International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 4, Jul - Aug. 2013 pp-2048-2055 ISSN: 2249-6645

WAP- Mobile Personal Assistant Application

Adnan Affandi[1], Abdullah Dobaie[2], Mubashshir Husain[3]
Electrical and Computer Engineering Department, King Abdul Aziz University P.O. BOX: 80204, Jeddah 21589

ABSTRACT: *Wireless Application Protocol (WAP) is a result of continuous work to define an industry-wide specification for developing applications that operate over wireless communication networks. The scope for the WAP Forum is to define a set of specifications to be used by service applications.*

Not everybody has a computer, but most of the people have mobiles. And not everybody knows how to use the internet, but most of the people have the ability to use the mobiles. This paper exploits this point and produced a unique novel service that gives everyone the access to the internet using their mobiles, which can be used in many applications such as:

- *Checking train table information*
- *Ticket purchase*
- *Flight check in*
- *Viewing traffic information*
- *Checking weather conditions*
- *Looking up stock values*
- *Looking up phone numbers*
- *Looking up addresses*
- *Looking up sport results*

And much more in these aspects. The Mobile Personal Assistant Application which has been developed by using WAP was the first of its kind here, where software has been developed.

I. INTRODUCTION

You and millions of other people around the world use the Internet every day -- to communicate with others, follow the stock market, keep up with the news, check the weather, make travel plans, conduct business, shop, entertain yourself and learn. Staying connected has become so important that it's hard to get away from your computer and your Internet connection because you might miss an e-mail message, an update on your stock or some news you need to know. With your business or your personal life growing more dependent on electronic communication over the Internet, you might be ready to take the next step and get a device that allows you to access the Internet on the go. That's where **wireless Internet** comes in. You've probably seen news or advertising about cell phones and PDAs that let you receive and send e-mail.

1.2 WML (Wireless Markup Language)

WAP uses Wireless Markup Language (WML), which includes the Handheld Device Markup Language (HDML) developed by Phone.com. WML can also trace its roots to eXtensible Markup Language (XML). A markup language is a way of adding information to your content that tells the device receiving the content what to do with it. The best known markup language is Hypertext Markup Language (HTML). Unlike HTML, WML is considered a meta language. Basically, this means that in addition to providing predefined tags, WML lets you design your own markup language components. WAP also allows the use of standard Internet protocols such as UDP, IP and XML.
There are three main reasons why wireless Internet needs the Wireless Application Protocol:

- Transfer speed
- Size and readability
- Navigation

Most cell phones and Web-enabled PDAs have data transfer rates of 14.4 Kbps or less. Compare this to a typical 56 Kbps modem, a cable modem or a DSL connection. Most Web pages today are full of graphics that would take an unbearably long time to download at 14.4 Kbps. Wireless Internet content is typically text-based in order to solve this problem.

The relatively small size of the LCD on a cell phone or PDA presents another challenge. Most Web pages are designed for a resolution of 640x480 pixels, which is fine if you are reading on a desktop or a laptop. The page simply does not fit on a wireless device's display, which might be 150x150 pixels. Also, the majority of wireless devices use monochrome screens. Pages are harder to read when font and background colors become similar shades of gray.
Navigation is another issue. You make your way through a Web page with points and clicks using a mouse; but if you are using a wireless device, you often use one hand to scroll keys.

WAP takes each of these limitations into account and provides a way to work with a typical wireless device.

1.3 Wireless Application Protocol

Here's what happens when you access a Web site using a WAP-enabled device:

- You turn on the device and open the minibrowser.
- The device sends out a radio signal, searching for service.

- A connection is made with your service provider.
- You select a Web site that you wish to view.
- A request is sent to a gateway server using WAP.
- The gateway server retrieves the information via HTTP from the Web site.
- The gateway server encodes the HTTP data as WML.
- The WML-encoded data is sent to your device.
- You see the wireless Internet version of the Web page you selected.

