WEBSHOP OPTIMIZATION FOR MOBILE DEVICES

Lahti University of Applied Sciences
Faculty of Business Studies
Degree Program in Business
Information Technology
Thesis
Spring 2008
Mei Long

Lahti University of Applied Sciences
Faculty of Business Studies
Mei Long: Webshop Optimization for Mobile devices
Thesis 65 pages
Spring 2008

ABSTRACT

This thesis deals with the programming technology which can be used to optimize the webshop design for mobile devices.

The purpose of this thesis is to demonstrate several possible solutions when doing implementation of webshop applications for mobile devices. Web application developers and companies who wants to optimize their products for mobile devices could also benefit from this study.

The aim of the theory section is to describe the general coding principles of WML, HTML, CSS, XML, Database, DHTML, Data mining technology specifically for mobile devices.

The study used experimental method to demonstrate several different theoretical solutions for the implementation of the display layer that offers web page rendering according to capabilities of the device, specifically for small-screen mobile phones.

In addition, the topic of individualized content display technique is introduced which will demonstrate how this technique facilitate the work of creating the individualized layer. And how this technique optimizes and affects the webshop user's experience.

Finally, the study will theoretically describes the existing mobile payment method, and hot to implement the method to the webshop.

Keywords: Mobile device, Webshop, Optimization, WML, HTML, CSS, DHTML, Individualization, layer, mobile payment.

1 INTRODUCTION

Have you been brow your wrinkles on the crowded Web pages misdisplayed on your mobile phone? We often encounter such trouble when using mobile phone surf the internet: search list usually displayed normal, but after we click one of those links and left search engine page, the page we spend time to download is often not normal to read, either horizontal scroll bar pulled very long or the pages shoved together. This is because the website is designed in accordance with the general PC Screen resolution, it is impossible to automatically adapt the small screen phones. In order to solve this problem, companies come up with different approaches. For example, Opera is researching one product named Automatic page size scaling browser; some companies work hard on webpage structure design, so that the code can identify with user's device and display targeted webpage content, but it undoubtedly enhances the difficulty of website code design; Domain name Mobi Specifically launched for mobile phones, its content still need to be greatly enriched on WAP.

As we can see, it would be a great attraction for users who would like to make tiny purchases via their mobile devices if e-commerce companies could take the advantage of these technologies, and providing such optimized web shop for these customers. However, the existing mobile browsing still remains a time-consuming, uncomfortable, and expensive process; this is because the mobile device needs long downloading time for product images and HTML tables. Nevertheless, optimizing a web shop for a mobile device do not only concerning device screen size customization, but also we need to consider that how much information to be displayed and how to display it. Possible

solution for example could be displaying only as much text as can be showed on the screen so it dose not need much changes in the webshop structure.

In this thesis, Author is not going to present all the solutions that could possibly solve all the questions mentioned above, yet some realizations will be explained and also technique will be introduced. Therefore, the study is more concentrating on the theoretical part.

1.1 Purpose of the study

The purpose of this thesis is to present several solutions to facilitate the optimization of a web shop for mobile devices. It is important to compare and understand the different solutions and their effects on the application. For example, using an extra individualized layer that could optimize the webshop functions which access the browsing device capabilities, user's navigation preference and interests.

The goal of the study is trying to abstract the best solution through research, compare the different solutions for the webshop optimization for mobile devices.

1.2 Research methods

This thesis is mainly using experimental research method to try different solutions and compare the results and effects. Materials and examples are gathered from different resources some are translated from Chinese to English.

2 Optimization of screen layout:

Different Mobile devices have different screen sizes. First step of the optimization is to ensure the information displayed properly on the device screens. The logical display layer is combined all the techniques that define in which way web site is showed.

In this thesis, different solution has been investigated:

1. Creating a separate skin for mobile device browsers with its entire individual layers.
2. Applying dynamic generated CSS style sheets.
3. Applying dynamic generation of web pages through XML.

This is depending on the properties of the browser and possibly the user preferences. Nevertheless, the above mentioned three solutions have their own pros and cons. The first solution is quite simple and easy to implement, and it seems to be a reasonable selection for such a system. Particularly, when the webshop architecture is based on the Tiles frame work, this solution can take advantage of the given layout architecture. Derived from jsp: "includes" concept, programmer could be more flexible when creating reusable pages by assembling display pages from component parts (reusable tiles).

This solution reduces the amount of HTML codes that needs to be maintained and makes it easier for the later change of website layout. But this solution has its drawbacks, it requires the separation between skins, and different mobile devices might have difficulties of correctly browsing the "tiles" e.g. the device cannot handle the page if skin are treated separately.

Another interesting solution to render the layout is using dynamically generated CSS style sheets. This method would considerably simplify the implementation, while individualized appearance could also be partly implemented simultaneously by hiding

certain object. Only setting display attribute of certain none-interested objects to "none". Unfortunately, not all mobile browsers support the CSS media type mechanism, as the result, this solution might not performed as powerful as earlier described.

