To send and receive data from an application server, mobile phones have to establish some sort of connection with the WAP gateway. A bearer service is the method they use to do this. In GSM networks, for example, we either use SMS (Short Message Service) or CSD (Circuit Switched Data). With the former bearer, the gateway has to divide the information that is to be sent to the phone into a lot of little message (just like when you send a text message to a friend using your mobile). With CSD, we communicate with the gateway using a data connection, which is not dissimilar to the way the modem in your computer communicates with the Internet Service Provider that you have an account with.

- **Content / Origin / Application Server:** These three names are used interchangeably. They denote the element that hosts the Internet content that is sent to clients when they make a request for it. A web server is an origin server, providing HTML content (but also WAP content if properly configured).

4.4 The Working:

To access an application stored on the server, the client initiates a connection with the WAP gateway, and sends a request for content. The gateway converts the requests coming from the WAP client into the format used over the Internet (HTTP) and the forwards them to the origin server. On the way back, the content is sent from the server to the gateway, which then translates it to WAP format, and then sends it to the mobile device. The gateway allows the Internet to talk to the wireless network.

The concept of connection is left deliberately vague, since the goal of WAP is to provide a protocol that is able to adapt to any type of mobile network. The connection is established between the WAP phone and the gateway by means of the bearer used. Whether we are accessing WAP services by sending data packets or SMS messages, we see the same functionality. It may however, affect the speed of the connection and therefore affect the cost of the connection, but this is less important to the developer.

As is the case with the Internet, content servers host the content or applications, but the case of WAP these are sent to the clients as WML and WML Script files, rather than HTML etc. WML (Wireless Markup Language) and WML Script are the languages used to design and write WAP content. WML has some similarities to HTML and XML, and WML Scripts do not differ much from JavaScript. These will be discussed in detail in the later chapters. The WML and WML Script files are sent, on request, to the WAP client via a WAP gateway, which translates the content into a form that is optimized for the narrow bandwidth radio interface. The client contains a micro browser that displays the received information to the user. Let us now look at some of the important components of WAP architecture in detail.

4.5 WAP Client:

The **WAP** specifications leave a great deal of autonomy to the device manufacturers. There is no WAP specification indicating what the WAP device should look like or how it should present and display the content it receives from the Internet. These kinds of decisions, together with those relating to the user interface and the internal organization of phone functionality such as the phonebook, are left to the vendor. The only requirement for a device to be WAP compliant is that it must implement a **WAE User agent**, a **WTA User Agent** and the **WAP Stack**.

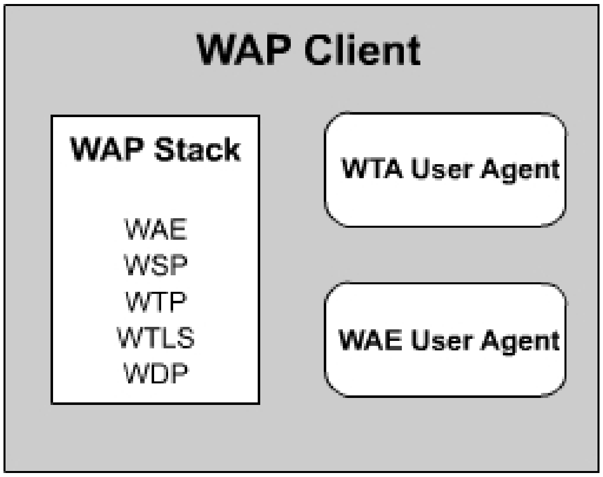

Figure 4.4 WAP Client

- The **WAE User Agent** (Wireless Application Environment User Agent) is the micro browser that renders the content for display. It receives the compiled **WML**, **WML** Script, and any images from the WAP gateway, and executes or displays them on the screen. Even if the implementation details are left to the vendor, the browser must implement all the functionality provided by WML and **WML** Script. It must also manage the interaction with the user, such a text input, and error or warning messages.

- The **WTA User Agent** (Wireless Telephony Applications User Agent) receives compiled WTA files from the WTA server and executes them. The WTA User Agent includes access to the interface to the phone, and network functionality such as number dialing, call answering, phonebook organization, message management and location indication services, which we discussed earlier.

- The **WAP Stack** implementation allows the phone to connect to the WAP gateway using the WAP protocols. We'll be looking at all the WAP protocols in detail later in this chapter 5.

4.6 WAP Gateway:

Usually in reading about WAP the terms *WAP gateways, servers and proxies* are used interchangeably, but this is not correct. On the contrary, in the world of networks these three elements are quite different logically and they have different functionalities as well;

- **Content /Origin / Application Server**: This is the element in the network the information or web / WAP applications reside. (Web servers belong to this category)

- **Proxy:**This is an intermediary element, acting both as a client and as a server in the network. It is located between clients and origin servers; the clients send requests to it and it retrieves and caches the information needed by contacting the origin servers.

- **Gateway:** This is an intermediary element usually used to connect two different types of network. It receives requests directly from the clients as if it actually were the origin server that the clients want to retrieve the information from. The clients are usually unaware that they are speaking to the gateway.

A WAP gateway form the bridge between the Internet (or another IP packet network) and the wireless phone/data network, which are fundamentally different in their underlying technologies. Figure 4.5 shows a WAP gateway, together with other elements in the

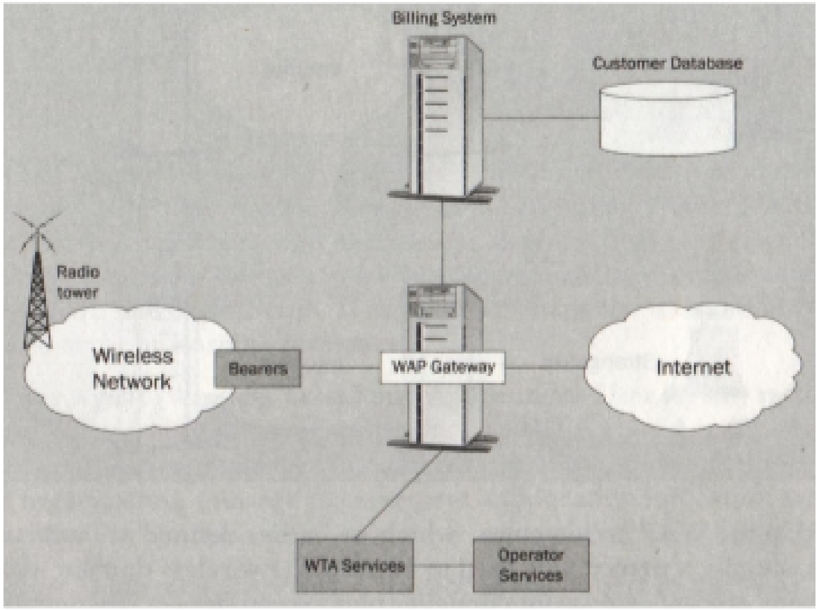

Figure 5.5 WAP Gateway & Wireless Network Elements

wireless network. This highlights how the WAP Gateway has to collaborate and interface with all the other elements in order to provide a proper service.

The gateway is basically a software that is placed between a network that supports WAP and an IP packet network such as internet. It acts as an intermediary that converts between the protocols of packet networks and the protocols on the WAP network. Whenever you start a WAP session on your mobile phone, the following steps are executed. (The details of the WAP protocols are dealt with in the next chapter).

- A connection is created via WSP (Wireless Session Protocol) between the mobile device and the WAP gateway, which we assume is present in the operator network.

- Are you enter the address of a WAP site (by typing it or selecting a bookmark, for example), the gateway is sent a request from the device's micro browser using WSP. WSP is the WAP protocol in charge of starting and ending the connections from the mobile devices to the WAP gateway.

- The gateway translates the WSP request into an HTTP request and sends it to the appropriate origin server.

- The origin server sends back the requested information to the gateway via HTTP.

- The gateway translates and compresses the information and sends it back to the micro browser in the mobile device.

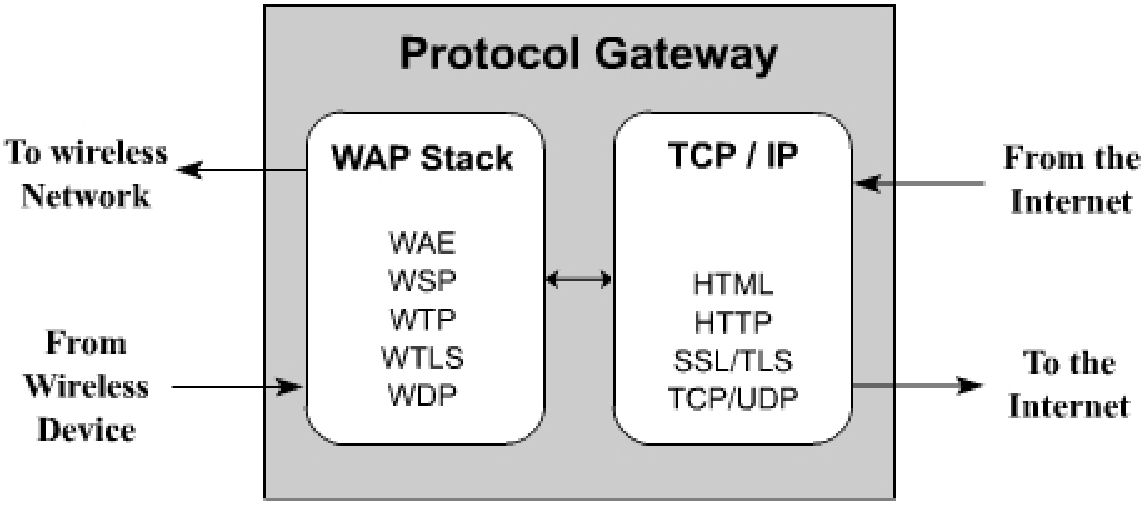

Figure 4.6 Protocol Gateway

The gateway part of the WAP proxy takes care of translating all the requests that are sent and received by the client using WSP to the protocol that the origin server is using (HTTP for example). This is illustrated in the Figure 4.6. The content provider sends its content using HTTP to the gateway. It then forwards all the content received to the WAP devices, using the WAP protocols

Functionally speaking, the gateway operates to some extent in a similar way to the current Internet web browsers. When you try to access an FTP or Gopher site using your web browser, you are completely shielded from the protocols and requests that your browser uses to contact the site. As far as you are concerned, both FTP and Gopher sites use the same protocol to communicate with the browser as a normal web site, since the information that is displayed on your screen is in the same format as when you access an HTML page.

The coder / decoder (CODEC) functionality within the gateway is used to convert the WML and WML Script content going to and coming from the client into a form that is optimized for low bandwidth networks.   This is illustrated in the Figure 4.7.

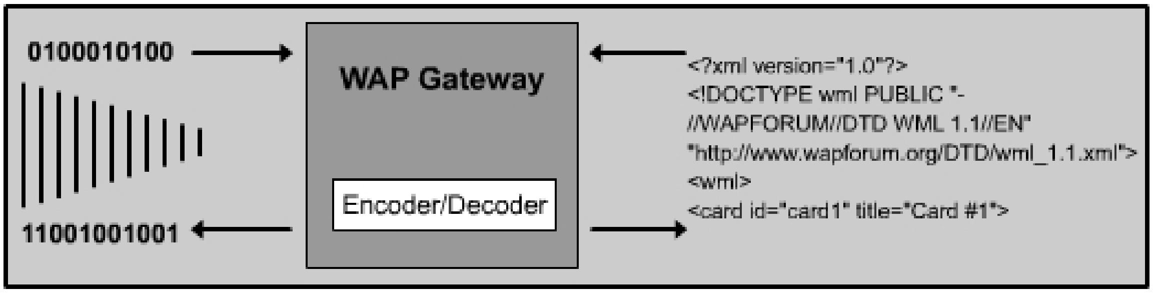
0100010100
11001001001
WAP Gateway
Encoder/Decoder
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-
//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="card1" title="Card #1">

Figure 4.7 Gateway CODEC

4.7 Functionality of WAP Gateway:

Functions that WAP gateway perform is as under described in detail. Some of these functions are optional. This means that some gateway products may not provide those functions which are not mandatory (optional functions are explicitly maintained, other wise they are assumed to be mandatory).following is the summary of the functions that a gateway perform

- Implement WAP protocol stack
- Access control
- Protocol conversions: WSP⬄HTTP
- Domain name resolution
- HTML to WML conversions
- Encoding of WML content
- WML script compilation
- Security
- Provide caching for freely accessed content

4.7.1 Implementation of WAP Protocol Stack Layers:

This is the most obvious function of WAP gateway and it contributes to the most of the functions of a gateway. Depending on whether the type of service is connection-oriented or connectionless, secure or not secure, the following stack layers need to be implemented:

- ➢ Non-secure connection-oriented: WSP⬄WTP⬄WDP
- ➢ Secure connection-oriented: WSP⬄WTP⬄WTLS⬄WDP
- ➢ Non-secure connectionless: WSP⬄WDP
- ➢ Secure connectionless; WSP⬄WTLS⬄WDP

4.7.2 Access Control:

This involves restricting specific content (like subscription services, or company intranet WAP services). Recognition of the device could be based on IP address or MSISDN number (its phone number). This depends on bearer being used. For example ,a WAP gateway using SMS as the bearer will use phone numbers because this information is the only one available to identify the device .in the of an IP bearer ,the gateway does not have an access to the phone number of the device if there is one. In addition, not all devices have IP addresses, depends on the bearer being used. If a bearer such as SMS is used, there is no need of the device to have an IP address associated with it .even devices that use an IP bearer will typically not have IP addresses associated with it on a permanent basis. For example, if an IP is used over circuit switched data through an ISP, the remote access server may have dynamically assigned an IP address while you were connected. However, in the case of wireless packet networks like GPRS, an IP address (or X.25 addresses if GPRS is used for X.25 services) will always be associated with the device as long as the device is switched on and subscription for the services exists, even when the device is not transmitting or receiving packets.

A more fine-grained access control can be achieved by using user authentication. This could use the HTTP basics or proxy authentication mechanism. It would not only control which devices were to retrieve content through the gateway ,but also control that content is made available to each device

4.7.3 Protocol Conversion: WSP ⇔ HTTP:

WSP support complete HTTP/1.1 functionality. This includes extensible request-reply methods (like GET, POST, etc), request, response and entity headers (like “accept: application/vnd.wap.wml”, a request header that specify the particular MIME types that a client can handle) and content negotiation content negotiation is the process of selecting the best representation suited for a client for a given response when there are multiple representation for the same content available from different server.

A request header is meta-information that is sent along with a HTTP request (like GET or POST requests). Similarly, a response header is meta–information in a HTTP response that is sent by the server as a response to a previous HTTP request. As part of the HTTP response, the server might also send an entity body (an HTML file for example) depending on the type of request. The meta–information sent to give more meaning to the entity body that was sent is known as an entity header

However, WSP headers are in a compact binary tokenized form as defined in the WSP specification. A token is a group of characters that has a specific meaning when used together as a string. For example, in the accept header as you see below, “accept:”, “text/plain” “text/vnd.wap.wml” etc. For all the string tokens, a binary token for these would an octet representation.

For example, the HTTP /1.1 request header below uses 122 octets (or bytes in other words):

```
Accept: text/plain, text/vnd.wap.wmlscript,
application/vnd.wap.wmlc, application/vnd.wap.wmlscriptc
```

The above request header indicates to the server that client can accept content in any of the above MIME formats, plain text, WML in both compact and text form, WMLscript as text and in its compact form

Using WSP, the same header is represented with just five octets:

```
0x80 0x83 0x88 0x89 0x94 0x95
```

For MIME types are not defined in WAP specification, encoding is done and the textual headers are sent as is.