To create wireless Internet content, a Web site creates special text-only or low-graphics versions of the site. The data is sent in HTTP form by a Web server to a **WAP gateway**. This system includes the WAP encoder, script compiler and protocol adapters to convert the HTTP information to WML. The gateway then sends the converted data to the WAP client on your wireless device.

What happens between the gateway and the client relies on features of different parts of the **WAP protocol stack**. Let's take a look at each part of the stack:

- **WAE** - The Wireless Application Environment holds the tools that wireless Internet content developers use. These include WML and WMLScript, which is a scripting language used in conjunction with WML. It functions much like Javascript.
- **WSP** - The Wireless Session Protocol determines whether a session between the device and the network will be **connection-oriented** or **connectionless**. What this is basically talking about is whether or not the device needs to talk back and forth with the network during a session. In a connection-oriented session, data is passed both ways between the device and the network; WSP then sends the packet to the Wireless Transaction Protocol layer (see below). If the session is connectionless, commonly used when information is being broadcast or **streamed** from the network to the device, then WSP redirects the packet to the Wireless Datagram Protocol layer (see below).
- **WTP** - The Wireless Transaction Protocol acts like a traffic cop, keeping the data flowing in a logical and smooth manner. It also determines how to classify each transaction request: Reliable two-way Reliable one-way Unreliable one-way The WSP and WTP layers correspond to Hypertext Transfer Protocol (HTTP) in the TCP/IP protocol suite.
- **WTLS** - Wireless Transport Layer Security provides many of the same security features found in the Transport Layer Security (TLS) part of TCP/IP. It checks data integrity, provides encryption and performs client and server authentication.
- **WDP** - The Wireless Datagram Protocol works in conjunction with the network carrier layer (see below). WDP makes it easy to adapt WAP to a variety of bearers because all that needs to change is the information maintained at this level.
- **Network carriers** - Also called **bearers**, these can be any of the existing technologies that wireless providers use, as long as information is provided at the WDP level to interface WAP with the bearer.

Once the information is received by the WAP client, it is passed to the **minibrowser**. This is a tiny application built into the wireless device that provides the interface between the user and the wireless Internet.
The minibrowser does not offer anything more than basic navigation. Wireless Internet is still a long way from being a true alternative to the normal Internet. It is really positioned right now for people who need the ability to connect no matter where they are. The WAP Forum is continually working on the specifications of the WAP standard to ensure that it evolves in a timely and useful manner.

II. CREATING A DYNAMIC WAP APPLICATION USING ASP

Microsoft Active Server Pages (ASP) is a viable technology for developing dynamic web content. Its popularity can be seen from the many web sites sporting documents with the .asp extension. Because ASP is a server-side technology, it is well suited to creating dynamic WAP applications, especially in the area of database access.

2.1.1 The MPA Application

The sample application that we will be building in this session is a Mobile Personal Assistant (MPA). The aim of this is to allow users to have a mobile time scheduler that enables them to check on their plans wherever they are.

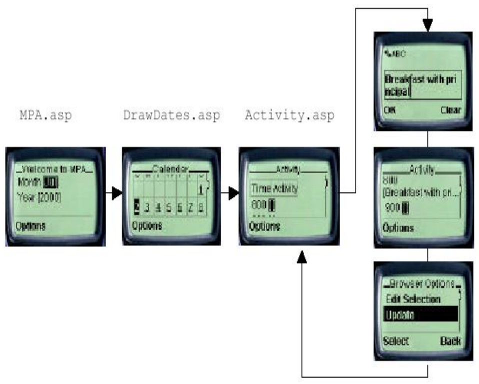
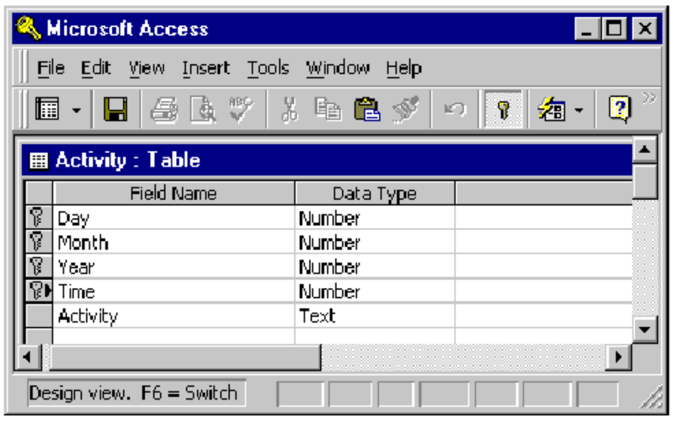