Finally, the combination of dynamically generated CSS together with an additional individualized module to facilitate the content presentation could be a competitive solution. The idea of this solution is to store the user's browser and user information in the XML file, then generate the web pages dynamically according to the contents structure of this XML file. This XML file could be generated automatically, together with user's mobile device configuration file and the user profile. This solution could be implemented as an individualized approach that is not only optimizing the webshop for mobile devices, but also for desktops according to client's connection speeds, screen configurations, and the browser type.

2.1 Optimization of layout with CSS

The example as the follows, CSS media Queries could be defined under each stylesheet.

```
<link
        rel = "stylesheet"
        type = "text/css"
        media = "handheld"
        href = "mobile.css"
>
```

```
<link
        rel = "stylesheet"
        type = "text/css"
        media = "screen"
        href = "desktop.css"
>
```

The above two statements could be embedded in any HTML website , by switching these two style sheet settings, user can get correct display no matter on desktop computer or on the mobile device screen. In addition, the media type could also be defined in the CSS style sheets by applying the @media – rule:

```
@media print {
BODY { font-size: 10pt }
}
@media screen {
BODY { font-size: 12pt }
}
@media screen, print {
BODY { line-height: 1.2 }
}
```

2.2 Optimization of layout with SMIL

Another interesting solution is based on the SMIL 2.0, switch-element facilitates the system switching to correct fragments according the client device property and user information.

Example:

```
<switch>
  <audio src="movie-aud-en.rm" system-language="en"
         system-overdub-or-caption="overdub"/>
  <audio src="movie-aud-de.rm" system-language="de"
         system-overdub-or-caption="overdub"/>
  <audio src="movie-aud-nl.rm" system-language="nl"
         system-overdub-or-caption="overdub"/>
   <!-- French for everyone else -->
   <audio src="movie-aud-fr.rm"/>
  </switch>
  <video src="movie-vid.rm"/>
  <switch>
   <textstream src="movie-caps-en.rtx" system-language="en"
          system-overdub-or-caption="caption"/>
   <textstream src="movie-caps-de.rtx" system-language="de"
          system-overdub-or-caption="caption"/>
   <textstream src="movie-caps-nl.rtx" system-language="nl"
          system-overdub-or-caption="caption"/>
   <!-- French captions for those that really want them -->
   <textstream src="movie-caps-fr.rtx" system-captions="on"/>
 </switch>
```

As illustrated above example. In SMIL the < switch > element allows a developer to define a list of optional elements according the device language settings.

2.3 Optimization of layout with TILES

"Tiles" which in fact derived from JavaServerPage "include derivative" concept, based one TAG library, it uses TAG elements as the channel, and developer could define the pages through the tiles. But the pages will be displayed according to the tile components. Furthermore, Tiles are able to generate so-called skins, i.e. a set of tile components that

form a consistent arrangement of a web site. This solution could be used as a mechanism of generate separate skins for mobile device browsers. By reading the client's device property, appropriate skins would be dynamically generated according to user's mobile device attributes.

Example:

```
<tiles:insert page="/basic/myLayout.jsp" flush="true">
        <tiles:put name="title"  value="My first page" />
        <tiles:put name="header" value="/common/header.jsp" />
        <tiles:put name="footer" value="/common/footer.jsp" />
        <tiles:put name="menu"   value="/basic/menu.jsp" />
        <tiles:put name="body"   value="/basic/helloBody.jsp" />
</tiles:insert>
```

3 Individualized Customization

With the fast developed Web-based applications, many website has developed quite complex individualization tools for their end users. These services help led customers to more effectively navigate and obtain adequate information. One of the most popular and widely used solutions is called" social filtering",

3.1 Social filtering

Usually, the preferences of active users are compared in the filter enabled systems with the records of other users in order to find the X most similar referenced user. These data will be used and processed as the prediction of preference of certain product on certain customer. These data could be used as recommendation of top X popular products. This might help customer choosing the suitable products. As this kind of systems is

specialized in collecting and comparing the similarities of online user behaviors, they are hence be called social filtering system or collaborative filtering system. Even though, social filtering system is very popular and widely used, it has been criticized on its limitation of lacking the scalability. For very large amount of data logs, this may cause unacceptable delay for providing recommendations.