4.7.4 HTML to WML Conversion:

One thing that should be kept in mind is that this is a optional feature. This conversion can never be perfect, and it can never be guaranteed that after conversion of an HTML page will be rendered properly on a wireless device.

4.7.5 Encoding of WML Content:

WML content coming from the Internet or another provider is encoded into a compact binary form at the gateway before it is sent to the wireless device. This process is called tokenization. During the process the gateways also performs check to verify that the WML content has no errors and is well formed. Incase where the verification fails, the gateways sends an error message to the user agent on the wireless device. With this mechanism, the user agents on the device can assume all the WML they receive is well formed and avoid complex error handling implementations that might have otherwise made the user agent consume more resource on the device.

4.7.6 WMLScript Compilation:

The compilation of WMLScripts on the gateway involves syntax and semantic checks, and the generation of byte code according to the WMLScript Instruction Set.

4.7.7 Security:

This involves providing the WTLS, between the gateway and the wireless device, and SSL between the gateway and the HTTP origin server. This is also an optional feature. It may be necessary to use a gateway product that implements security features, depending on the kind of content provided.

4.8 WAP Gateway Configuration:

From the discussion so far, it is clear that WAP gateways sits somewhere between the wireless device and the HTTP origin server. We will now evaluate in more detail the possible location in which a WAP gateway can be situated and the advantage and disadvantages of each choice.

Here is a summary of the useful locations of a WAP Gateways

- A WAP Gateway provided by the network operator
- A WAP Gateway Provided by the content provider

- A WAP Gateway Provided by the ISP

4.8.1 Gateway Provided By the Network Operator:

We assume here that the bearer is IP over dialup circuit Switched. In the figure 4.8, it is clear that in this case the WAP gateway is part of the infrastructure that belongs to the cellular network operator. The WAP gateway houses a pool of modems that take up phone ports on the mobile switch. The phone user would have to configure their phones with the access phone number(s) to dial in to the Remote Access server setup.

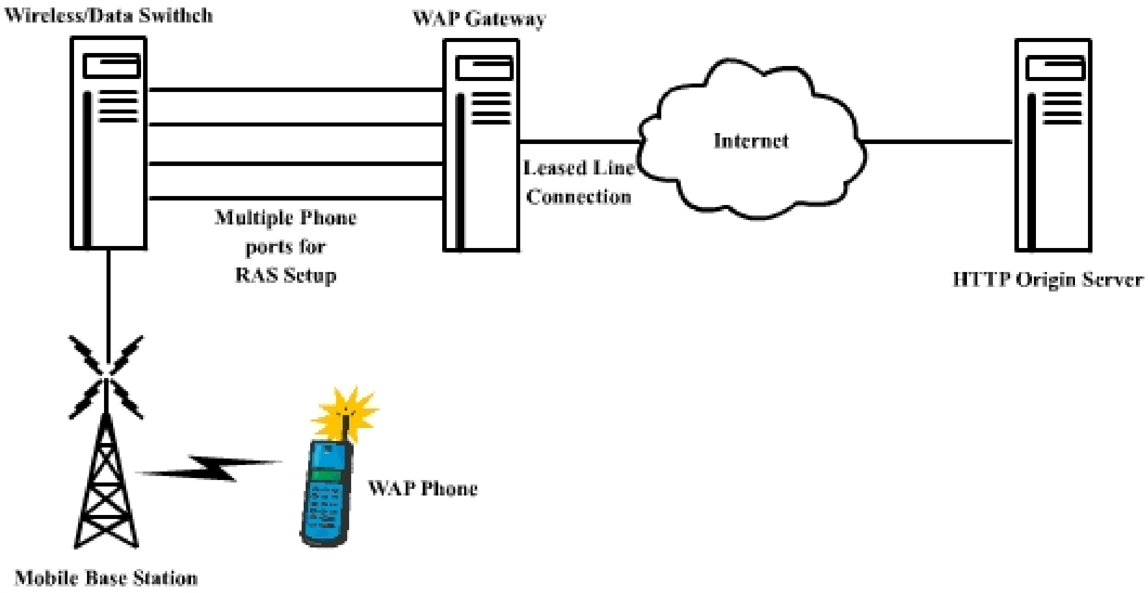

Figure 4.8

Even in the case of packet switched network as GPRS, the network operator can put up a WAP Gateway just before the operator infrastructure terminates on the internet.

One of the advantages of the above setup is:

- The mobile device only needs a single gateway setting to access any internet content. The user simply fires up the micro browser in the mobile device to access information without having to fiddle with the configuration, since it is likely that the device come preconfigured when the mobile was bought in the packaged subscription. The network the network operator could also use an Over-the-Air (OTA) configuration, using SMS to do this job. OTA provisioning strings are quite specific to each mobile vender, thought many WAP gateway venders do support this feature.

Disadvantages for this scenario are:

- The network operator might introduce additional advertising content which will piggyback on the content from the Internet when it is sent to the wireless device. The content provider on the internet probably not like the idea of their content being obscured in this way
- Even if the secure HTTP and SSL are used between the WAP gateway and the application server as well as WTLS between the user agent and the gateway, the requested content will be in an unencrypted form in the main memory and disk cache of the WAP gateway. Almost every operating environment provides mechanism to read the memory of the existing process as long as administrative privileges are present. Because gateways cache content after it is received at the WSP layer, data will not be encrypted when stored on the disk cache. This could cause security problems, which will be a cause of concern on some cases such as banking applications. However as long as a non-disclosure agreement for the data exists between the content provider and the network operator, this will not be an issue.

The network operator may chose to block access to all but a few ‘approved’ WAP sites. Some subscribers may protest, move to competitor or even use internet access provided by an ISP to access a public WAP gateway. However most subscribers either will be unaware of how to bypass the WAP gateway in the preconfigured settings on their mobile device, or not interested in fiddling with the provided settings. Many of them may not even bother to try to access services/hyperlinks other than the ones provided by an ‘approved’ WAP portal. Eventually however competition and awareness on the part of the subscribers should level the playing field.

4.8.2 A WAP Gateway Provided By the Content Provider

A gateway is shown in figure 4.9 is part of the infrastructure of a content provider:

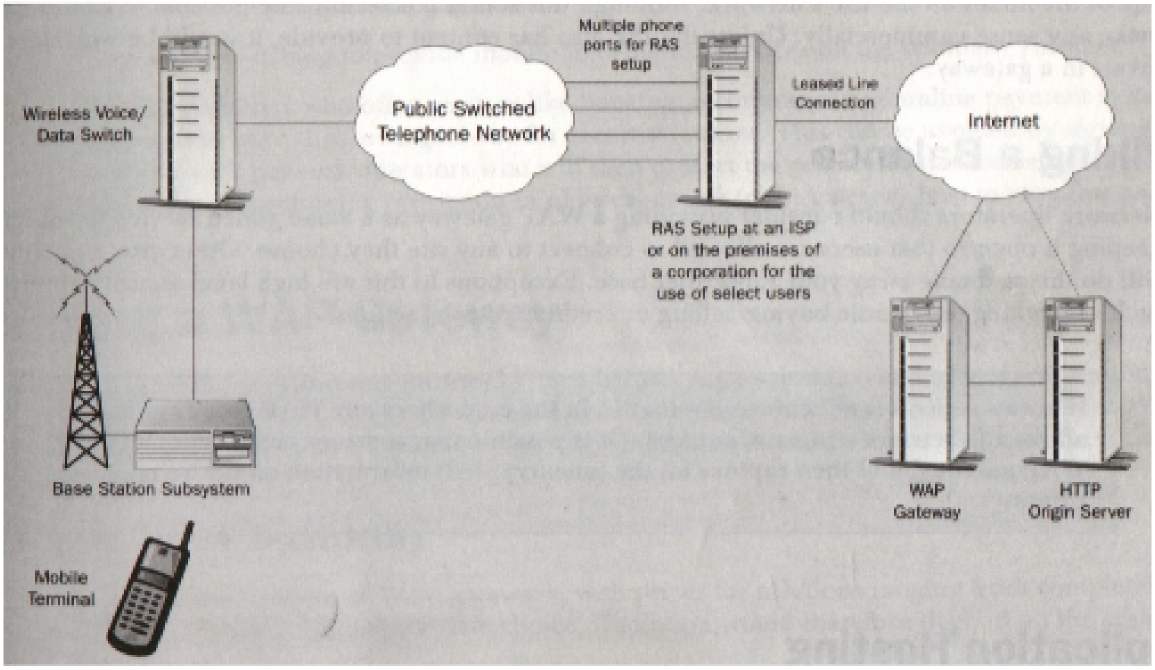
Multiple phone ports for RAS setup
Wireless Voice/ Data Switch
Public Switched Telephone Network
Leased Line Connection
Internet
RAS Setup at an ISP or on the premises of a corporation for the use of select users
Base Station Subsystem
WAP Gateway
HTTP Origin Server
Mobile Terminal

Figure 4.9

A content provider might decide to have its own WAP gateway at its web farm (the location where the clusters of origin servers are located). They would then advertise the configuration information needed to use their gateway.

One advantage of this solution is:

In the case of the secure application, like banking, access to the content on the origin servers through a WAP gateway other than the content providers own could be disabled for the sake of security. If mobile users try to access the secure content through another WAP gateway, they could be presented with the message asking them to configure their device to use the content provider’s gateway. In this manner, all data can be transfer between the device and the gateway using WTLS and the unencrypted data is present only on the private network of the content provider.

The disadvantage include;

- If every content provider choose to adopt this solution ,it will be impossible for the mobile user to have all the necessary gateway configuration set up on their device .for instance,Nokia 7110 supports up to maxmimum 10 gateway configurations. The user will also need to switch configuration every time they need to access a different WAP site.

4.8.3 A WAP Gateway provided by the internet service provider (ISP):

An ISP could also host a WAP gateway. The architecture is very similar to the previous case except for the position of the gateway. The gateway will now be closer to the RAS server and will be one of the nodes on the ISP's network. Although this solution is technically possible, it would hardly make any sense commercially. Unless the ISP also has content to provide, it would be worthless to invest in a gateway.

Chapter 5

Protocol Stack

5.1 WAP Protocol Stack:

In the next sections, we will look at how the WAP protocol is structured and how the different WAP layers map into Internet protocol layers.

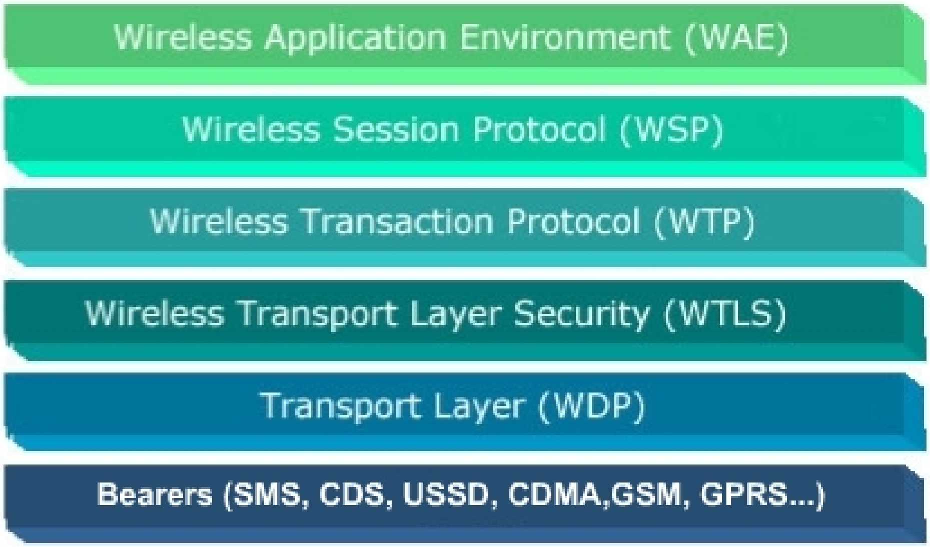
Wireless Application Environment (WAE)
Wireless Session Protocol (WSP)
Wireless Transaction Protocol (WTP)
Wireless Transport Layer Security (WTLS)
Transport Layer (WDP)
Bearers (SMS, CDS, USSD, CDMA,GSM, GPRS...)

Figure 5.1 WAP Protocol Stack

The WAP stack, illustrated in the Figure 5.1, has 5 different layers:

- **Application Layer**: WAE (Wireless Application Environment) provides and application environment intended for the development and execution of portable applications and services.

- **Session Layer**: WSP (Wireless Session Protocol) supplies methods for the organized exchange of content between client / server applications.

- **Transaction Layer**: WTP (Wireless Transaction Protocol) provides different methods for performing transactions, to varying degree of reliability)

- **Security of Layer**: WTLS (Wireless Transport Layer Security) is an optional layer that provides, when present, authentication, privacy and secure connections between applications.

- **Transport Layer**: WDP (Wireless Datagram Protocol) is the bottom layer of the WAP stack, which shelters the upper layers from the bearer services offered by the operator.

The WAP stack was derived from, and inherited most of the characteristics of, the ISO OSI reference model (ISO 7498). The main difference between the two is the number of layers: WAP has just five layers, while the OSI model has seven of them.

Now let us discuss each layer in detail.

5.1.1 Wireless Application Environment (WAE):

The application layer of WAP provides and environment that includes all the elements related to the development and execution of applications. The Wireless Application Environment (WAE) allows the developer to use specific formats and services, created and optimized for presenting content and interacting with limited capability devices. WAE consists of two different user agents located on the client side, the WAE user agent – including the micro browser and the text message editor – and the WTA user agent.

The WAE specifications say nothing about the implementation of the user agents. All the browsers, message editors, and phonebooks contained in WAP devices can vary greatly while still complying with the specifications. WAE formally specifies just the formats, such as images and text formats, which the user agents have to be compliant with. This is an important characteristic of WAP in general, as we will soon see when looking at WML. Beginning with WAP 1.2, there will also be another scenario; it will be possible to push content towards a WAP client without any request made by the client. The main building blocks of the WAE are the following:

- A lightweight markup language: WML
- A lightweight scripting language: WMLScripts
- An interface to local services and advanced telephony services: WTA (not yet implemented)

5.1.2 Wireless Session Layer (WSP):

The Wireless Session Protocol enables services to exchange data between applications in an organized way. It includes two different protocols:

- **Connection oriented Session Services** - Operates over the Wireless Transaction Protocol (WTP)

- **Connectionless Session services** – operates directly over the Wireless Transport layer (WDP)

Session Services are those functionalities that help to set up a connection between a client and a server. A service is delivered through the use of the primitives it provides. **Primitives** are defined messages that a client sends to the server to request a service facility. In WSP, for example, one of the primitives is S-Connect, with which we can request the creation of a connection with the server.

The **Connection – oriented** Session service provides facilities used to manage a session and to transmit reliable data between a client and a server. The session created can then be suspended and resumed later if the transmission of data becomes impossible. Also, once the push technology takes off, unsolicited data can be pushed from the server to the client in a confirmed or unconfirmed way. In **Confirmed push** the server is notified upon reception of the data by the client, in **unconfirmed push** the server is not notified of the reception of the pushed data. Most of the facilities provided by the connection-oriented session service are confirmed, meaning that the client can send Request primitives and receive Confirm primitives and the server can send Response primitives and receive Indication primitives. The **Connectionless** session service provides only non-confirmed services; in particular only unreliable method invocation (asking the server to execute an operation and return a result) and unconfirmed push are available. In this case clients can only use the Request primitive and servers are only able to use the Indication primitive.