Fig1. Application Overview

As can be seen from the above diagram, there are three decks involved in this application. They are:

- ✓ MPA.asp
- ✓ DrawDates.asp
- ✓ Activity.asp

In implementing this application, the technologies used are ASP and ActiveX Data Objects (ADO). Let's first take a look at the database:

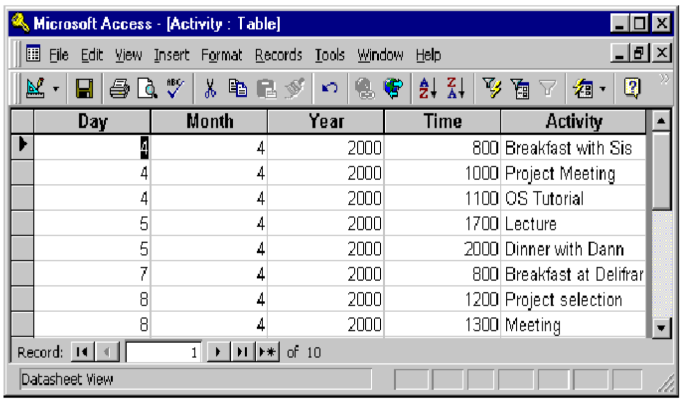


Field Name	Data Type
Day	Number
Month	Number
Year	Number
Time	Number
Activity	Text

Design view. F6 = Switch

Fig.2 Microsoft Access Database

This Microsoft Access database (PIM.mdb) contains a single table for storing the activities of the user. A typical schedule looks like this:


Microsoft Access - [Activity : Table]
File Edit View Insert Format Records Tools Window Help

Day	Month	Year	Time	Activity
4	4	2000	800	Breakfast with Sis
4	4	2000	1000	Project Meeting
4	4	2000	1100	OS Tutorial
5	4	2000	1700	Lecture
5	4	2000	2000	Dinner with Dann
7	4	2000	800	Breakfast at Delifrar
8	4	2000	1200	Project selection
8	4	2000	1300	Meeting

Record: 1 of 10

Datasheet View

Fig.3. MicroSoft Access Database

2.1.1.1 Creating a Calendar

To enable the user to locate a day and month in their schedule easily, its needed to create a calendar on screen. This is achieved by the code in MPA.asp.

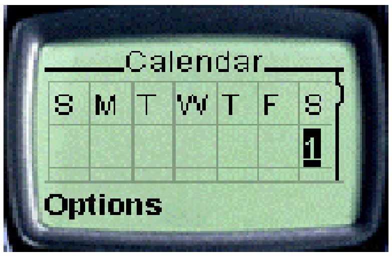


Fig.4. Calendar Created

2.1.1.2 The DrawDates.asp Document

To draw the calendar on the user's screen, so that they can select the day in which to enter their schedule. The generateDate() function is used to generate a table containing all the days in a particular month and year:

Once the month and the year have been selected, they are sent to the DrawDates.asp document using the GET method. A <postfield> element is used to accomplish this:

The DrawDates.asp Document

I need to draw the calendar on the user's screen, so that they can select the day in which to enter their schedule. The generate Date () function is used to generate a table containing all the days in a particular month and year:

2.1.2.2The Activity.asp Document

Once the user has selected a day, they can click on the Follow Link option to retrieve the activity page:

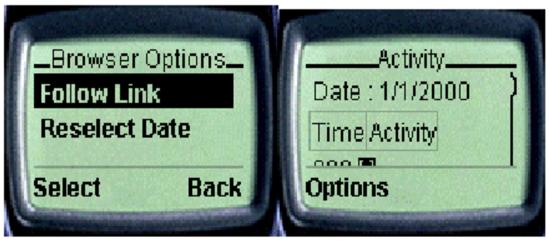


Fig.5 Showing Browser Options & Activity page

Here we have generated all the different time intervals from 8am to 10pm. The time interval is one hour. We use an <input> element to let users modify their activities, and at the same time retrieve the previously saved activity from the database

Notice that in our case we have thirteen different time intervals, and each <input> element has a different name. The first one has the name TimeActivity800, the second one is TimeActivity900, and so on.