3.2 User profile information

The user profile could be defined as a set of information that contents precise features of the user. Many professionals believe that rendering user profile is becoming more and more important in future especially with the new innovation of diversified mobile devices. In the near future, the demand of providing more customized accessing services will be vital important accordingly. This enables mobile device users to find their preferred contents or links more easily. User profile data could be collected in the manner of ambiguously or accurately. System can collect user profile data accurately only if the user actively visits the site and constantly providing the data of their interests or preferences. Users are allowed to control their profile information; this method can take different forms such like filling online forms, doing the survey or register their personal information. In addition, users might provide the information by rating or ranking the certain product or give feedbacks. Website could take the advantage of this method in order to let their customers tell directly what they really need and how they need it. When a website only wants to collect ambiguous user profile data, in fact just on the opposite way of precise profile collection, users are not required to participate any activities on the

site, the system only retrive the information from the system logs or other returned script data, yet the amount of information would be very limited.

4 Mobile devices and World Wild Web

Some years ago, mobile devices which functioned with World Wide Web started entering the market. At the moment, internet browsing becomes a normal function in mobile phones. But the real situation of the mobile devices specifically for mobile phones is that, people are rarely using it to access the internet. The website content can hardly adapt to different mobile device browsers. Another problem is that network appliances have low network bandwidth, this will led to slow access to the rich media content pages. Besides, most network appliance has limited computational power that makes content customization "mission impossible". Some internet portals are offering WAP and XHTML formatted website which optimized for different mobile devices. For example, Google has made its Google search page more appropriate displayed for mobile devices.

4.1 WML and WAP devices

In order to get better understanding and usage of WML language, WML developers need to have good knowledge about the characteristics of WAP devices which support WML language.

Generally speaking, WAP devices which support WML usually have the following characteristics:

- Smaller compared to ordinary personal computers;
- Limited memory capability, and its CPU performance is limited;

- Low communication bandwidth, longer delay.

Let's take mobile phone, PDA as an example, WML support equipment mainly have the following characteristics:

1. The device has one display screen,
2. Allow user to navigate content by using arrow buttons;
3. Support ASCII code printing;
4. There are usually two programmable function keys which are Accept key and Options key, usually located close to the bottom of the screen;
5. Usually have navigation key "Prev".

4.2 WML program structure and composition

Example:

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

<card id="card1" ontimer="#card2" title="Tookit Demo">
        <timer value="50"/>
          <p aligh="center">
                  </br></br></br>
                  <big>
                          <!--Write your card implementation here.-->
                          Welcome to....
                  </big>
         </p>
</card>

<card id-"card2" ontimer="#card 3"title="Toolkit Demo">
        <timer value="50"/>
            <p align="center">
                    <br/><br/>
                <b>
                      The Nokia<br/>
```

```
                </b>
                    Wireless Application Protocol
                </u>
            </p>
</card>