To start a new session, the client invokes a WSP primitive that provides some parameters, such as the server address, the client address and client headers. These can be linked to HTTP client headers and can, for example, be used by the server to retrieve the type of user agent within the WAP client (which might be both the version and type of the browser). This is useful when we want to format the output differently, depending on the client's device type. For example, one phone may have a 20 character wide display; another may have a 16 character wide display. In some respects WSP is basically a binary form of HTTP. As previously mentioned, the binary transmission of data between a server and a client is an essential adaptation made for the narrow bandwidth mobile network. WSP supplies all the methods defined by HTTP/1.1 and allows capability negotiation to gain a full compatibility with HTTP/1.1.

5.1.3 Wireless Transaction Layer (WTP):

The Wireless Transaction Protocol provides services to accomplish reliable and non-reliable transactions and operates over the WDP layer or over the optional security layer, WTLS. WTP, as all the other layers in WAP, is optimized to adapt to the small bandwidth of the radio interface, by trying to reduce the total amount of replayed transactions between the client and server. In particular, three different classes of transaction services are supplied to the upper layers:

- **Unreliable Request:** The initiator (in this case a content server) sends a request to the responder (the user agent) who does not reply with an acknowledgment. The transaction has no state and terminates once the invoked message is sent:

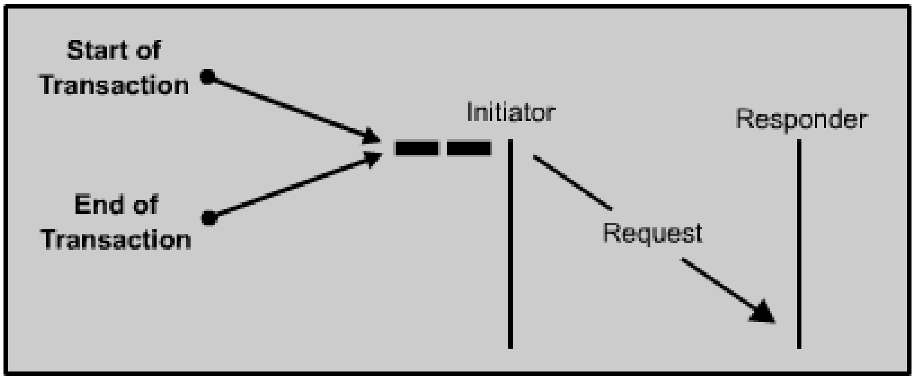

Figure 5.2 Unreliable Request

- **Reliable Request:** The initiator sends a request to the responder who acknowledges it. The responder stores the transaction state information for some time, so that it can re-transmit the acknowledgement message if the server requests it again. The transaction ends at the initiator when the initiator receives the acknowledgement message:

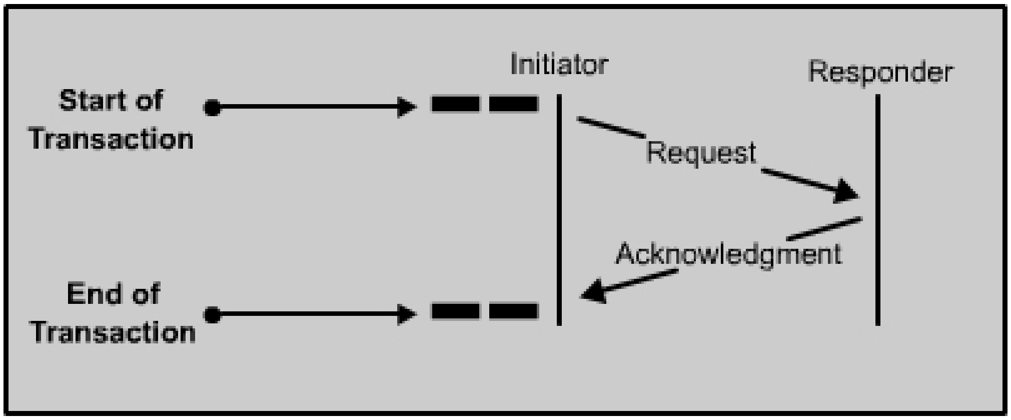

Figure 5.3 Reliable Request

- **Reliable Request with One Result Message:** The initiator sends a request to the responder who implicitly acknowledges it with a result message. The initiator then acknowledges the result message, maintaining the transaction state information for some time after the acknowledgment has been sent, in case it fails to arrive. The transaction ends at the responder when it receives the acknowledgement message.

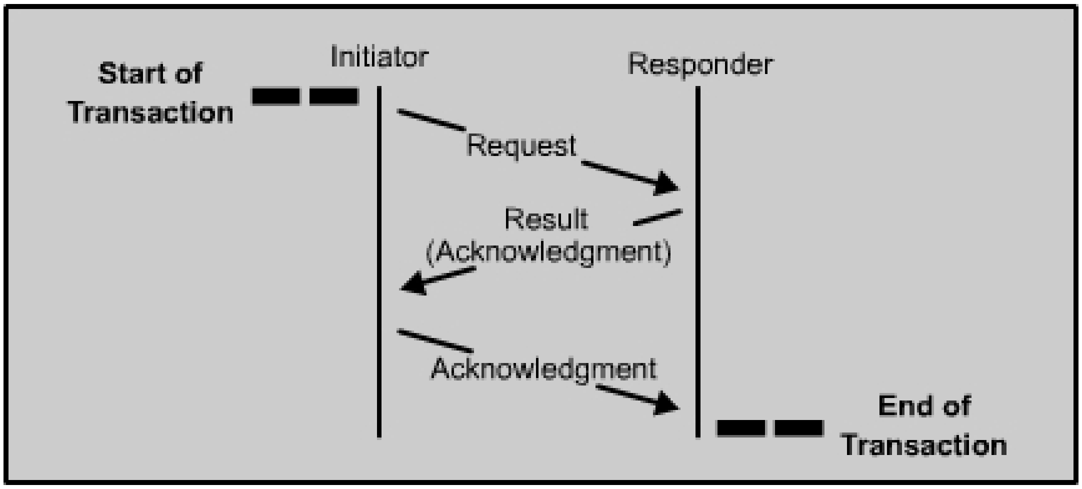

Figure 5.4 Reliable Request with one Result Message

5.1.4 Wireless Transport Layer Security (WTLS):

WTLS is the solution to the security issue, provided by WAP Forum. WTLS is an optional layer and is based on TLS (Transport Layer Security) v1.0, which in turn is based on SSL (Secure Sockets Layer) v3.0, which are Internet protocols. WTLS operates over the transport layer (WDP).

During the past few years, security over the Internet has become a big issue. E-commerce, e-banking and e-trade experienced a big evolution once SSL was standardized. By providing guaranteed privacy, confidentiality and authenticity over the TCP protocol, SSL enabled commercial solutions to expand their services.

It should be obvious that WAP also had to adapt to this situation, by offering ways to protect, when needed, the data requested from or sent to the user. In WTLS we find the same fundamental characteristics we observed in all the previous layers in the WAP stack: it is an adaptation of an Internet protocol both to the high-latency, narrow-bandwidth air interface and to the limited memory and processing power of the WAP device. WTLS attempts to lighten the overheads associated with establishing a secure connection between two applications. It provides the same grade of security that is supplied by SSL 3.), while reducing the transaction times. It provides services that ensure **privacy, server authentication, client authentication** and **data integrity**.

- **Privacy** guarantees that the data sent between the server and the client is not accessible to anyone else. No one can read the unencrypted message, although they can see the encrypted message.

- **Server authentication** ensures that the server really is who it claims to be and that it is not an imposter.

- **Client authentication** provides a way for the origin server to limit the access to the content it provides. Just those subscribers that are recognized as trusted ones can gain access to the site.

- **Data integrity** takes care that no one can alter the content of a message being transmitted between server and client without one of them noticing.

In the Figure 5.5 that follows, we show how the WAP gateway handles secure sessions.

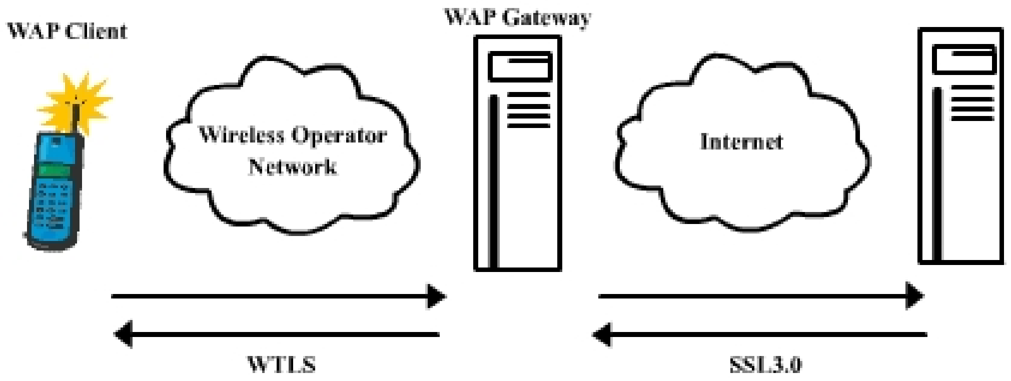

Figure 5.5

A standard SSL session is opened between the web server and the WAP gateway and a WTLS session is initialized between the gateway and the mobile device. The encrypted content is sent through this connection from the server to the gateway, which translates it and sends it to the mobile phone.

WTLS empowers the SSL protocol by adding effective features such as datagram support, optimized handshake and dynamic key refreshing. Today, WAP gateways are available which provide public / private key encryption with a key length up to 1024 bits. To use a secure connection the origin server has to be installed as if we were setting up a secure connection over the Internet, the gateway will take care of matching the SSL connection to a WTLS one. The translation between SSL and WTLS takes place in the memory of the WAP gateway. It is important that unencrypted information is not stored anywhere in the gateway, since this defeats all the security measures used to protect the stored data from being seen by unauthorized people.

Even though WAP gateways are provided with many features to supply the maximum level of security, there is still a lot of concern surrounding the WAP security solution. Banks and all the companies that really have to protect their data, still prefer to host and install their own gateway, giving them the ability to send encrypted data right to the mobile phones, with no need for translation. Time will show whether WTLS will be gradually adopted as the standard for if it will just be ignored.

WTLS is an optional layer in the WAP stack. This means that *security in* WAP *is only available on demand and is not a built in feature of the WAP architecture.* Hence, the information traveling to and from the WAP gateway is normally not encrypted, unless we use SSL connections to communicate between the origin servers and the gateway.

5.1.5 Wireless Datagram Protocol (WDP):

WDP is the bottom layer of the WAP stack and is one of the elements that makes WAP, the extremely portable protocol that it is, operable on extremely different mobile networks. WDP shields the upper layers from the bearer services provided by the network, so allowing the applications a transparent transmission of data over the different bearers. Bearer services are the nitty gritty of communication between the mobile phone and the Base Stations (the antennas). They include GSM, GPRS, SMS, CSD, USSD, DECT, and CDMA. The physical layer prepares the data to be sent from the mobile device over the air interface, and sends the data using the bearer service implemented in the network that the device is operating in.

5.2 WAP Vs TCP/IP:

The differences between the **WAP protocol stack** and a typical Internet protocol stack is the most important part of enabling wireless access for mobile devices. The WAP protocol stack does not map directly onto other stacks, although some comparison is possible. This is illustrated in Figure 5.6.

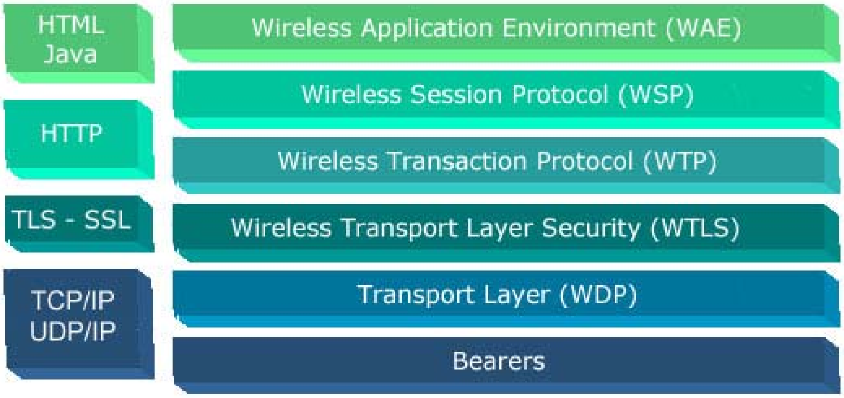
HTML
Java
HTTP
TLS - SSL
TCP/IP
UDP/IP
Wireless Application Environment (WAE)
Wireless Session Protocol (WSP)
Wireless Transaction Protocol (WTP)
Wireless Transport Layer Security (WTLS)
Transport Layer (WDP)
Bearers

FIGURE 5.6. WAP Protocol Stack

The kind of functionality that is provided by HTML and Java in the Internet world is incorporated in WAP in the WAE and, to some extent, the WSP. WSP and WTP between them cover the functionality that is provided by HTTP. If security is required in the fixed-wire world, Transport Layer Security (TLS) is usually used, and in the wireless world there is the **Wireless Transport Layer Security (WTLS)** equivalent.

The preferred transport layer is UDP over IP. However, many of the mobile networks cannot support IP at this point in time, so there is an additional protocol that allows transmission of packets over circuit-switched networks. This is the **Wireless Datagram Protocol (WDP).** Underlying WDP there is a large number of over-the-air bearers that are supported. Another point to note is that WAP can work on a number of bearer networks like GSM, GPRS, CDPD, CDMA, TDMA CSD, PDC-P, and SMS etc.

Chapter

6

6. WAP Security

6.1 Introduction:

In this chapter we will look at some of the security issues that impact on the WAP applications and the environment in which they will be deployed and used. We will also look at some of the potential solutions and technologies that can be used to address these issues. There is a set of concerns over how secure WAP is as a technology, and whether it is robust enough to implement mobile commerce applications, and other applications with stringent security requirements.

Before beginning this investigation of WAP security, it is worth noting that there is no such thing as a secure system. The phrase 'secure system' means one that cannot be compromised or accessed without authorization. Considering that hackers who set out to compromise or penetrate systems are resourceful and always target unexpected aspects of the systems, it would be a brave fool who declared a system to be immune to attack. What can be said is that a particular system meets certain predefined security criteria in that it can withstand attacks of a known type, and is therefore considered secure enough for its intended purpose.

6.1.1 Role of Security:

Security is both an enabling and disabling technology. Its purpose is to enable communications and transactions to take place in a secure environment without fear of compromise, while at the same time disabling non-legitimate activities and access to information and facilities. Non-legitimate activities include eavesdropping, pretending to be another party (also known as impostering or spoofing), or tampering with data during transmission. In general these activities are either unacceptable or illegal outside of the digital environment, so security simply helps to enforce the status quo in that sense.