Once the user has keyed in their activities, they can proceed to save the activity into the database:

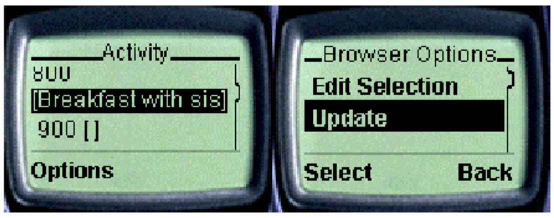


Fig.6 User can save the activity into the Database

2.1.2 Developing Dynamic WAP Applications

Some issues involving in developing dynamic WAP applications:

- ✓ Redirecting pages
- ✓ Caching on WAP devices
- ✓ Cookie support on WAP devices, and maintaining state using the ASP Session object
- ✓ Maintaining state without cookie support
- ✓ Detecting WAP devices
- ✓ Passing values from ASP server-side variables to WML client-side variables, and vice versa.
- ✓ Tips for using ASP to generate WML content
- ✓ Common errors

2.1.2.1 Redirecting Pages

In web application development, it's not unusual to see a browser being redirected to another web page. This may happen, for example, because you don't have the permissions required to view a page, and you must therefore log in before you can do so.

In ASP, you can use the Response.Redirect() method to perform page redirection.

Buffering is disabled by default in ASP 2.0 but enabled by default in ASP 3.0.

If a page is not buffered, the web server will send the HTTP headers to the client before the ASP parser has finished parsing the ASP document. In the case of page redirection, this will cause an error. Try setting the Response.Buffer property to False, and you'll see the error message:

The error message is:

The HTTP headers are already written to the client browser. Any HTTP header modifications must be made before writing page content.

2.1.2.2 Caching on WAP Devices

WAP devices generally cache WML decks. This is to reduce the time that is wasted in trying to reload a page that has been loaded earlier. However, for dynamic content, this is not a good feature — the user may be reading some stale information from the cache.

There are two issues to consider when we talk about caching. They are:

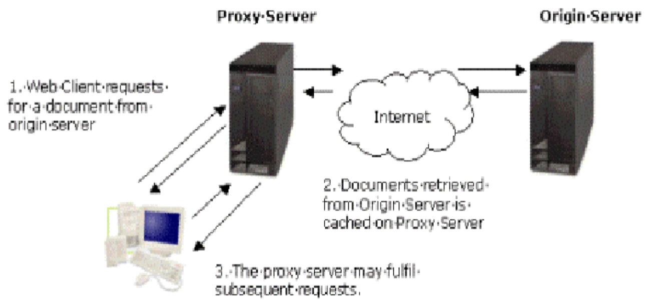
- ✓ Caching by proxy servers
- ✓ Caching by WAP devices

2.1.2.3 Caching by Proxy Servers

Proxy servers help to reduce the time needed to fetch a document from the origin server. In a network, a proxy server may be set up to act as a halfway house between the Internet and the web clients in the various departments. The

proxy server will cache all documents requested by the network, so that it may fulfill subsequent requests for a document retrieved earlier. The trip to the origin server is reduced to making a trip to the proxy server, as the following diagram explains. Proxy servers should not cache ASP documents, as this defeats the purpose of using ASP in the first place. To turn off caching by proxy servers, use the Response.CacheControl property:
<% Response.CacheControl="private" %>

If for any reason you wish to enable caching by the proxy server, you can set the cache control property to public.
<% Response.CacheControl="public" %>
What about caching by a WAP gateway? The WAP Caching Model specification states that a WAP gateway must faithfully implement the role of an HTTP/1.1 proxy with respect to caching and cache header transmission.