<card id="card3"title="Toolkit Demo">
    <p align="center">
        <br/><br/><br/>
      <big>
        <i>
          Toolkit
        </i>
      </big>
   </p>
</card>
</xml>
```

The program will be run in the WAP mobile phone; three pages will be display on the screen. First page, "Welcome to ..." second page " The Nokia Wireless Application Protocol ...", the final page "Tookit!." Heading "Tookit Demo" are displayed on each page, Delay between two adjacent pages is 50, the unit size is 1 / 10 second, 50 is quals to 5 second delay.

4.3 UAProf and CC/PP

If we want to customize the website for different mobile devices browsers, theoretically, we need to find the way to abstract information from mobile device when user send the request to the web server. But is it possible to find such information? And if it is possible, where can we find it? A simple way to find out more about a client requesting is to analyze the contents of the HTTP headers that are sent with the request. And in the HTTP header we can find a link to the browsers' X-Wap-Profile:

User-Agent: OPWV-SDK UP.Browser/7.0.2.3.119 (GUI) MMP/2.0 Push/PO

X-Wap-Profile: http://devgate2.openwave.com/uaprof/OPWVSDK70.xml

This HTTP header shows that the client is an Openwave browser, but in fact we should pay attention to the second line. This line of information is The X-Wap-Profile property which contains the URI of a UAProf/XML document. UAProf is a WAP Forum specification which allows wireless mobile devices to inform their device capabilities to data servers and other network components. So this is actually the correct information we need for identifying the mobile devices. Before going further, I would like to introduce another device profile standard: CC/PP. A CC/PP profile contents the information of device capabilities and user preferences. This is often taken as a device's delivery context, it also can be used as a guide for the website content adaptation for that kind of device. CC/PP is recommended by W3C, it uses RDF as a format to declare the device capabilities and preferences. CC/PP profile will be formed as follows:

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:ccpp="http://www.w3.org/2002/11/08-ccpp-schema#"

xmlns:ex="http://www.example.com/schema#">

<rdf:Description

rdf:about="http://www.example.com/profile#MyProfile">

<ccpp:component>

<rdf:Description

rdf:about="http://www.example.com/profile#TerminalHardware">

<rdf:type

rdf:resource="http://www.example.com/schema#HardwarePlatform" />

<ex:displayWidth>320</ex:displayWidth>

<ex:displayHeight>200</ex:displayHeight>

</rdf:Description>

</ccpp:component>

<ccpp:component>

<rdf:Description

rdf:about="http://www.example.com/profile#TerminalSoftware">

<rdf:type

rdf:resource="http://www.example.com/schema#SoftwarePlatform" />

<ex:name>EPOC</ex:name>

<ex:version>2.0</ex:version>

<ex:vendor>Symbian</ex:vendor>

</rdf:Description>

</ccpp:component>

<ccpp:component>

<rdf:Description

rdf:about="http://www.example.com/profile#TerminalBrowser">

<rdf:type

rdf:resource="http://www.example.com/schema#BrowserUA" />

<ex:name>Mozilla</ex:name>

<ex:version>5.0</ex:version>

<ex:vendor>Symbian</ex:vendor>

<ex:htmlVersionsSupported>

<rdf:Bag>

<rdf:li>3.2</rdf:li>

<rdf:li>4.0</rdf:li>

</rdf:Bag>

</ex:htmlVersionsSupported>

</rdf:Description>

</ccpp:component>

</rdf:Description>

</rdf:RDF>

4.4 Sessions and cookies in mobile device

A session is a lasting connection that influences subsequent steps in the course of viewing the pages. In general, a session begins with the first connection to an application by a client and ends after that client's last connection. However, because of the stateless nature of the web, it is not always possible to define a precise point at which a session ends. How ever, Sessions are very important and useful component in many web based applications. Server by saving relevant information, session creates the connection between user and a server by storing certain identical information for later use. That information is available for future requests. The information in sessions could be very important for security, authentication, or user preferences. Mostly, client application maintains sessions, the server will produce a unique key to identify the session. A

successful session initiation relies on wheather or not the browser is in capability of supplying the key when the subsequent request happens.

Normally cookies are treated as a tool to transmit session keys in HTTP. But not all mobile devices provide support for cookies and sessions. as the result, if HTTP cookies are not supported, we should consider different solutions.

• An intermediate (proxy) could supply this functionality, as available in the WAP protocol. Additionally, since WAP 2.0 WSP15 is a available. WSP provides HTTP/1.1 functionality and incorporates new features, such as long-lived sessions. Suspend/resume.

• URL reloading: Sessions keys are embedded in URLs.

4.5 Mobile Web

The idea of mobile web is to facilitate the variety of mobile devices access World Wide Web more easily. This means that the Internet can be accessed around the world at anytime. Experts believe that the key of the future network lies in mobile devices. Today's Web sites are only optimized for desktop customers. Users are often deceived by the very poor navigation facilities when browsing the Internet and mobile devices Page. It is a time-consuming process when navigating on traditional desktop-based Web site and on mobile phone. Larger image greatly slowed down the loading process. Horizontal and vertical scroll can not meet the needs of navigation, if one day the mobile Internet becomes a reality, and adapted traditional site is urgently needed.