6.1.2 Security Issues:

There are a number of basic issues around security that have to be addressed. Almost all of these have parallels in the real world, and often the solutions are based on, or similar to, real-world solutions. These basic issues are:

- **Authentication** – being able to validate that the other party participating in a transaction is who the party claims to be, or a legitimate representative of that party
- **Confidentiality** – being able to ensure that the content and meaning of communications between two parties do not become known to third parties
- **Integrity** – being able to ensure that messages received are genuine and have not been tampered with or otherwise compromised
- **Authorization** – being able to ascertain that a party wanting to perform some action is entitled to perform that action within the given context
- **Non-repudiation** – being able to ensure that once a party has voluntarily committed to an action it is not possible to subsequently deny that the commitment was given by that party

6.2 WTLS (Wireless Transport Layer Security):

WTLS is the **Wireless Transport Layer Security** protocol. As can be ascertained by the name, it operates at, or more correctly just above, the transport layer in the OSI protocol stack. It is based on transport layer security (TLS), which is the defacto security implementation on the Internet. It works by establishing a session between a client and a server (which in the case of WTLS is the WAP gateway), during which it negotiates security parameters to be used to protect the session. These include the encryption protocols to be used, signature algorithms, public keys, pre-master secrets, or the exchange of certificates, depending on the capabilities of both the client and the server and the required level of security. The process of establishing a session is called the **handshak**e. Once a session has been established all communications between the mobile device and the WAP gateway are encrypted, and therefore should be unintelligible if they are intercepted. WTLS includes support for both a full handshake, with negotiation of all security parameters, and for a 'lightweight' handshake in which the security parameters of another session are reused. Support is also provided for session suspend and resume, which is useful in a wireless environment where reception quality is not always that good and where connections can easily be lost. The sessions can continue to exist despite a terminated connection and can be resumed on reconnection. Using this facility, it is possible to have sessions that last for days at a time.

Another advantage of WTLS over TLS is that it operates over UDP. TLS requires a reliable transport protocol, in particular TCP, so it cannot be used over UDP. WTLS addresses this shortcoming, and also functions over WDP in the absence of UDP. WTLS therefore provides a comprehensive, optimised solution for both client and server based authentication using certificates, secure exchange of symmetric keys, anonymous and authenticated encryption of data, and support for digital signing of data. There are three classes of WTLS implementation defined in the WAP specification. They are:

- **Class 1**: Anonymous key exchange with no authentication.
- **Class 2**: Certificate based server authentication. Server key is anonymous or authenticated, client key is anonymous.
- **Class 3**: Certificate based client and server authentication. Both client and server keys are anonymous or authenticated.

6.3 Communication Models:

The best way to achieve an understanding of the merits of the implementation of security in the wireless environment is to compare it to the implementation of security in the fixed-wire world, that is, the Internet.

6.3.1 Internet Communication Models:

The Internet communication model assumes that a client PC connects to a server via an ISP dial-up connection. The client will be connected into the ISP systems over a PSTN or ISDN link, with PPP usually used as the bearer protocol. The connection point on the ISP network is to a RAS server, which will perform certain functions on behalf of the remote client. A typical example of the Internet communication model is shown in the in figure 6.1.

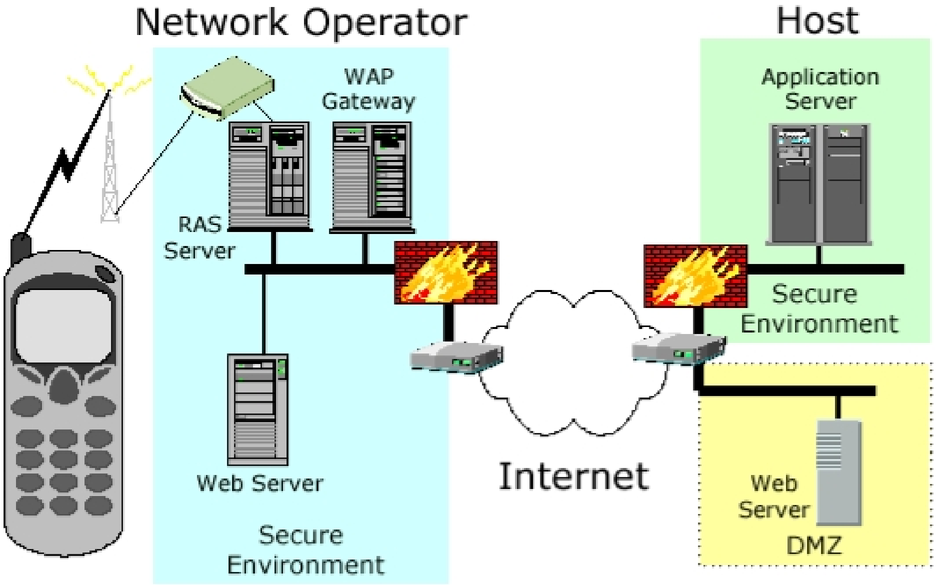

Figure 6.1. Internet Communication Model

The RAS server is responsible for validating the client that is dialing in, and there are various means at its disposal to do that. The RAS server is typically on a secure part of the ISP network and thus provides the illusion to all other devices that the remote client is in fact also on the local network. The ISP secure network environment is usually isolated from the Internet by means of a firewall of some sort. This firewall will attempt to regulate traffic that enters the local network, and protect the devices on the local network from malicious attacks over the Internet. The ISP may also choose to run one or more web servers and/or other facilities in a way that is more easily accessible to the public, and by extension also more vulnerable to attack. This area of the network is referred to as the **demilitarized zone** (DMZ), and is usually on a separate network segment from the secure area.

Access to the Internet is typically facilitated by one or more gateway devices, which are connected both to the ISP network and to some other network, possibly one run by one of the global Internet backbone providers. Any message entering the network across the gateway will be forwarded from gateway to gateway across the Internet, until it arrives at the destination network. It will then cross the gateway and enter the local network of the target host. In a way similar to the ISP, the host may also have a DMZ which houses the web server, with traffic entering the secure network filtered through a firewall.

In examining the Internet model from the perspective of who controls or has the ability to influence the connection from a security point of view, it is apparent that the TLS connection exists between the client device and the web server. In effect this forms a tunnel between the client and server, and anyone penetrating this tunnel would not be able to decipher any messages intercepted. The ISP retains responsibility for the devices on its own network and for validating that the client is permitted to connect to the network in the first place, but has no ability to influence the TLS session. The extent of each parties influence is illustrated in figure 6.2.

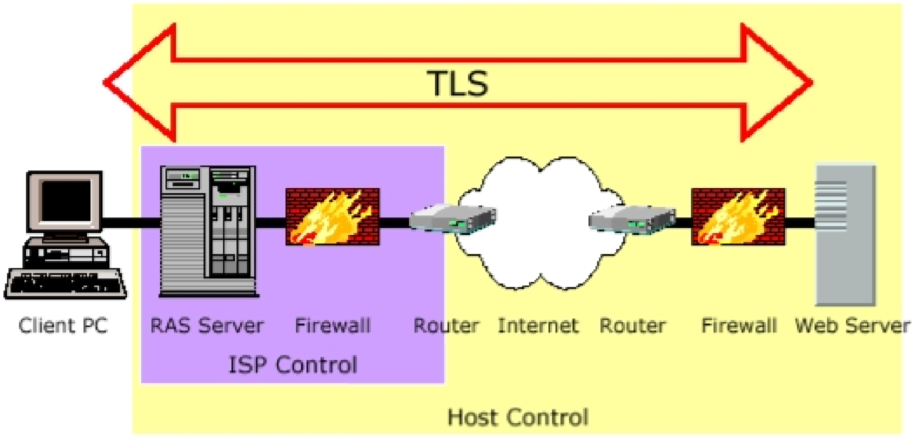

Figure 6.2

6.3.2 WAP Communication Models:

The wireless communication model is more complex because there are more ways in which the connection could be achieved. The model that we will examine at this point in time is one which many, possibly the majority of, connections that take place between the person-in-the-street and some WAP enabled web site will take place over. That is not to say that this is necessarily the best model from any particular point of view, just that many connections will be effected in this manner. This model is illustrated Figure 6.3. In this model the remote client is a mobile device, but still dials into an RAS server on some network somewhere. This is likely to be an RAS server hosted and owned by the network provider, and is therefore likely to be on the network provider's own local network. The network provider will typically also host the WAP gateway, and a web server to provide access to the premium rate services that the network provider offers to their members. If access is required to services hosted on another server somewhere across the network, then the WAP gateway will act as a proxy for the client mobile device in establishing the required sessions with the remote host.

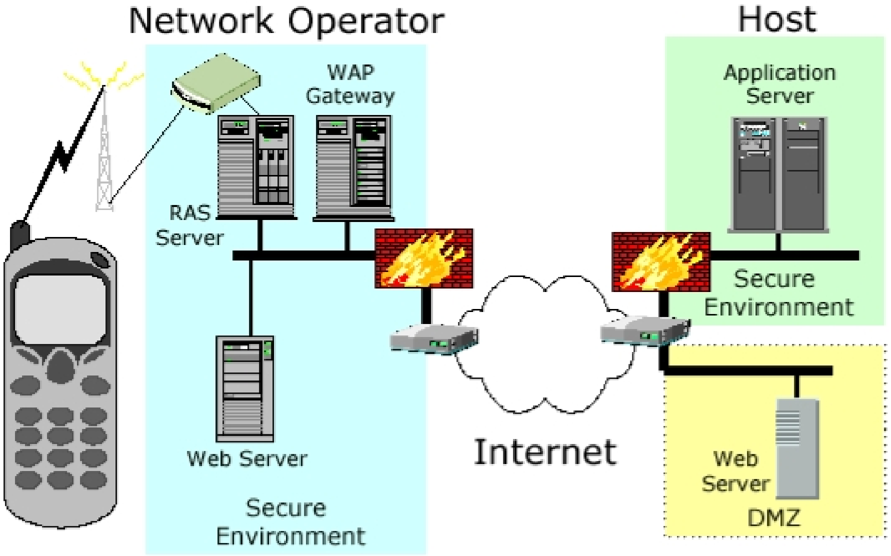

Figure 6.3 WAP Communication Model

From the point of view of security, this scenario has various implications. WTLS is the security protocol that will be used to secure communications to and from the mobile device, but the mobile device's session is necessarily with the WAP gateway rather than the remote host's web server. At the gateway, the secure session terminates and all encrypted material is decrypted. Should there be a requirement for a secure session for communication with the web server, it will be established by the WAP gateway on behalf of the mobile device. The WAP gateway will use TLS to establish such a secure session. While TLS is obviously a robust security protocol, it remains a fact that the secure session is not between the mobile device and the web server. There are actually two secure sessions in play: one between the mobile device and the WAP gateway and the other between the WAP gateway and the web server. This means that there is a security gap, in which the data is not encrypted, at the WAP gateway. This gap, and the span of control of the host server and network operator are illustrated in figure 6.4.

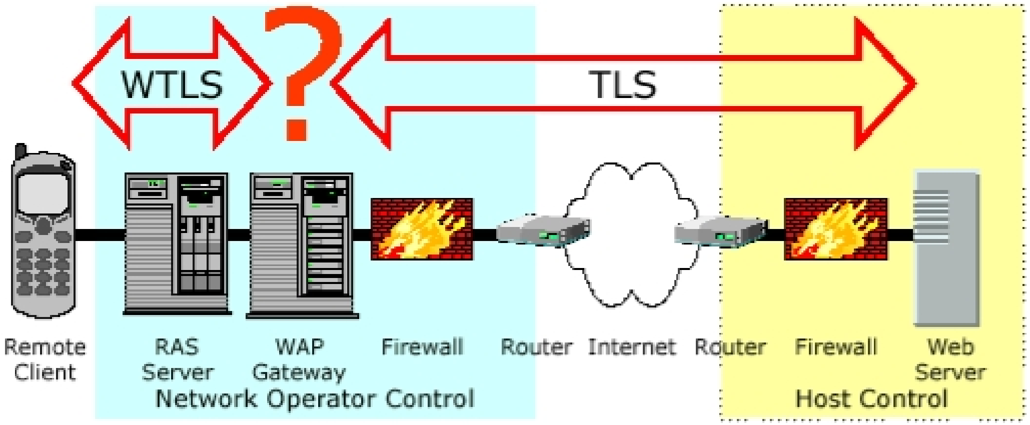

Figure 6.4

The host server's span of control is severely compromised in comparison to the Internet model. In fact, the host has absolutely no control over the security that exists between the mobile device and the WAP gateway. The host also has limited control over the TLS session between the WAP gateway and the web server, and will be limited to providing security that does not exceed a level determined by the network operator. This may or may not be adequate for the host.

6.4 WAP Security Issues:

There are two issues with regard to security in the WAP environment. There are ways of addressing both of these issues, but they both remain issues that need to be addressed.

6.4.1 The Gateway:

We have established that there is a security gap in the WAP model in the form of the WAP gateway. Because of the way that WAP works it is not feasible to do away with the gateway, so we need to establish to what extent it actually is a risk and what the alternative ways of addressing the risk are. It can be argued that the WAP gateway is not actually a security risk because the gateway vendors are aware of the issue and therefore take steps to ensure that the process of decrypting from WTLS and re-encrypting into TLS cannot easily be compromised. Typical of the steps taken will be to ensure that the decryption and re-encryption takes place in memory, that keys and unencrypted data are never saved to disk, and that all memory used as part of the encryption and decryption process is cleared before being handed back to the operating system.

The first problem with all of this is that there are no standards of guarantees about these precautions. You have no way of ascertaining how robust your vendor's implementation actually is, and in the case of a gateway that is hosted by a network operator you may not even be able to tell whose implementation it is. One can also questions of the vendor's promises: in a heavily loaded server, how exactly does the gateway prevent the operating system from swapping memory pages out to swap space? If we accept that there is an exposure at the gateway, no matter how small or how hard the vendors work to protect the unencrypted data, the real question then becomes: who hosts the gateway? Whoever hosts the gateway has the responsibility for protecting it and the data that goes through it, and also has access (potentially, at least) to all of the data that goes through the gateway in unencrypted form. The good news is that it is entirely possible for you to host your own gateway, although before doing so you should consider the implications, in terms of cost and otherwise, of doing so. There are also two different architectures that can be implemented to facilitate hosting your own gateway, and each has different characteristics in terms of security and cost overheads.

6.4.1.1 Model 1:

The first model, which is shown in the figure 6.5, is probably only suitable if you want to provide access to a limited number of people who are not the general public, possibly employees. Here, security is absolutely paramount. In this scenario you would choose to establish an environment similar to any other highly secure dial-up environment. You would establish a bank of dial-up modems connected to one or more RAS servers on your local network. You would be responsible for establishing, maintaining and administering the environment, including details such as dial-up security (possibly through RADIUS or similar). You would then be able to strictly control who has access to the gateway, when this access is possible, and via what telephone numbers. You could implement dial-back to a limited set of numbers, control the IP addresses available, issue and use your own certificates for authentication, and anything else that would contribute to your secure environment. All of the relevant servers would be a secure segment of your local network, and access to and from the Internet may or may not be available. If it is available it will almost certainly be protected by one or more firewalls.