Proxy·Server
Origin·Server
1.·Web·Client·requests· for·a·document·from· origin·server
Internet
2.·Documents·retrieved· from·Origin·Server·is· cached·on·Proxy·Server
3.·The·proxy·server·may·fulfil· subsequent·requests.

2.1.2.4 Caching by WAP Devices
Just as a proxy server caches pages, WAP devices generally come with some memory to act as a cache. Remember that bandwidth is still a limitation for current WAP devices, so caching WML pages can generally led to improved performance.

While a cache memory helps to load a page that has been loaded previously without making a connection to the origin server, it is a nuisance where timesensitive pages are concerned. Pages like stock information, weather reports, and online ticketing systems are highly time-sensitive.

By default, WML pages are cached. To force a WML deck to expire immediately after it has been received, I use the Response

The Cache-Control <meta> element allows you to set the caching characteristics of the target WAP device. The content attribute can contain the following values:
- ✓ max–age=time_in_seconds
- ✓ must-revalidate
- ✓ no–cache

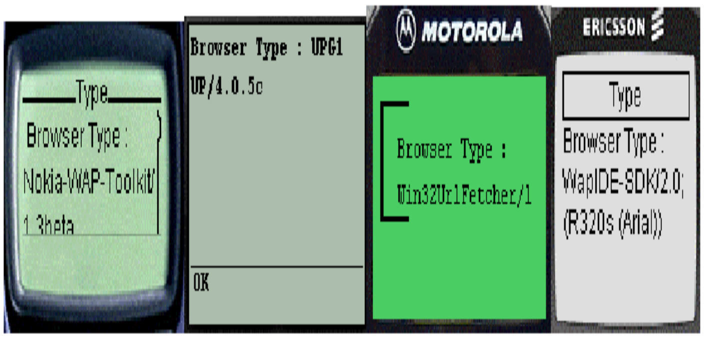
When the value of content is set to "max–age=0", it expires immediately after it has been loaded. It is equivalent to "no–cache".

We have looked at the two methods for controlling caching on WAP devices, but which is better? In general, controlling the HTTP headers is a better option than using the <meta> element. This is because some WAP gateways may not pass caching-related headers to the WAP device, especially the <meta> elements. Also, <meta> elements are not supported by all WAP devices.

2.1.2.5 Cookie Support on WAP Devices
At the time of writing, WAP 1.1 devices have very limited support for cookies. If you're planning to make use of ASP's Session object, you have to be aware of this and make appropriate allowances.
The Session Object and Cookies

The ASP Session object requires the use of cookies in order to work correctly. Provided that's the case, using it is straightforward.

2.1.2.6 Detecting WAP Devices
To detect what WAP browser the user is using, you can use the Request.ServerVariables collection, as I do in Useragent.asp:


Type
Browser Type : Nokia-WAP-Toolkit/ 1.3beta

Browser Type : UPG1 UP/4.0.5c
OK

MOTOROLA
Browser Type : Win32UrlFetcher/1

ERICSSON
Type
Browser Type : WapIDE-SDK/2.0; (R320s (Arial))

It's up to the developer to decide on the level of compatibility that they want to achieve for a particular application. For a list of user agents, point your browser at http://amaro.g-art.nl/useragent/.

If the user agent contains the word "Mozilla", then the user is using a web browser and I need to redirect them to another HTML page using the Response.Redirect() method.

2.1.2.7 Passing Values from ASP to WML

A common question asked by developers new to WML is how can they pass values between WML client-side variables and ASP server-side variables.

Tips for Using ASP to Generate WML Content

When generating WML content using ASP, bear in mind that a deck can contain multiple cards. If your content for each card is generated dynamically, it may have an impact on the efficiency of your web server.