4.6 Principles of Mobile Browsing

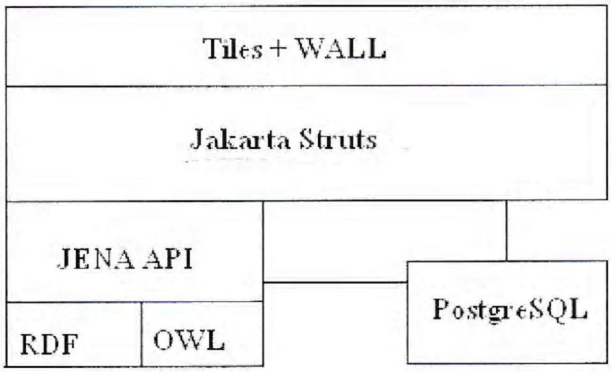
Users will abandon the page if it takes more than 10 seconds to load, this means for general web site design that the information contents in one web page should not take more than 9 seconds to load. However, the internet connection speed has been dramatically increased year by year. 9 seconds could load a lot more information than before. But these web pages are designed for desktops, it is impossible for mobile devices to display the webpage comfortably. There are many differences between Mobile and desktop clients:

- less memory
- small display
- slow connection speeds

In order to meet theses requirements, web site designers need to make a considerable number of modifications on their websites. Websites design for mobile device should follow the principle that it should be as light weighted as possible while preserving necessary information to display.

5 Implementation Details

5.1 Java and Struts Solution



This solution renders the presentation layer with three tiles skins. Each of the skins is designed according to different kind of devices. It provides solutions for three different types of devices: desktop computers, new-generation mobile phones, and mobile devices which support WML, CHTML or XHTML-MP. WALL tag-library is used to implement the CHTML, XHTML-MP, and WML.

Jakarta Struts is implemented to support business logic as the bridge between the presentation layer and the actual business logic. Business logic is the implemented as Java Bean and Struts action. PostgreSQL database is the main database storage for web shop.

Database could be accessed through the struts business logic. The personalization module is implemented on top of the JENA API, a Java-based Semantic Web framework that provided all the necessary methods for the creation of the RDF user profiles.

5.2 ASP and WML solution

5.2.1 Potential of WAP

Nowadays, WAP is a very popular mobile Internet technology. Technically speaking, it is impossible for mobile phones compete with Personal computer. Mobile phones can only display very few characters, and its bandwidth is very limited, but also very clumsy input. But why bother to introduce it? There are two major reasons that make me think that can explain success of WAP technology.

Everywhere with internet at anytime:

If you have mobile phones, you can visit the wap site anywhere at anytime.

Billing agencies:

In the near future, we will be able pay for all our costs through our mobile phones, searching large amount of information. Only there is additional fee added to our mobile phones costs which is "WAP service charges".

In this way we can use our mobile phones to pay for goods, credit cards are no longer necessary anymore. It will become a new consumption patterns.

5.2.2 WAP basis

WAP (Wireless Application Protocol) v1.1 is an open framework; it attempts to regulate mobile phone access information and mobile service industries. This standard is created

by WAP Forum, a civil society organization. Moreover, WAP 1.2 version agreement will be considered to be the last WAP standard.

Basic concept of WAP is to re-use internet agreement, this agreement must be very clearly and easily request WAP services based on the existing technique.

WAP agreement is like a hierarchical structure of computer networks. In the lower level, it is quite different than conventional web application. but the top level, it is in fact our old friend - HTTP. The data is sent and received from a server; information can be stored in the server. Request may also use CGI or other methods, such as ASP.

WAP Forum defines WML (Wireless Mark-up Language) as a xml format language. WML is the simplified HTML. WAP Forum has also designated WMLScript, like JavaScript, it is only used in mobile phones; It will be able to handle client application.

5.2.3 WML

Although I have been explained the communications between the phone and the server, but the client is not actually mobile phone. A software (WAP gateway), in fact is the real client. It converts the request of mobile phones into HTTP request; it will also return the server's response to the phone simultaneously, WAP GateWay will compile this WML document into binary files.

Of course, the simulator does not need WAP gateway to convert the WML files, and WML files will be downloaded directly from the server (or directly open the WML file in the local device), at the same time shows it in the mobile phone screen.

5.2.4 WAP emulators

In order to use WAP services, for the people like me who wants to test it in the personal computer, we will need a WAP simulator. You can go to download a Nokia Toolkit from Nokia website. At the same time, you have to install Java Virtual Machine 1.2.2 which can be downloaded free of charge from the company SUN.

5.2.5 WML structure

WML is a document composed of many tiles, a tile contains many cards. In practice, you can put a tile as a HTML consists of many name tags. Mobile phone downloads a complete tile, but only showed a card at one time. Browse different places through links. We will see a simple WML documents, you have to understand the basic point of XML.