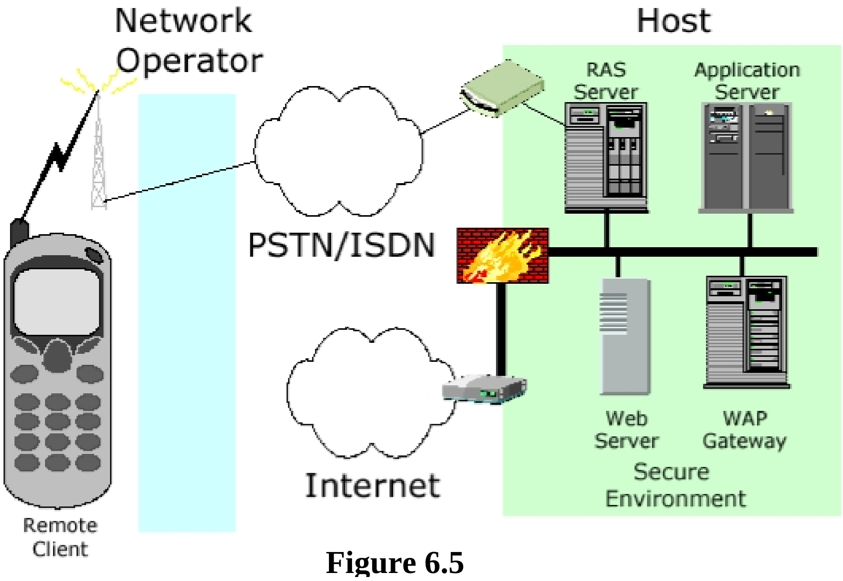

Figure 6.5

6.4.1.2 Model 2:

The second model eliminates the need for the modems and RAS server by making use of the services provided by an ISP. This model is shown in Figure 6.6.

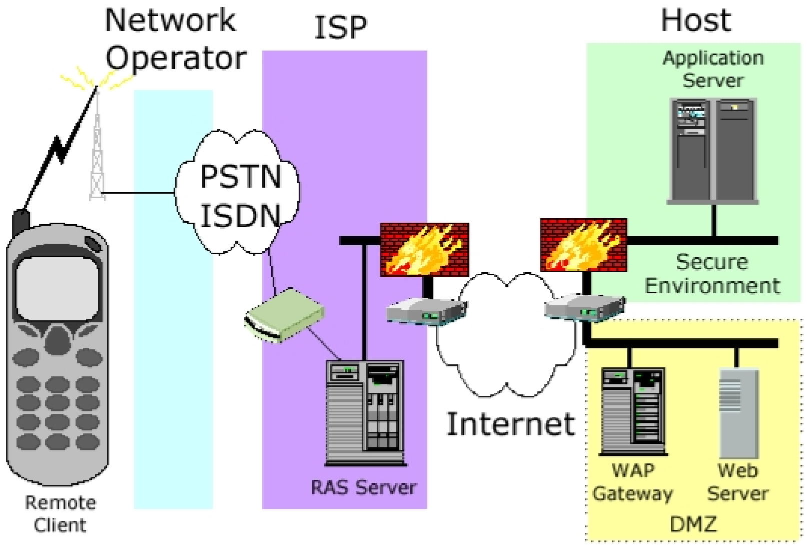

Figure 6.6

This model is in fact very similar to the Internet model, although there are some differences. The remote mobile device will establish a dial-up connection with the ISP's RAS server through a modem hosted by the ISP. The network operator is restricted to connecting the call and has no further influence on the session or the security environment. The RAS server at the ISP acts as a proxy for the mobile device on the ISPs network, and provides all the services that it would to a fixed-wire dial-up client. The ISP network is connected to the Internet via a gateway and is protected by a firewall.

The host's environment would usually be similar to an environment for access by fixed-wire clients over the network. The major difference would be that the host would have a WAP gateway available on the network, typically in the DMZ. Any secure connection from the mobile device would establish a secure session that tunnels through the ISP's RAS server to the WAP gateway. The WAP gateway would then establish a secure TLS session through to the web server, which would make use of services on the application servers hosted on the secure network behind the firewall.

In this scenario we are making use of WTLS in a similar way to a Virtual Private Network (VPN), in that the mobile device establishes a secure tunnel through to the target network. In the case of a VPN, the tunnel is typically to the router on the network, although it doesn't have to be, whereas in this model the 'VPN' tunnels to the WAP gateway. You will need to examine the security requirements of your application to determine whether WTLS provides a secure enough 'VPN' for your application. The other thing to be aware of in this model is that the WAP gateway is typically on the DMZ, which means that it is not as heavily protected as it would be if it were behind the firewall on the secure network segment. This makes it more vulnerable to attack by hackers.

6.4.2 User versus Device:

The second issue that is worth considering with mobile devices, and which is not really a consideration for fixed-wire devices, is the issue of who or what is being authenticated by the certificate. I mentioned previously that a certificate is a reasonably large and complex thing, certainly too complex to type in each time it is required. The result is that the certificate usually ends up being held on your computer, often without you even being aware that it is there, and the system will take card of presenting and validating certificates as and when required. While this is very convenient, it does have some security implications, in that anyone who gains access to your computer can make use of your certificates. The prerequisite is for the person to gain access to your computer. In many cases this is not that easy to achieve, requiring breaking and entering or something similar.

The most immediate way of tackling this problem is to accept that the certificate is going to be stored on the phone, and the phone may be lost. The certificate is still made use of to validate that the mobile device is entitled to access the network, which at least serves to eliminate all of those mobile devices that do not have the required certificate. Once the mobile device is reported missing the certificate can be placed on a certificate revocation list to ensure that it does not provide access in the future. To further validate that the current user of the authenticated device is the rightful user you can make use of a variety of systems, which vary in their complexity and robustness from a simple PIN number through to a SecureID token. While it is easy to dismiss a PIN as being inadequate, pause to remember that almost all of us make use of automated teller machines, and in doing so daily rely on simple PIN numbers to protect our financial assets. Of course when asking users to enter PIN numbers on a mobile device the necessary precautions must be taken, such as masking the numbers with asterisks, and so on.

6.5 Conclusion:

There has been a lot of fuss about security in the WAP world, some of it justified, but most of it being misinformation and misunderstanding. I have often heard it observed that WAP 1.1 does not include security. This is an example of the misinformation that has been around in the industry: WTLS was part of WAP 1.1 and is almost unchanged in WAP 1.2. Security has been there all the time. What is true is that not all vendors have implemented all parts of the specification, and WTLS has often not been implemented at all or has only been implemented at class 1. This will be resolved in time, as vendor's products become more mature and robust, and as the public need for robust security implementation forces vendors to include security in their product offerings.

Even if your WAP gateway does not include WTLS, a WTLS gateway can be obtained from some reliable security solution vendors, like Baltimore Technologies, which will sit on your network between the mobile device and your WAP gateway to provide a WTLS implementation. This type of solution is only feasible if you are hosting your own WAP gateway. WAP can and does provide a robust, secure environment in which an organization can conduct m-commerce or communicate securely. Attention does need to be paid at this stage to the specifics of the implementations, so I would advise a thorough evaluation before committing to a particular vendor's implementation. However, there are robust products our there that you can use to implement a secure environment.

Chapter
7

7. Application Development/Execution

7.1 General Overview

eMusic.com is an m-commerce application developed using WAP. It has been brought together through the combination of software coding and latest telecommunication architecture. It is a B2C enabling application, which uses the latest tools to bring together a technique utilized for marketing..

It is basically a 3 tier application. It involves the use of different technologies and software tools. There are a number of combinations of tools and languages that could have been used. In this case WML and WML Scripts were used at the front end or the 1st tier. Java servlets have been used at the middle tier and Microsoft Access at the 3rd tier. Figure 7.1 shows a typical 3 tier application.

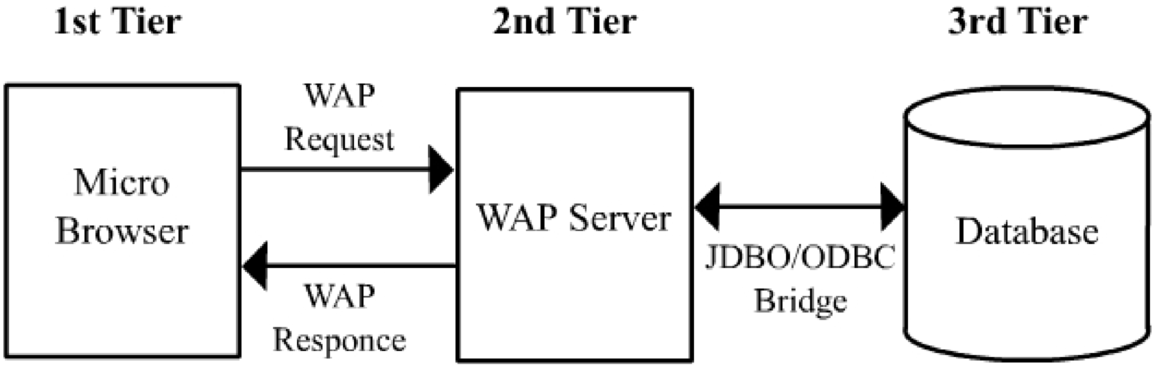

Figure 7.1 A Three Tier Application

7.2 Activity Diagram:

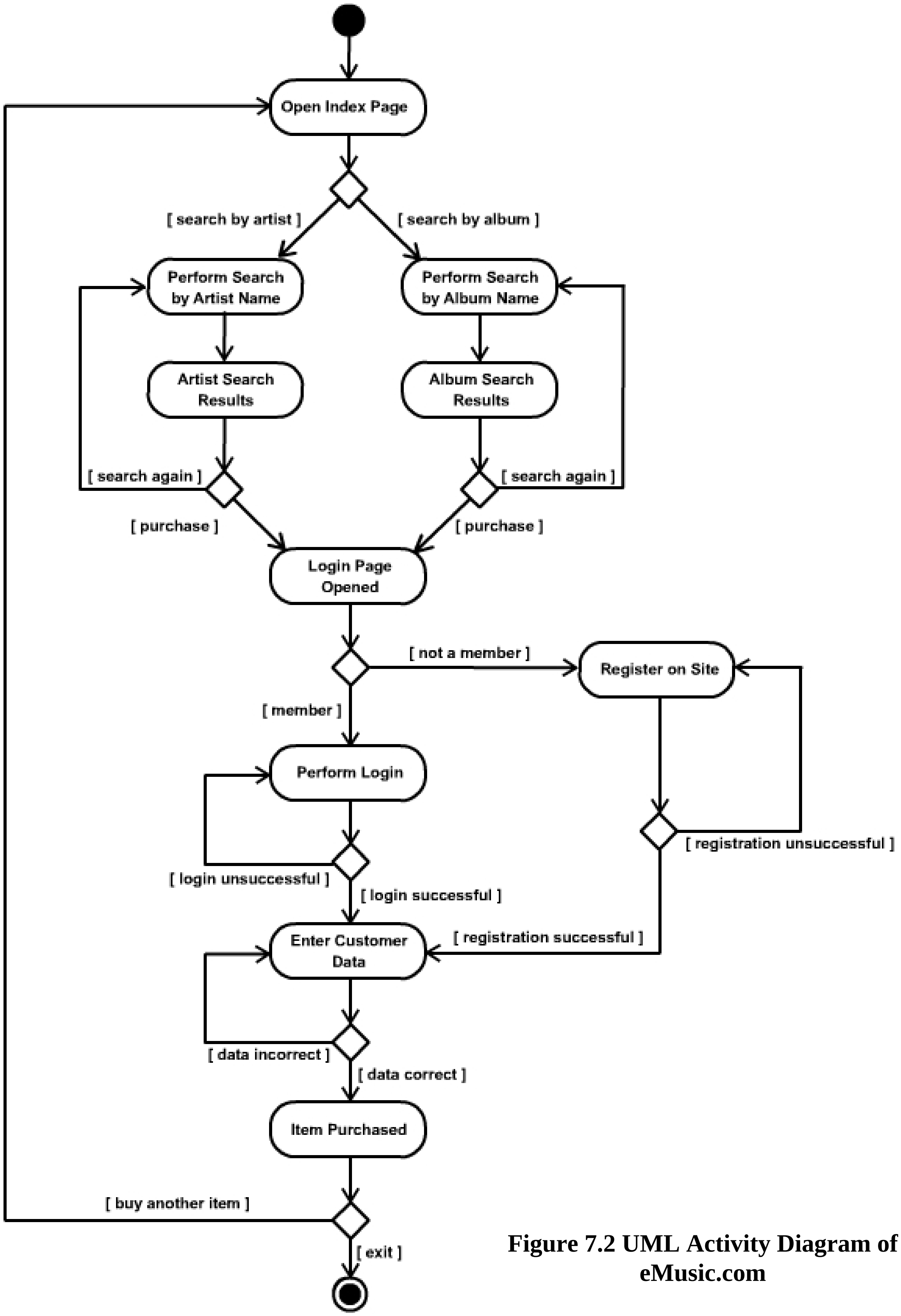

Figure 7.2 UML Activity Diagram of eMusic.com

7.3 Software Design Methodology:

Great emphasis had been laid down at the start of the project for choosing the specific paradigm model but due to the ever changing terrain at the start of the project, the group considered using an evolutionary software process model. The one selected for the project was the **Incremental Model**. The incremental model consists of elements formed from both linear sequential model and the prototyping concepts.

The graphical representation in figure 7.3 of the methodology used by the group is shown below. It consists of 5 stages, where after every design phase completion, the group concurrently began the coding for the existing stage and also the analysis of the next stage. This helped greatly in making sure that the project was completed in time.

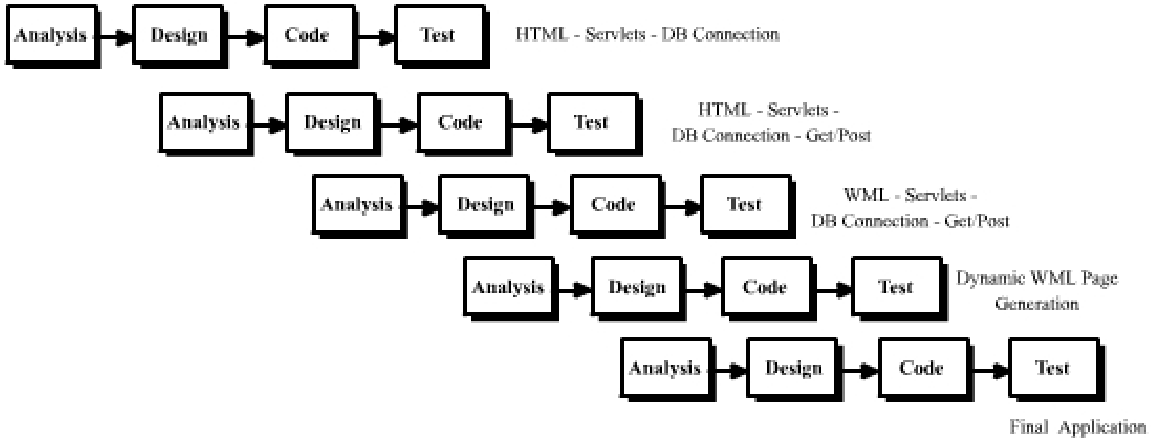

Figure. 7.3 Development Stages

Due to the newness of the field, at the beginning of the project, there are few problems relating to information gathering and ways of going about the project. That is the main reason why the incremental model was chosen. It gave us a chance to develop our project one stage at a time and at the same time to stay abreast with the latest development, which may affect the overall outcome of our project. Initially the approach of staying close to the web architecture was considered. At the same time work was being done to later add a working WAP page to a servlet and to check its working.