When the ASP parser parses the above deck, both cards will be 'given' the same time. When the user moves from card1 to card2, they will see the same time on both cards. This may not be what we intended — we might want the cards to contain the different times that they were displayed on the WAP device. To solve this problem, I need to make sure that caching is turned off on the WAP device. For this, I can use the Response.Expires property, as described earlier in the paper. However, this causes the deck to be parsed twice by the ASP parser, as the WAP device will have to reload the same deck from the origin server to display the second card.

Remember: caching is at deck level. When a user moves from a card to another within a deck, using a Response.Expires property will not cause a reload of the deck from the origin server.

Hence, when you have dynamic content on multiple cards within a deck, it is important that these cards be separated into different decks. The diagram showing the request and response between the WAP device and origin server is valid provided caching can be done at the card level.

As can be seen, every time the WAP device requests the same deck, the ASP server spends time parsing it. If the deck contained a lot of cards, the time needed would be substantial: this is inefficient. Instead, by breaking the cards down into multiple decks, the time spent parsing individual decks is reduced.

2.1.2.8 Common Errors

Finally, let's briefly examine some of the common errors that developers may encounter when using ASP to generate WML content.

- **Not setting the correct MIME type**

The WML MIME type must be set using the Response.ContentType property, like this:

<% Response.ContentType = "text/vnd.wap.wml" %>

- **Cookies are not supported**

When using cookies or the Session object, always ensure that the target platform is capable of supporting cookies.

The POST method may not work correctly on some WAP devices

When using a <postfield> element, be aware that some WAP devices (the Nokia 7110, for instance) have problems in supporting the POST method.

You need a web server to run ASP!

Very often, people just run ASP documents straight from their local hard disk. ASP documents require a web server (in particular, the ASP parser) to process them, so that the resulting page can be sent to the client. Also important point to note is that you need to use the http:// syntax, and not the physical filename.

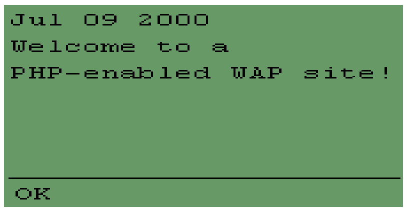
III. PHP/WML

In this paper we have done three integrated PHP/WML examples, showing you:

- **How to insert dynamic content into WML:** Illustrates how PHP can be integrated directly within a WML deck, so that you can feaure dynamic content for the user (in our example, a date).
- **How to send email to a WAP-enabled Device:** Introduces user interactivity by providing a mechanism in which a user can send email.
- **How to interact with a MySQL Database:** Introduces database interaction to enable the display of information in the Web browser, such as a current soccer score.

3.1 How to Insert Dynamic Content into WML

Consider a simple example where a message is generated along with a current date. The example produces the following output:



```
Jul 09 2000
Welcome to a
PHP-enabled WAP site!
```

Code Flow

- Send the proper header to the WAP browser.
- Use PHP's date() function to output the current date
- Use PHP to print out a simple message
  PHP's date function offers a very flexible way in which to display the current date.

3.2 How to Send Email to a WAP-Enabled Device

Let's apply what we've learned thus far by building a basic WML/PHP application to send email to a wireless device. Two decks are involved, the first consisting of three cards, and the second consisting of one card:

- The first deck prompts the user for various pieces of information, namely the destination email address, the message subject, and the message itself.
- The second deck actually sends the message to the destination address and outputs an appropriate message to the user.

Note: This script will not peform any error checking. However, error checking can easily be scripted. For example, a few extra lines of code can be inserted to ensure that the user enters a valid email address, defines a subject, and inserts a message

Code Flow

Deck 1 (email1.wml):

- Send the proper header to the WAP browser.
- Prompt user for Email recipient.
- Prompt user for message subject.
- Prompt user for message.
- Pass information to second deck, entitled email2.wml

Deck 2 (email2.wml):

- Send message to the recipient.
- Inform user as to whether message has been sent.

3.3 How to Interact with a MySQL Database

The final example involves the extraction of data from a MySQL database for display within a WML card. Suppose you wanted to display the outcomes of a series of soccer matches.