```
<?xml version="1.0"?>
<!DOCTYPE WML PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<!-- root element -->
<card id="card1" title="Example 1">
<p> <!—card only support p and do-->
<do type="accept" label="go to card 2">
<go href="#card2"/>
</do>
This is the first card.
</p>
```

```
</card>
<card id="card2" title="Example 1">
<p>
This is the second card.
</p>
</card>
</wml>
```

Once the phone received this block, it will show a card, when you press the left button, you will see a second card.

5.2.6 Another way to book your movie ticket

So far I have explained the most basic WAP and WML the concept, in this section we will look at how the ASP and WAP combined.

Would it be nice that the cinema offers a phone-based system to sell tickets?

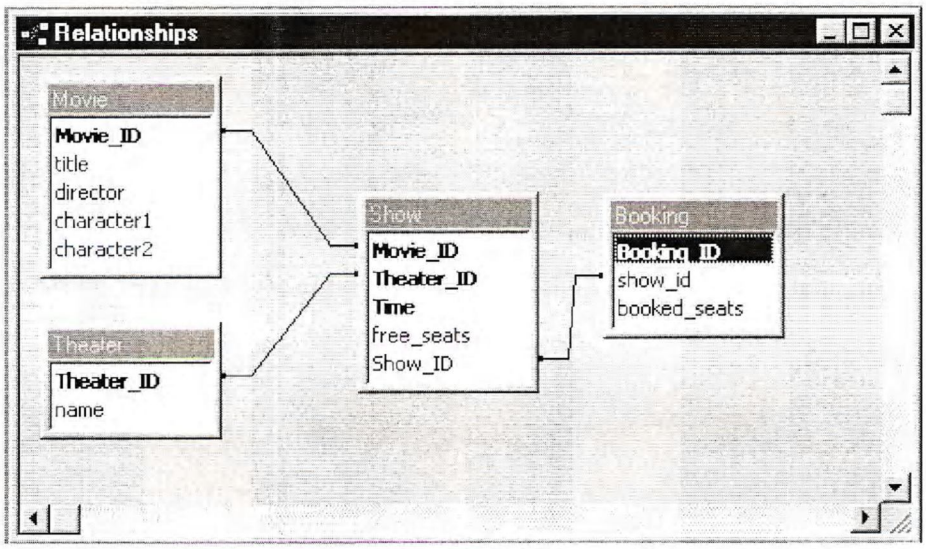
Here you can see a simple demonstration of WAP applications which allows mobile phone users booking movie ticket through their phones: A surely promising service. Users do not have to remember those annoying code, they can directly select from the menu of movies and cinemas, and the user does not need to be certified. In this example users were asked in the film 40 minutes before the start of payment, but in real life, users can also phone to checkout.

This application is created under the conditions that a film is played in a number of cinemas at the same time, a cinema play different movies at different times.

Many issues about possible mistakes were not considered, because this is not the focus of this chapter.

To be simple, we use Access database as the database for the application; of course, the real system will not use it. There is no need to change code when using other databases such as SQL Server,

Database charts



Movie and Theater table is clearly necessary. Show table is used to track how many seats can be sold.

How to debug this application?

In order to access the WAP service, you will need a WAP emulator, this procedure primarily in the Nokia Toolkit 1.2 debug.

Choose the film

Select a movie from the list:

Here is the code:

The following is quoted fragment:

<! - # Include file = \ "conn.asp \" -> <% 'send the right MIME type

Response.ContentType = \ "text / vnd.wap.wml \"

The first thing is WML statement, if the simulator can not declare the XML in the right place, WML will not accept the request, even Nokia 7110 can ignore this, but this can not guarantee that the application will work on other phone lines, so you have to do so. Similarly, you have to set MIME type correctly.

<? Xml version = \ "1 \">

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

Before Entering the next card, there will be a flashing screen to display a WBMP picture.
This event is determined by the timer attributes; here we set the time for 5 seconds.
Ontimer event is triggered another card.
< card id=\"splash\" ontimer=\"#card1\" title=\"Welcome to\" > < timer value=\"50\"/ >

< p align=\"center\" >

< br / >

< img src=\"pix/logo.wbmp\" alt=\"WAP movies\"/ >
< /p >
< /card >
< card id=\"card1\" title=\"choose a film\" >
< %
sqlQuery = \"SELECT [Movie_ID], [title] FROM Movie\"
set rsMovies = conn.Execute(SQLquery) :
% >
< p >
< select name='movie' >
```

```
< %
Do while not rsMovies.eof
response.write(\"< option value=\" & rsMovies(\"Movie_ID\") & \"' >\" & rsMovies(\"tit
le\") & \"< /option >\" & vbcrlf)
rsMovies.MoveNext
loop % >
< /select >
```

This part is the focal point of this chapter. It is nothing special for an asp programmer Reading data from database, but this simple application appears very different when it is used in one kind of brand-new server now. This has made me believe even more strongly that ASP is better than Java servlet when building the WML-based commercial website.

```
< small > < anchor title=\"next!\" >Next

< go href=\"step2.asp\" method=\"get\" >
< postfield name=\"movie\" value=\"$(movie)\" / >
< /go >

< /anchor >
< /small >
< /p >
```

The form can be submitted through the function (do and anchor) to complete. In this example, form is submitted simply through embedded links. It makes the user interface more intuitive and friendly. Post fields is almost like hidden in the HTML, but unlike html WML variables do not need to transmit via JavaScript, it can be directly written in WML. In this example, data transmission methods are get and post; in the post method WAP specifications also support these two methods. the simulator can also recognize it, but unfortunately, in the Nokia 7110, the post method will not be supported.

Choose the cinema and movie time

This part of the code allows users to choose what time to see the movies, all the records in accordance with a pre-selected the film in the last card.

```
movie_id = Request(\"movie\")
sqlQuery = \"SELECT title FROM movie WHERE Movie_id = \" & movie_id
set rsMovie = conn.Execute(sqlQuery)
movie_title = rsMovie(\"title\")
:
sqlQuery = \"SELECT [name], [time], [show_id] FROM Show, Theater \" &_
\"WHERE show.movie_id = \" & movie_id &_
\" AND theater.theater_id = show.theater_id\"
set rsShows = conn.Execute(SQLquery)
```

If you have carefully studied this code, you will want to use the Session to store the movie information, it is simpler make inquiry on this page. It is also unfortunate that Session need the cookies support, although this is supported in WAP standard, but Nokia 7110 dose not support this approach yet, which means that we still can not use the session in WAP services.