7.4 Class Diagram:

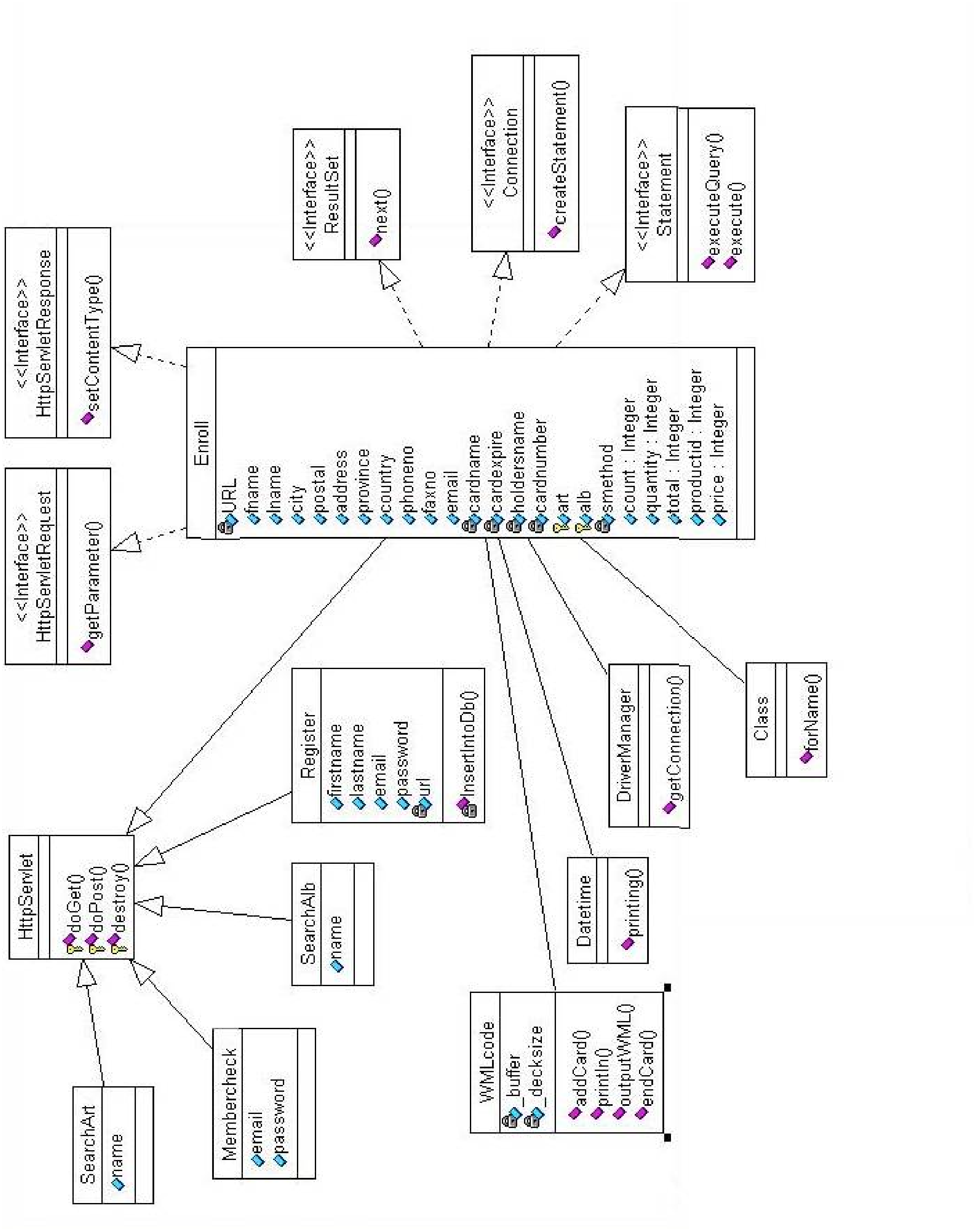

Figure 7.4 Reduced Class Diagram

7.5 Servlets:

The different servlets that are working at the server side carry out all the transferring of data from the client side to the database and vice versa. These servlets are used to communicate between the back end database and the front end client through the WAP Gateway. The functions that the servlets performed are explained below:

7.5.1 SearchArt Servlet:

"SearchArt" deals with running a query on the "products" table of the database searching for the artist name that has been provided by the user. In return the servlet returns the album names of the said artist and its product id from the database.

7.5.2 SearchAlb Servlet :

"SearchAlb" deals with running a query on the "products" table of the database searching for the album name that has been provided by the user. In return the servlet returns the artists/groups that contain albums of the said name, along with the product ID.

7.5.3 Enroll Servlet:

"Enroll" servlet is the servlet which is responsible for taking and arranging the different data at the time of purchasing the Product.

7.5.4 Membercheck Servlet:

"Membercheck" is used to check whether the user is an existing member or a new customer. It is used as a security measure to ensure safe passage of data.

7.5.5 Register Servlet:

"Register" servlet makes an entry for a new user when he/she is ready to purchase a product for the first time. This entry is made in order to make sure that the user's data is kept only to him and secure from any other users.

7.6 Helper Class for Java Servlets:

In order to generate dynamic WML content for the user, a helper class was constructed. This class contained the different tags used by WML, and were referred through member functions. An object of this class was formed and incorporated in the servlets, which carried out the computation on the users data. Hence, with the help of this helper class the developer was able to create the WML content directly from the servlets and tailored to meet the requirement of the user.

7.7 The Database:

No m-commerce site can be complete without a database at its backend. In the case of e-music.com, the database constructed is in MS-Access. The database consists of a total of 6 tables and are shown as below:

- Customer
- Login
- Order
- OrderDetails
- Payments
- Products

The use of these tables is to store the different values entered by the user at different intervals of his visit of the site. Details of the tables would make evident the use of the different tables.

7.7.1 Customers Table:

This table is used to store the values of the user when he/she has agreed to purchase the desired item/product. It includes values such as Customer name, Customer address and other information pertaining to the customer.

7.7.2 Login Table:

At the time when the user has run the search through the database for his/her album or artist of interest, he/she is asked to register to the system. This operation is carried out to ensure security of data and content entered by the user. The record of the user is stored in this table. It contains the users login and password along with his/her name.

7.7.3 Order Table:

Whenever a customer purchases an item/product, he/she gets a “Customer ID” which is automatically allocated to him/her. This ID is made to correspond to a certain “ Order ID” so that a certain purchase by a customer can be recorded. “Order” table also keeps track of the Order date and the shipping method selected by the client.

7.7.4 OrderDetails Table:

OrderDetails is used to contain the attributes of a certain order made by a client. The order details and the appropriate client are linked together using the Order ID present in the both the “Order” table and the “OrderDetails” table. “OrderDetails” contains fields such as the quantity of the product to be purchased and its net cost.

7.7.5 Payments Table:

“Payments Table” is used to store the information pertaining to the payment of the product made by the user. It contains the fields dealing in the type of credit card, the credit card number and other information.

7.7.6 Products Table:

“Products” is the table that contains all the different albums and artists whose work is present for purchase. It contains information such as the artist’s name, album name, the price of the album and its availability.

All these tables are present in a database by the name of WAP. As it is an MS-ACCESS file hence it is stored as WAP.mdb. Before the using of the tables, the database is to be provided a DSN in the ODBC. It is through this DSN that the servlets get to trigger the values present in the tables of the database.

7.8 Converting Web Server to a WAP Server:

For a web server to be able to understand WAP content, the respective MIME types relating to WAP have to be added. MIME (Multi-Purpose Internet Mail Extensions) is an extension of the original Internet e-mail that lets us to use the protocol to exchange different kinds of data files on the Internet: audio, video, images, application programs, and other kinds. In fact servers insert the MIME header at the beginning of any Web transmission. Clients use this header to select an appropriate "player" application for the type of data the header indicates. Following mime types are added. In order to add these MIME types, the developer has to log into the web server as an administrator.

Extensions	MIME Types
.wml	text/vnd.wap.wml
.wmlc	application/vnd.wap.wmlc
.wmls	text/vnd.wap.wmlscript
.wmlsc	applicaion/vnd.wap.wmlscriptc
.wbmp	image/vnd.wap.wbmp

Figure 7.5 MIME Types

7.9 Content Type in Server Side Code:

As MIME types were added to make server communicates with the client, similar to this, Content Types are used by the server to generate the response for particular request. Whenever a servlet creates a response to the server, the servlet needs to be told what sort of the response would be generated. This response generation creates awareness to the web server and is provided to the servlet's response object method, **ContentType**. The argument to this function is usually "**text/html**" but in the case of WAP, as the content generated is not html, hence the Content Type is stated as "**vnd.wap.wml"**

7.10.9 Technologies Used:

eMusic is an m-commerce application developed using WAP. It involves the use of different technologies and software tools. There are a number of combinations of tools and languages that could have been used but the ones for the construction of e-commerce are given as below:

7.10.1 Languages:

The main languages used were Java for server side programming, WML and WML Scripts.

7.10.1.1 JAVA:

The programming trends have moved from one language to another over the decades. The language, which is nowadays considered as being under the limelight, is JAVA. During the design phase of our project, it was considered that such a language should be chosen which has its strengths in networks and also is considered as being "new". After considering java, the choice was to consider either JSP or java's Servlet API for the server side coding. At the end the group considered Servlets not only for its closeness to Java but also because it was also the underlying structure of JSP.

After considering Java’s Servlet API much time was devoted at first learning its working with HTML and then later with WML. Servlets consists of a service method, which indicates whether the action to be performed on the Servlet is a “Post” or a “Get”. After selecting one of the two service methods, the required code is placed in the Servlet’s service method. All the computation is carried out by the Servlet and it is also responsible for generating the dynamic content for the user. Simple Java helper classes were also written which facilitated the servlets in providing the desired WML output.

7.10.1.2 Wireless Markup Language (WML):

WML is a markup language based on **eXtendable Markup Language (XML)** and is intended for use in specifying content and user interface for narrowband devices, including cellular phones and pagers. It has also borrowed elements from HDML 2.0 (Phone.com's proprietary markup language) and HTML. It is case sensitive. WML is based on a **deck** of **cards** metaphor, in which a document is analogous to a deck, and a card is approximately analogous to an individual screen or unit of display. The unit of transmission between the gateway and the mobile device is the deck, and the unit of user interaction is a card within the deck.

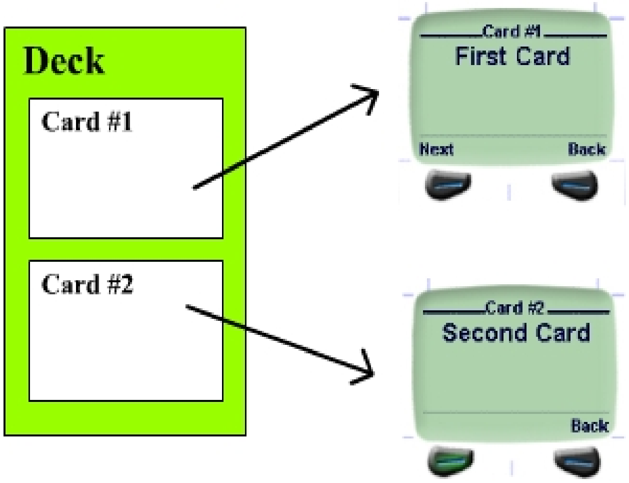

Figure.7.6 Deck and Card Structure

This structure is illustrated in Figure.7.6 Rather than focusing on the details of the rendering of UI elements, or of how the user should interact with the browser, WML focuses on the semantic meaning of the element. Separating the rendering from the meaning allows the actual rendering and implementation on the device to be adapted to the capabilities of the device. WML elements support a number of features including text and images, the ability to interact with the user, navigation capabilities and variables.

Layout and presentation hints can be included with text and images, but it is ultimately up to the browser how it renders the content. Templates can be used to specify a set of characteristics that apply to all cards in the deck.

User interaction is facilitated in the form of entering text or selecting options or actions, which trigger events. Navigation between cards within the deck is supported, as well as between decks. Navigation is either through URL hyperlinks or through a history stack. A history of URLs is maintained in a stack that supports typical stack-like functions (push, pop and reset). A Detail list of WML tags is given in Appendix A. The WML code for the WML deck in Figure 7.6 is given in Figure 7.7

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
        <template>
                <do type="prev"><prev/></do>
        </template>

<card id="card1" title="Card #1">
        <do type="accept" label="Next">
                <go href="#card2"/>
        </do>
        <p align="center">
                <big><b>First Card</b></big>
        </p>
</card>