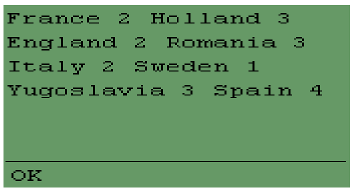
The following MySQL table could be defined:

```
mysql>create table soccer_scores (
   >team1 char(20),
   >score1 int,
   >team2 char(20),
   >score2 int );
```

The following could be stored within the table:

Team1	**score1**	**team2**	**score2**
France	2	Holland	3
England	2	Romania	3
Italy	2	Sweden	1
Yugoslavia	3	Spain	4

We can create a simple script capable of retrieving this data and then displaying it within a WAP browser, as shown below:



```
France 2 Holland 3
England 2 Romania 3
Italy 2 Sweden 1
Yugoslavia 3 Spain 4
```

Code Flow

- Send the proper header to the WAP browser.
- Begin a new WML deck.
- Connect to the MySQL server ("www.zend.com")
- Select a MySQL database ("sports")
- Select information from the MySQL table "soccer_scores"
- If data exists within the table, output it using the format: team1 team1_score team2 team2_score
- If data does not exist, output appropriate message.

IV. WAP SECURITY

Security of applications and computer systems is an issue that, quite rightly, many of IT professionals are concerned about. As corporations have utilized technologies, such as remote access, Java and component technologies, and infrastructural advances like the Internet, to facilitate new ways of working, new ways of doing business with clients, partners and suppliers, and even to create entirely new products, services and business models, the need for mechanisms to secure applications, networks and systems has become more and more important.

- WAP is another technology that extends the reach of communication networks, provides new opportunities for innovative corporations, and adds to the complexity of the environment within which applications need to be designed, built and deployed. 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.
- If your interest in this project is to come out with a definitive statement as to whether WAP is 'secure' or not, you will be disappointed. It is only feasible to make the assertion that WAP is or is not 'secure enough' for a particular application when you understand the security requirements of that application, the environment in which that application is to be deployed, the likelihood that the application will be subject to attempts to compromise its security, and the nature of the attempts that are likely to be made. Even then the statement is only valid until something changes in the environment, or someone discovers a new security exposure in the network, the environment, the technologies used or the platform on which the application is deployed.

V. CONCLUSION

In this paper, a Mobile Personal Assistant (MPA) Application using WAP has been developed.
XML (and friends) is at the moment the best possibility to handle the problems occuring with the many combinations of possible mobile devices, their browser versions and the WAP-gateways they are using. Almost every combination has its effects on the user-interface style. The weak commitment to WAP-standards of some vendors makes difficult for developers of mobile services. Nevertheless XML is powerful enough to handle such difficulties and also showed its power in this prototype implementation.

REFERENCES

[1] Lee,W.M. and Foo,S.M. and Watsom K. and Wugofski T., Beginning WAP, WML, & WMLScript, UK, Wrox Press, 2000.
[2] Arehart C. and Chidambaram N. and Guruprasad S. and Homer A. and Howell R. and Kasippillai S. and Machin R. and Myers T. and Nakhimovsky A. and Passani L. and Pedley C. and Taylor R. and Toschi M., Professional WAP, UK, Wrox Press,2000.
[3] The WAP forumat http://www.wapforum.com. At this site the current WAP specifications can be found, together with suggested future specifications
[4] The WAP group at http://www.wireless3.com. Here you can gain access to a network of professionals, all working towards the widespread dispersal of knowledge and the debeloplemt of WAP technique
[5] The site at http://www.wirelessdevnet.com/ is useful for industry news and other information resources.
[6] The site at http://www.w3schools.com/wap/wap_intro.asp is contain WAP tutrial and WAP resources.
[7] UP.SDK Release 4.1, WML Language Reference, Openwave Systems Inc., 2000 http://www.geek.com/menu_databases.htm
[9] Robin Beaumont 28/03/00 Tel: 0191 2731150.
[10] Wei Meng Lee, Ngee Ann Polytechnic, Singapore W.J. Gilmore August 3, 2000
[11] The site at http://www.stc.com.sa is Saudi Telecom Company web site.