The following code is very interesting:

```
< select name='show' >
    < %
    Do while not rsShows.eof
response.write(\"< option value='\" & rsShows(\"show_id\") & \"' >\" & Left(rsShows(\"n
ame\"),cutter) & \" (\" & rsShows(\"time\") & \")\" & \"< /option >\" &vbcrlf)
    rsShows.MoveNext
    loop % >
    < /select >
   Dim cutter
    if InStr(Request.ServerVariables(\"HTTP_USER_AGENT\"), \"Nokia7110\") then
    cutter = 12
    else
    cutter = 7
    end if
```

This code displays different options based on different device. We have sufficient reasons to do so. Nokia Toolit 1.2 simulator like to cut options into only a few characters, and we want to show the film name and release time, we will be forced to reduce the number of the characters of the film name. but this problem dose not exist in the real mobile phone, so we must first determine the type of the device.

Tickets

The next step is to allow users to choose the number of tickets. This part of code shares some similarity with other part. System will still need to search the same data from the database, because the Session can not be really used in the WAP mobile phones, it needs to confirm some of the data to see whether there are seats available.

```
SQLquery = \"SELECT * FROM show WHERE Show_id = \" & show_id
 set rsShow = conn.Execute(SQLquery) :
 seats = rsShow(\"free_seats\") :
 if seats = 0 then
 Response.write(\"Sorry, no more seats\")
 rsShow.close
 set rsShow = nothing
 Response.write(\"< /p >< /card >< /wml >\")
 Response.end
 else
 if seats > 6 then 'book up to 6 tickets or max available
```

```
max_seats=6
else
max_seats = seats
end if
end if
% >
< %=movie_title% > at < % =theater_name% >
< select name='ticket' >
< %
dim i
i = 1
Do while i < = max_seats

response.write(\"< option value=\" & i & \"' >\" & i & \" ticket(s)\" & \"< /option >\" &v
bcrlf)
i = i + 1
loop % >
< /select >
```

Save the tickets amount.

Now we already have all the data needed, we can save them now:

```
tickets = Request(\"ticket\")
        :
        free_seats = rsShow(\"free_seats\")
        :
        free_seats = free_seats - tickets
        :
        SQLUpdate = \"UPDATE Show \" &_
        \"SET Show.free_seats=\" & free_seats & \" \" &_
        \" WHERE Show_ID=\" & show_id
        conn.Execute(SQLupdate)
        SQLquery = \"SELECT max([Booking_ID]) as bookingnumber FROM booking\"
        Set rsBooking = conn.execute(SQLquery)
        maxbookid = rsBooking(\"bookingnumber\") + 1
        SQLinsert = \"INSERT INTO Booking ( show_id, booked_seats ) \" & _
        \"VALUES (\" & show_id & \"', '\" & tickets & \"')\"
        conn.Execute(SQLinsert) % >
        You have booked < %=tickets% > ticket(s) for < %=movie_title% >< br / >
        The show will take place at < %=theater_name% > (< %=time% >)
        < br / >
        Your reference number is < %=maxbookid% >
```

The transaction completed.

5.3 PHP and WAP solution

5.3.1 General explanation about PHP

As we have introduced WAP in the previous chapter, we should begin to gradually understand why combine PHP and WAP services to make applications. PHP is different than HTML, WML, JavaScript and Java. PHP is a server language running on the server end. JavaScript, Java, WAP, WMLScript running on the client side, but most of those languages can be easily integrated with PHP.

PHP has great flexibility in the World Wild Web; it can generate any needed HTML code, and even JavaScript code. Similarly, we can still use such dynamic, flexible features of

PHP language with WAP, generate flexible WML code. In this way, PHP is enabled to create WAP services.

In addition to its flexible features, another reason to combine PHP and WAP service is that PHP can greatly facilitate the use of the database. Users can access PHP Oracle, Sybase, MS SQL, MySQL, dBase and Informix any database type which support ODBC standard, this feature just fulfill the needs of the WAP business.

In practical, PHP documents are general needed, when a user sent the request to the server ask to access the PHP documents, the server will produce HTML or WML content through PHP code, and send it to the browser or WAP terminals.

5.3.2 PHP-WML

PHP system platforms, working methods, and installation methods, which are not in the scope of the discussion in this chapter, after all, our focus is on how PHP collaboration with the WAP. Generally speaking, in order to make PHP work properly, we need a PHP module for Web servers, or servers that support PHP, Apache is the world's most popular Web server. In addition, we also need to install the PHP software and databases such as MySQL.

Below we discuss how to produce WML PHP code.

In WWW, usually PHP generates the first line in the content:

content-type: text / html

However, WAP terminal do not know how to compile is the title. The unit that WAP download from the server terminal is Deck, the unit which are navigated is Card. Generally a Deck is the formation of a WML document. Well, for WAP services, PHP files often need to contain the following code:

```
header("Content-type:text/vnd.wap.wml");
      echo "<xml version=\"1.01">\n"
      echo"<! DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML
1.1//EN\"\"http://www.wapforum.org//DTD//wml_1.1.xml\">\n";
```

The above three lines of code produce the head of WML documents (Deck), this enables WAP terminals identify whether Deck which has been downloaded is in WML format, and then display the contents of the remaining Deck.