<card id="card2" title="Card #2">
        <p align="center">
                <big><b>Second Card</b></big>
        </p>
</card>
```

Figure 7.7 WML code for Deck in Figure 7.6

7.10.1.3 Wml Scripts:

Wireless Markup Language Script (WMLScript) is part of WAP application layer and it can be used to add client side procedural logic to WML cards and decks. WMLScript is to WML what JavaScript is to HTML; it is a scripting language, which (like JavaScript) is based on ECMAScript, although it is not fully compliant with ECMAScript. It includes all of the usual procedural constructs that you would expect in a scripting language, such as loops, conditions, and so on. For the purposes of efficiency, it is compiled on the server into byte code, which is then executed on the mobile device in the VM. Functions are also built into WMLScript, and additional functions are available in the standard WMLScript libraries, which include functions

7.10.2 Tools Used:

The tool used for the project are:

7.10.2.1 Nokia WAP Development Toolkit:

This is an IDE developed by Nokia. It is used for writing and compiling WML and WML Scripts. Its also has a few emulators for Nokia WAP Phones.

7.10.2.2 WAP Gateway:

Two gateways were used in the project one was Zaramond WAP Gateway and the other was Nokia Active Server 2.1. The Zaramond WAP Gateway is a very light and compact gateway provided by Zaramond Limited. It runs on Windows operating system. A free evaluation version is available for downloading on the company’s site. The Nokia Active server was used in the real time testing.

7.10.2.3 Java Web Server:

Java web server was modified and used as a WAP Content Server. The main reason for using this server was its compatibility with Java Servlets.

7.10.2.4 Various emulators by Nokia, Phone.com and M3Gate:

A number of WAP phones emulators were used for the reason that different WAP browsers display content different. In order to make content viewable on all WAP phones it is a good practice to check it on various emulators.

Chapter

8

8. Problems During Implementation:

There were a number of problems faced in the initial research phase of the project because WAP was a new technology at the time the project was undertaken. There was no technical support in the sense that no software house was working on WAP nor were they willing to work on it. Also there wasn’t a lot of material available on WAP.

The main problem was the learning of three new languages i.e. Java, WML and WMLScript. Initially the problem felt in the group was the understanding of WML and WMLScripts. Later when the situation was molded from Web to WAP, serious problems were felt at the programming phase. The configuring of the Web server to work for WAP content was another matter, which took understanding and experience of the Java Web server being used.

There were a lot of issues that needed to be resolved regarding the integration of Java servlets and WML. As WML is a xml based language it is a very syntax intensive. Unlike HTML in which syntax errors are ignored and rest of the page is displayed by the browser, the micro browser doesn’t display any thing if there is even a single error in the wml code. So the coding of wml was very tedious work. There were specially a number of problems when dynamic wml pages were generated using the servlets. Also the generated pages had to be less then 1024 bytes in size.

Chapter

9

9. Future Advancements in Project

At the start of the project the aim was to carry out detailed study of WAP and its components. It involved the development of a prototype M-commerce application that would run on WAP. During the research and development phase of the project, the group had carried out extensive work on how an m-commerce application would be made to work. In the course of this effort, it started to gain knowledge regarding the architectural frame work of WAP and how its networks are established and setup. Such studies provided the incentive to the group to go ahead and setup the entire network required for real time execution of WAP. Hence work began with its humble beginnings and ended up in creating the entire WAP scenario which, under the current situation, is complete. The current implementation of WAP is found only to be present at one of the two digital bearer services in the country and that too is in the testing phase.

The implementation of WAP from a network point of view is complete; however work can be done on one of the following:

- Wireless Telephony Applications (WTA)
- PUSH Architecture
- End to End Security on WAP
- New Wireless technologies like MeXE, EPOC and Symbian
- Controlling hardware devices using WAP. The group had also done a considerable amount of work on this aspect of WAP. A prototype application was developed that would provide security services in a home or office using WAP phone. This involved interfacing of electronic devices with WAP. The idea behind the application was that the user would log on to a server and control different electronic devices in his home or office. In order to create such an application, the choice of programming domain was preferred to be Microsoft

Technologies. The reason for such a consideration was that in order to provide interface to the hardware devices, programming had to be done in a domain where the code can provide extensive utilization of the parallel ports and can also provide interaction with the mobile agent.

Chapter

10

10. Future WAP Technologies

In this chapter we will look at some technologies that are not yet easy to implement fully right now, but which will become very common in the future.

10.1 WTA - Wireless Telephony Applications:

Wireless Telephony Applications are those applications that have mechanism defined to interact with the telephony related function present in the mobile phone and those provided by the phone network. Some of the telephony related functions on a commonly available mobile phone are Making a mobile originated call, Receiving a call, Sending and receiving Short Text Messages (e.g. SMS), Adding, searching and removing phonebook etc. There can be a number of functions apart from the ones mentioned above. Some of the network dependent telephony application can be *Call Hold*, *Call Transfer*, *Call Forwarding*, *Conferencing*, and *Voicemail.*

The latest specification of WAP defines WTA in a more precise manner than any of the previous versions. It will represent the first practical implementation of different technologies bringing together both the interactive information access and the simplicity of a no-hassle electronic device, making such a device easy to use for people of any skill level.

Although presently WTA has not been implemented properly but here a wish list of the features of WTA is presented:

- Basic features like making and receiving voice calls should be possible in an interactive, yet device and network independent manner.
- If the network operator so chooses, it should be possible to provide mechanism to operate a telephony features like call forwarding, call redirection and possibly a

whole lot of others, with the same user interface for all users regardless of the vendor of WAP phone being used.

- From simple browsing applications, it should be possible to initiate phone calls just by selecting a hyperlink in a WML deck that is being displayed on the phone.
- A URI based naming model should be available to access not only basic telephony features but also network services like Voicemail.

10.2 PUSH Message Framework:

This term is used to describe all the protocols, service interfaces and software entities, which together provide the capability of "pushing" data to the user agent in a WAP client device in a asynchronous manner. This means that the user does not specifically requests for the data. Rather it is sent to him/her. Figure 9.1 shows the Push Message Framework architecture. This framework provides the content providers the ability to make available to their mobile customers new services like automatic notification of new email, update stock quotes, advertisements and number of other useful services. In Figure 9.1 the numbered arrows depict the sequence in which the discrete events, which comprise a Push Message transmission, occurs. Inside the arrows, the content type of the

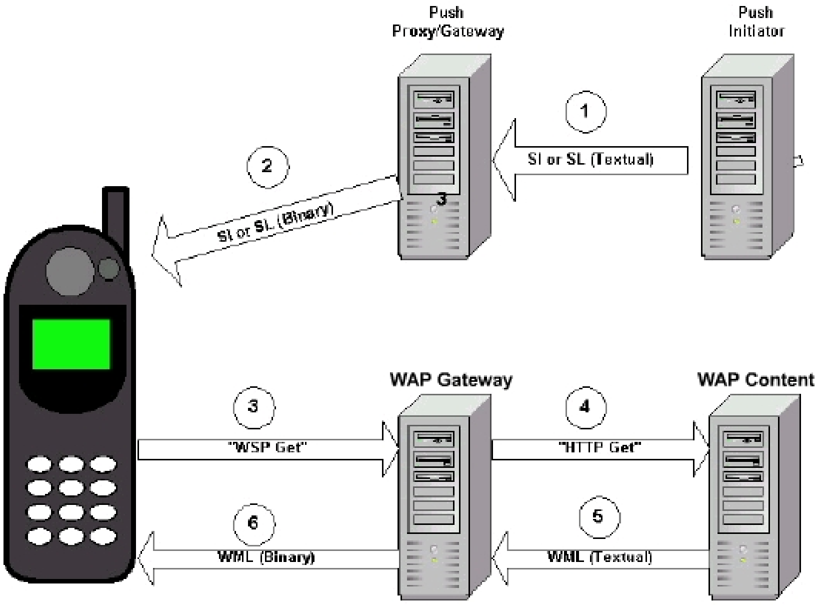

Figure. 9.1 Push Message Framework

transmission is shown. Typically, WML content is not directly pushed to the user agent because its reception and display could disrupt the user's current browsing activities. Instead the push initiator program sends a Push Message, which contains a priority (or intrusiveness) level, a URL to fetch and an alert message to display to the user. This alert message is displayed to the user at some point depending on the priority level. The text of the message presents the user the opportunity to fetch the URL, which contains the WML content announced by the alert message. A Push Message can contain one of the two following content types:

10.2.1 Service Indication (SI):

This content type provides the means to send an alert message to the user asynchronously. When this descriptive message is displayed to the user, the user can choose to load the WML content specified by the URL. The SI content type is an application of XML 1.0.

10.2.2 Service Loading (SL):

This content type provides the means to transmit to the mobile device a URL; the client device then "pulls" from the origin server without the knowledge of the mobile client user. This content type is also an application of XML 1.0. As the user might be busy in other activities, the service loading mechanism also provides the push application developer to set the level of intrusiveness. SL may also direct the pulled content to be cached only and not displayed or executed, as performance enhancements.

10.3 Mobile Service Initiative (M – Service):

The GSM Association is concerned about fragmentation in the handset market and the inability to depend on specific handset features which it believes are necessary for the success of its members. To address this fragmentation, the GSMA has published feature guidelines for mobile phones that aim to help ensure that operators could depend on a certain set of consistent features and services in mobile handsets. This will allow the operators and content providers to focus on building 'globally available compelling applications that will excite users and create opportunities for new revenue streams based on a common application framework.

At a high level, the requirements in the M-Services Guidelines cover.

- Definition of a Graphical User Interface for WAP browsers.
- A common framework for downloading consumer-oriented content.
- Multimedia messaging.

Additionally, the Guidelines suggest usage models for a number of common applications in phones.

The GSMA approved the M-Services guidelines in late May 2001. The guidelines lists what it believes are conformance recommendations for phones and have made its recommendations freely available to all manufacturers of handsets and software developers. It is now up to the handset manufacturers to implement M-Services into their handsets in a quality way. The M-Services Guidelines will potentially positively benefit many operators, content developers, and handset vendors in addition to consumers. Many operators and content developers are already designing services based on these common handset features to be available this year.

The existence of the M-Services Guidelines will allow operators to focus on building compelling services that will work across a broad range of devices. The M-Services Guidelines will also serve as a bridge to WAP 2.0 services based on high-bandwidth 2.5G and 3G networks as the infrastructure and devices are made available over the next few years. The GSM Association has received overwhelming support from its operator members for the M-Services initiative. Openwave has committed to support these requirements as well, both through licensing of the Openwave Mobile Browser to manufacturers and licensing of the Openwave Download Fun Server and other infrastructure products to operators. Additionally, Openwave has agreed to license essential intellectual property specified in M-Services to third party vendors who desire to add support for those protocols to their browsers and servers. Additionally, and most importantly, most leading GSM handset manufacturers will provide handsets based on the M-Services guidelines. Companies that have announced support include Alcatel, Ericsson, Motorola, Nokia, Siemens, Samsung, and Sagem, among others. The M-Services Guidelines is not a standard. It is however founded on existing standards and protocols that exist today. The set of M-Services guidelines together make up a common baseline of features that application developers and content providers can depend on in designing and developing their services. How will this impact 2.5G and 3G. Does this mean that the carriers will be able to recoup their investment in 3G licenses? Certainly one of the goals is for operators to generate revenues based on their existing 2G and 2.5G networks. The philosophy is to build compelling services on existing networks and bandwidth now - and upgrade those services to higher bandwidth when their network is there and there is compelling new content that requires that high bandwidth. The successful rollout of M-

Services may reduce the false idea that 3G is required for successful revenue-generating data services, allowing 3G networks to be rolled out at a more rational pace as demand for high bandwidth grows.

10.4 Future of Security:

10.4.1 WTLS

WTLS, being based on an established and stable standard, is unlikely to change significantly or fundamentally for the foreseeable future. I expect that most of the changes in the next few releases will be oriented towards clarifying some issues in the specification and general 'housekeeping'. The 1.2 specifications did this, and added some advice about guarding against certain types of attacks. All of this information is only of relevance to people who are developing their own WTLS implementation, and also much of the information dates very quickly, so I would expect it to be refreshed in just about every release. I do not, however, anticipate any major changes unless a major security exposure in one of the ciphers is identified.

10.4.2 End-to-End Security

The WAP Forum has made it clear that they are aware of the issues around the security gap at the WAP gateway. They have also make it clear that they intend to plug the gap by providing an end-to-end security standard in a subsequent release. There have been hints that they would attempt to address this through changes to WTLS, but I think that this is marketing rather than technology speaking. The issue does not arise because of any weakness in WTLS and is caused solely by the position that the gateway fulfils in the WAP communications chain. In order to address the issue, either the gateway has to be eliminated or some other solution has to be implemented, probably at a higher level in the protocol stack. The WAP Forum has also indicated that the WMLScript Crypto library may be extended in the future to include cryptographic functions. At this point in time there is only a function that supports signing data. To my mind, it seems logical that the way to implement end-to-end security is by means of encryption functionality at the application level. A necessary prerequisite for this, however, will be the capability of mobile devices to deal with the processing loads associated with encryption functions. Part of the solution to this problem may actually lie in the WIM.

10.4.3 WIM

The Wireless Identity Module specification is new in the WAP 1.2 specification. It provides a means to offload the storage of keys and of cryptographic functions onto what is described as a tamper proof device. This is basically a smart card, although it could also be a SIM. The specification covers only the low level capabilities of a WIM in the current specification, and doesn't present an API for making use of a WIM when present, although I expect that an API and framework will be provided in a future release. The introduction of the WIM could help to address the issues around authenticating the device as opposed to the user.

Chapter

11

11. Conclusion

The main objective of the project was to have a detailed look at WAP and its constituent components, what issues it tries to address and how it does this, and what facilities are available for the developer to build applications that are appropriate for deployment on mobile devices and across wireless networks, and also to develop a real time application for WAP.

This project gave us the chance to learn a lot of different technologies, languages and tools. It involved both networking and programming side, and also included a lot of detailed research. WAP being a very new technology gave us an insight into relatively new technologies like Java, WML and WMLScripts. In due course of the project a proper design methodology was followed. This gave us the chance to practice our software engineering skills. One of the requirements of the project was knowledge of wireless bearer services so a little research was also done on this topic too.

As for WAP the technology itself is right in its infancy. It is under a lot of criticism but like everything else it will take time to mature. And once it has this is turn out to be very useful technology in the future, one that will some day become a necessity for every one.

Chapter 12

12. References

- Professional WAP by Charles Arehart, Nirmal Chidambaram, Shashikiran Guruparsad, Alex Homer, Ric Howell, Stephan Kasippillai, Rob Machin, Tom Myers, Alexander Nakhimovsky, Luca Passani, Chris Pedley, Richard Taylor, Marco Toschi. WROX Press Ltd. August 2000

- WAP Architecture Specifications, WAP-238-WML-20010626-p, www.wapforum.org

- WMLScript Specification, WAP-193-WMLS-20001025-a, www.wapforum.org

- Wireless Markup Language Specification, WAP-238-WML-20010626-p, www.wapforum.org

- Wireless Markup Language Reference, Version 1.1, www.forum.nokia.com

- Wireless Markup Language Script Reference, Version 1.1, www.forum.nokia.com

- Nokia WAP Toolkit Version 2.0 Developer’s Guide, Version 2.0, www.forum.nokia.com

- M-Service Initiative Guidelines, Version 3.0.0, www.gsmworld.com

- www.vbxml.com

- www.mformobile.com

- www.anywhereyougo.com

Chapter

13

13. Appendixes

Appendix A	WML Reference
Appendix B	Glossary
Appendix C	Complete Class Diagram

Appendix A WML Tag Reference

<a>

Specifies that the text within the tags a hyperlink. The destination of a link is specified as a URI: the address or ID of another tag. Authors are encouraged to use the a tag instead of anchor where possible. It is invalid to nest anchor tags. The a tag is a short form of the anchor tag; it is essentially bound to a go task without variables.

<access>

Specifies access control information for the entire deck. It is an error for a deck to contain more than one access element. If a deck does not include an access element, access control is disabled. When access control is disabled, cards in any deck can access this deck. deck's domain and path attributes specify which other decks may access it. As the browser navigates from one deck to another, it performs access control checks to determine whether the destination deck allows access from the current deck. If a deck has a domain and/or path attribute, the referring deck's URI must match the values of the attributes. Matching is done as follows: the access domain is suffix-matched against the domain name portion of the referring URI and the access path is prefix matched against the path portion of the referring URI. Domain suffix matching is done using the entire element of each sub-domain and must match each element exactly (e.g. www.wapforum.org shall match wapforum.org, but shall not match forum.org). Path prefix matching is done using entire path elements and must match each element exactly

<anchor>

Specifies that the text within the tags a hyperlink. The destination of a link is specified as a URI: the address or ID of another tag. Authors are encouraged to use the a tag instead of anchor where possible. It is invalid to nest anchor tags.

<b>

Indicates that the text within the tags should be rendered with bold formatting. Authors should attempt to use the strong and em tags in place of the b, i and u tags, except where explicit control over text presentation is required.

<big>

Indicates that the text within the tags should be rendered with a large font.

<br>

Ends the current line and starts a new line. Should also be supported within tables.

<card>

A WML deck contains a collection of cards. The card element is a container of text and input elements that is sufficiently flexible to allow presentation and layout in a wide variety of devices, with a wide variety of display and input characteristics. A card can contain markup, input fields and elements indicating the structure of the card. A card’s id may be used as a fragment anchor.

<do>

The do tag provides a general mechanism for the user to act upon the current card. The representation of the do tag is dependent on the device and the author must only assume that the tag is mapped to a unique user interface widget, such as a button, that the user can activate. The do tag may appear at both the card and deck-level: Card-level: the do tag may appear inside a card tag and may be located anywhere in the text flow. Deck-level: the do tag may appear inside a deck template, indicating a deck-level do tag. A deck-level do tag applies to all cards in the deck, and is equivalent to having specified the do within each card. For the purposes of rendering, the browser must behave as if deck-level do tags are located at the end of the card’s text flow. A card-level do tag overrides (or “shadows”) a deck-level do tag if they have the same name. For a single card, the active do tags are defined as the do tags specified in the card, plus any do tags specified in the deck’s template and not overridden in the card. Non-active do tags and active do tags with a noop task are not displayed. All active do tags with a task other than noop will be shown in some manner. When the user activates the do tag, the associated task is executed.

<em>

Indicates that the text within the tags should be rendered with some form of emphasis. Authors should attempt to use the em and strong tags in place of the b, i and u tags, except where explicit control over text presentation is required.

<fieldset>

The fieldset element allows the grouping of related fields and text. This grouping allows the optimising of layout and navigation. Fieldset elements may nest, providing the author with a means of specifying behaviour across a wide variety of devices

<go>

Declares a ‘go’ task, indicating navigation to a new URI. If the URI names a WML card or deck, the execution of the task will cause that item to be displayed. This task executes a ‘push’ operation on the browser’s history stack.

<head>

The head tag contains information relating to the deck as a whole, including meta-data and access control tags.

<i>

Indicates that the text within the tags should be rendered with italic formatting. Authors should attempt to use the strong and em tags in place of the b, i and u tags, except where explicit control over text presentation is required.

<img> The img tag indicates that an image is to be included in the text flow. Image layout is done within the context of normal text layout.

<input> The input element specifies a text entry object. The user input is constrained by the optional format attribute.

<meta> The meta element contains generic meta-information relating to the WML deck. Meta-information is specified with property names and values. This specification does not define any properties, nor does it define how browsers must interpret meta-data.

<noop> The noop tag specifies that nothing should be done - that is, 'no operation'. This can be used in a card to shadow a task that has been specified in a template at the deck level.

<onevent> The onevent element binds a task to a particular intrinsic event for the immediately enclosing element. For example, specifying an onevent element inside a card element associates an intrinsic event binding with that card element. The browser will ignore any onevent element specifying a type that does not correspond to a legal intrinsic event for the immediately enclosing element.

<optgroup> The optgroup element allows the author to group related option elements into a hierarchy. The browser may use this hierarchy to facilitate layout and presentation on a wide variety of devices.

<option> This element specifies a single choice option in a select element.

<p> The p element establishes both the line wrap and alignment parameters for a paragraph. If the text alignment is not specified, it defaults to left. If the line-wrap mode is not specified, it is identical to the line-wrap mode of the previous paragraph in the current card. Empty para-graphs (ie, an empty element or an element with only insignificant white space) will be considered as insignificant and ignored by browsers. If the first p element in a card does not specify a line-wrap or alignment mode, that mode defaults to the initial mode for the card. The browser will insert a line break into the text flow between significant paragraphs.

<postfield> The postfield element specifies a field name and value for transmission to an origin server during a URL request. The actual encoding of the name and value will depend on the method used to communicate with the origin server.

<prev> The prev tag declares a 'prev' task, indicating navigation to the previous URL on the history stack.

<refresh>

The refresh tag declares a refresh task, indicating an update of the screen and device context as specified by the setvar tags, for example. User-visible side effects of the state changes (e.g. a change in the screen display) occur during the processing of the refresh task.

<select>

The select element lets users pick from a list of options. Each option is specified by an option element. Each option element may have one line of formatted text. Option elements may be organised into hierarchical groups using the optgroup element.

<setvar>

The setvar element specifies the variable to set in the current browser context as a side effect of executing a task. The element must be ignored if the name attribute doe not evaluate to a legal variable name at runtime.

<small>

Indicates that the text within the tags should be rendered with a small font.

<strong>

Indicates that the text within the tags should be rendered with some form of strong emphasis. Authors should attempt to use the strong and em tags in place of the b,and u tags, except where explicit control over text presentation is required.

<table>

Used together with the tr and td tags to create sets of aligned columns of text and images in a card. The table tags determine the structure of the columns. The tags separate content into columns, but do not specify column or intercolumn widths. The number of columns for the row set must be specified by the columns attribute. If the actual number of columns in a row is less than the value specified by the columns attribute, the row will be effectively padded with empty columns. The orientation of the table depends on the language. For left-to-right languages, the leftmost column is the first column in the table. Columns are added to the right side of a row to pad left-to-right tables. Columns are added to the left side of a row to pad right-to-left table. If the actual number of columns in a row is greater than the value specified by the columns attribute, the extra columns of the row will be aggregated into the last column, such that the row contains exactly the number of columns specified. A single inter-word space will be inserted between two cells that are being aggregated. The table will be rendered as narrow as possible given the contents. A non-zero width gutter is used to separate each non-empty column.

<td>

The td element is used as a container to hold a single table cell data within a table row. Table data cells may be empty. Empty cells are significant, and must not be ignored. The browser will attempt to deal with multiple line data cells that may result from using images or line breaks.

<template>

The template element declares a template for cards in the deck. Event bindings specified in the template element (e.g. do or onevent) apply to all cards in that deck, although a card element may override the behaviour specified in the template element. In prticular: DO elements specified in the template element may be overridden in individual cards if both elements have the same NAME attribute value. * Intrinsic event bindings specified in the template element may be overridden by the specifica-tion of an event binding in a card element.

<timer>

The timer element declares a card timer which exposes a means of processing inactivity or idle time. The timer is initialised and started at card entry and is stopped when the card is exited. Card entry is any task or user action that results in the card being activated, for example, navigating into the card. Card exit is defined as the execution of any task. The value of a timer will decrement from the initial value, triggering the delivery of an ontimer intrinsic event when it reaches zero. Note that timer resolution and the interaction of the timer with the browser's user interface and other time-based device functionality is implementation dependent. It is an error to have more than one timer element in a card. The timer timeout value is specified in units of one-tenth (1/10) of a second. The author should not expect a particular timer resolution and should provide the user with another means to invoke a timer's task. A timeout value of zero disables the timer. Invoking a refresh task is considered an exit. The task stops the timer, commits its value to the context, and updates the browser accordingly. Completion of the refresh task is considered an entry to the card. At that time, the timer must resume.

<tr>

The tr element is used as a container to hold a single table row. Table rows may be empty (i.e., all cells are empty). Empty table rows are significant and must not be ignored.

<u>

Indicates that the text within the tags should be rendered with underline formatting. Authors should attempt to use the strong and em tags in place of the b, i and u tags, except where explicit control over text presentation is required.

<wml>

The wml tag defines a deck and encloses all information and cards in the deck.

Appendix B Glossary

1G First generation wireless: analog cellphones.

2.5G 2G plus faster data services.

2G Second generation wireless: digital cellphones.

3G Third generation wireless: digital plus high speed data and global roaming. Known as IMT 2000 by the ITU and implemented in Europe as UMTS and cdma2000 in North America. Goals are high quality multimedia and advanced global roaming (inhouse, cellular, sat

ASP Active Server Pages. Serverside scripting technology to make interactive web pages. Based on VBScript.

Base station The central radio transmitter/receiver that maintains communications with a mobile telephone. Most countries require several hundred base stations, in order to give approximated full coverage for mobile subscribers.

Bluetooth Wireless personal area network (PAN) standard geared for home and office. Uses 2.4GHz band at 720 Kbps within 30 foot range.

Bytecode Content encoding where the content is typically a set of low level opcodes and operands for a targeted hardware (or virtual) machine.

Card Basic unit of WML navigation or user interface.

CDMA Code Division Multiple Access QUALCOMM's spread spectrum air interface method. It codes each conversation expanding it 128 times, which makes it easy to decipher at receiving end.

cdma2000 3G CDMA evolution from cdmaONE supported by cdmaONE operators. Now known as the 1x Multi Carrier mode (1x MC) in an overall standard for 3G CDMA.

CDPD Cellular Digital Packet Data. A digital wireless transmission system that is deployed as an enhancement to the existing analog cellular network. Based on IBM's CelluPlan II, it provides a packet overlay onto the AMPS network and moves data at 19.2 Kbps over ever-changing unused intervals in the voice channels. If all the channels are used, the data is stored and forwarded when a channel becomes available.

CDPD was developed as a wireless extension to an IP network and uses the four octet (0.0.0.0) address for connections. CDPD networks cover most of the major urban areas in the U.S. and has been deployed by AT&T, Ameritech, GTE, BellAtlantic Mobile and other carriers. By the late 1990s, incompatibility issues had been worked out, and

	roaming agreements and interoperability between carriers is generally nationwide. CDPD modems are available on PC Cards for laptop and handheld computers.
Client	A device or application that initiates a request for connection with a server.
Content	Data that is stored or generated at the origin server. It is usually displayed or interpreted by a user agent in response to a user request.
Content Encoding	Act of converting content from one format to another. It is can also specify a particular format or encoding standard or process.
Content Format	Actual representation of content.
Deck	A collection of WML cards. A WML deck is also an XML document.
DTD	Document Type Definition. A DTD defines the names and contents of all elements that are permissible in a certain document. A DTD is used to specify XML document structure.
GPS	Global Position System. Based on US defence satellite system, enables tracking of individuals. This technology may prove helpful when navigating a car in the city, or help emergency rescue-team to locate the person in need of help.
Device	Network entity capable of sending and receiving packets of information and has a unique device address.
Gateway	A hardware and software combination that runs on the OSI application layer and allows dissimilar protocols to communicate by filtering communications through industry-standard protocols. Examples of protocols a gateway might use are TCP/IP, X.25, and SNA.
GPRS	General Packet Radio Service allows packet rather than circuit switch connections on cellular networks. This allows high speed mobile access and the ability to only connect to the mobile network when internet access is required.
GSM	Global System for Mobile Communications. Digital cellphone system used throughout Europe based on TDMA. Introduced SIM card and short messaging (SMS). GSM has a maximum data transfer rate of 9.6 Kbps
HDML	Handheld Device Markup Language. Forerunner of WML.
HTML	HyperText Markup Language [HTML4]
HTTP	HyperText Transfer Protocol [RFC2068]
i-Mode	Packet based information service for mobile phones from NTT

	DoCoMo (Japan). First to provide Web browsing from cellphones.
IP	Internet Protocol. The IP part of the TCP/IP communications protocol. IP implements the network layer (layer 3) of the protocol, which contains a network address and is used to route a message to a different network or subnetwork. IP accepts "packets" from the layer 4 transport protocol (TCP or UDP), adds its own header to it and delivers a "datagram" to the layer 2 data link protocol. It may also break the packet into fragments to support the maximum transmission unit (MTU) of the network.
IP address	Internet Protocol address. The address of a computer attached to a TCP/IP network. Every client and server station must have a unique IP address. Client workstations have either a permanent address or one that is dynamically assigned to them each dial-up session. IP addresses are written as four sets of numbers separated by periods; for example, 204.171.64.2. The TCP/IP packet uses 32 bits to contain the IP address, which is made up of a network and host address (netid and hostid).
Java	An object-oriented programming language developed by Sun Microsystems that creates applications that work on multiple operating systems.
JavaScript	Programming Language used to add dynamic behavior to HTML documents.
Microbrowser	A Web browser specialized for a smart phone or PDA. It is optimized to run in the low-memory and small-screen environment of a handheld device.
NTT DoCoMo	Wireless division of Nippon Telegraph and Telephone, Japan. A Japanese cellular provider and chief developer of i-Mode.
Origin Server	Server on which a given resource resides or is to be created.
Packet	A piece of data transmitted over a packet-switching network such as the Internet. A packet includes not just data but also its destination.
PDA	Personal Digital Assistant. Handheld computer or personal organiser device.
PIN	Personal Identification Number
Resource	A network data object or service that can be identified by a URL.
Server	Device or application that passively waits for connection requests from one or more clients.
SGML	Standardized Generalized Markup Language. This is a general purpose

	language for domain specific markup languages.
Short Messaging	Sending small text messages to cellphones. GSM pioneered Short Message Service (SMS), now used in all digital cellphones.
SIM	Subscriber Identity Module
SIM card	Smart card that gives GSM phone its user identity. Lets phones be easily rented or borrowed.
Smart Phone	A digital cellular phone that has text messaging, Web access and other data services along with voice.
SMS	Short Messaging Service. GSM coined the phrase, but similar text messaging used in most digital cellphone systems.
TCP/IP	Transmission Control Protocol/Internet Protocol. A communications protocol developed under contract from the U.S. Department of Defense to internetwork dissimilar systems. Invented by Vinton Cerf and Bob Kahn, this de facto UNIX standard is the protocol of the Internet and has become the global standard for communications. TCP provides transport functions, which ensures that the total amount of bytes sent is received correctly at the other end. UDP is an alternate transport that does not guarantee delivery. It is widely used for realtime voice and video transmissions where erroneous packets are not retransmitted. IP provides the routing mechanism. TCP/IP is a routable protocol, which means that all messages contain not only the address of the destination station, but the address of a destination network. This allows TCP/IP messages to be sent to multiple networks within an organization or around the world, hence its use in the worldwide Internet. Every client and server in a TCP/IP network requires an IP address, which is either permanently assigned or dynamically assigned at startup.
TDMA	Time Division Multiple Access. Air interface for digital cellphones that interleaves data in time slots and provides multiple access to a small number of wireless frequencies. It is a component of GSM.
UI	User Interface
UMTS	Universal Mobile Telecommunications System. Is a new generation technology for rapidly moving data and multimedia over wireless devices. The European implementation of the 3G wireless phone system. UMTS provides service in the 2GHz band and offers global roaming and personalized features. Designed as an evolutionary system for

	GSM network operators, multimedia data rates up to 2 Mbps are expected.
URI	Uniform Resource Identifier. The addressing technology from which URLs are created. Technically, URLs such as HTTP:// and FTP:// are specific subsets of URIs, although the term URL is mostly heard.
URL	Uniform Resource Locator [RFC2396]. The address that defines the route to a file on the Web or any other Internet facility. URLs are typed into the browser to access Web pages, and URLs are embedded within the pages themselves to provide the hypertext links to other pages. The URL contains the protocol prefix, port number, domain name, subdirectory names and file name. Port addresses are generally defaults and are rarely specified. To access a home page on a Web site, only the protocol and domain name are required.
User	Person who interacts with the user agent to view, hear, or use a resource.
User Agent	Any software or device that interprets WML, WMLScript, or other resources.
W3C	World Wide Web Consortium. An international industry consortium founded in 1994 to develop common standards for the World Wide Web. It is hosted in the U.S. by the Laboratory for Computer Science at MIT.
WAE	Wireless Application Environment
WAP	Wireless Application Protocol. Determines how wireless devices utilise internet content and other services.
WAP gateway	Software that takes raw WML data and compiles it for the microbrowser and vice versa.
WBMP	Wireless BitMap. Image format used in the Wireless Application Protocol.
Web clipping	Extracting relevant information from a Web page for display on a smart phone or PDA.
Web Server	Network host that acts as an HTTP server. A computer that provides World Wide Web services on the Internet. It includes the hardware, operating system, Web server software, TCP/IP protocols and the Web site content (Web pages).
Wireless Modem	A modem and antenna that transmits and receives over the air. Wireless modems come in several varieties, including units for CDPD, ARDIS, Mobitex, Ricochet, 802.11, OpenAir, BellSouth Intelligent Wireless Networks and other proprietary products.

Wireless Portal	A Web site that supports a user with a smart phone or alphanumeric pager. It may offer a variety of features, including providing a springboard to other (WAP based) wireless sites, the ability to select content to be pushed to the user's device as well as providing a point of entry for anyone to send the user a message.
WML	Wireless Markup Language. A markup language for devices using WAP. It is based on the Handheld Device Markup Language (HDML). Ordinary web browsers cannot read WML.
WMLS (WMLScript)	Wireless Markup Language Script. A subset of JavaScript, used to program mobile devices.
WSP	Wireless Session Protocol. This protocol sends smaller amounts of data than HTML/HTTP.
WTA	Wireless Telephony Application. A framework for accessing the telephony related functions in a mobile terminal.
WTAI	Wireless Telephony Applications Interface
WTLS	Wireless Telephony Layer Security
WTP	Wireless Transport Protocol
WWW	World Wide Web
XML	Extensible Markup Language. An open standard for describing data from the W3C. It is used for defining data elements on a Web page and business-to-business documents. It uses a similar tag structure as HTML; however, whereas HTML defines how elements are displayed, XML defines what those elements contain. HTML uses predefined tags, but XML allows tags to be defined by the developer of the page. Thus, virtually any data items, such as product, sales rep and amount due, can be identified, allowing Web pages to function like database records. By providing a common method for identifying data, XML supports business-to-business transactions and is expected to become the dominant format for electronic data interchange

Appendix C

Class Diagram