Below is one of the simplest Deck, which WAP terminal display the "Hello World".

```
<?xml version="1.0" encoding="ISO-8859-1"?> <! DOCTYPE wml PUBLIC "-//WAPFORUM//DTD
WML 1.1//EN" "http://www.wapforum.org//DTD//wml_1.1.xml">
      </xml>
       <!--Nokia Parser Info:Phone = Nokia 7110; Height = 90; Width = 130; CurrentDeckSize = 38;
MaxDeckSize = 1600; CardsOnEachLine = 5; CardsVerticalGap = 30-->
      </card id="card1" ordered="true" newcontext="false">
      <p align="left">
      Hello World
      </p>
      </card>
      </wml>
```

We have established the corresponding PHP file as follows:

```
<?php
    header("Content-type:text/vnd.wap.wml");
    echo "<?xml version=\"1.0\">\n";
    echo "<!DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML 1.1//EN\"
\"http://www.wapforum.org//DTD//wml_1.1.xml\">\n";
    echo "\n";
     echo "<!--Nokia Parser Info: Phone = Nokia 7110; Height = 90; Width = 130;
CurrentDeckSize = 38; MaxDeckSize = 1600; CardsOnEachLine = 5; CardsVerticalGap
= 30-->\n";
    echo "<card id=\"card1\" ordered=\"true\" newcontext=\"false\"> ";
    echo "<p align=\"left\">\n";
    echo "Hello World";
    echo "</p>";
    echo "</card>";
    echo "</wml>";
    ?>
```

We can keep the documents as index.php, when the WAP terminal accesses the site, the Web server will be automatically generated the above WML content in accordance with the contents of the above index.php, and send it to Gateway for processing. WAP terminal received the Deck, it will be displayed on the screen as "Hello World" message.

The collaboration of PHP and WAP is not just relying on several "echo" statements to transmit WML browser code. PHP in the higher end of the WAP application will, of course, has something to do with PHP database operation and the object-oriented programming knowledge. PHP supports object-oriented (OOP), which enhance the sustainable development of PHP. When we develop WAP pages, PHP's object-oriented features, will make the code more flexible, more versatile, system will be more completed, the level of clarity and reusability could be improved also.

Here, we will tell how to use the object-oriented programming approach to the development of PHP and WAP.

5.3.3 General plan

As an Object-Oriented Programming, it naturally needs to determine the link between objects. For developing WAP pages, first identify the types of modules or elements, which can then be identified to species target for the number of the objects.

Deck is the WAP browser's smallest downloading unit; Deck can contain a number of Card. To simplify the design, we only need to consider the situation of one card. Thus, Card and Deck has one-to-one relationship, then we can only recommend one object for them. (if one Deck can contents many card, then we need to create object for them separately.)

For Card, which can contain text (Text), images (Image), form (Table), the input (Input), links (Hyperlink), separate objects are required for each element.

In practical, for PHP documents are needed. when a user send the request to the server, the server will prepare the HTML or WML code, and send it to the browser or WAP terminals.

Continue to refine, forms (Table), in order to facilitate management, we can build (the Row) object, and each object will contain a number of modules (Cell) object, here object

modules can be text, it can also be a Image, therefore do not need extra definition. In addition, the input (Input), select (Select), action (Do), and other server-side interactive elements, we have a definition of their parent Objects (Interact), which working as a container, it facilitate understanding and programming also.

5.3.4 Object-oriented features

For the realization of specific object, we need to follow PHP syntax. Object Includes attributes, attributes are used to describe the characteristics of the object. In other words, if we can identify the attributes, then we can easily identify the Object. Methods in the object described the operation of these methods; they may include attributes settings, read, display, etc... Therefore, when we define the above mentioned objects, we should provid as defined by the need to provide the complete attributes and the necessary methods.

Deck cited the example, if we define a Deck (Card), we need to know all the elements within it, the definition of these elements is that we have defined before Image, Text, and other objects, Deck should includes these attributes of these objects. In addition, Deck also includes some set of parameters, such as header and so on, we can also define the "Title" and other attributes. As for the methodology, constructor function and the creation function are the most important, Since the constructor function is the initialization in the creation of an object (class), such as the page title settings; creation function creates WML code of the objects in order to form a complete WML pages. The object-oriented method simplifies the creation of function, because every time we only need to call the creation function of that object or attribute. For instance, object Deck can create the table code only by calling directly its Table functions. Similarly,

table can create rows by calling Row Table creation function. Some of the nested object mechanism will undoubtedly simplify the development process.

In addition, the Deck, as its attributes can be Text, Image, Table, Form and a Link. It would need to determine the type of each of its attributes, for the corresponding rendering. In order to achieve this, object should define one method in order to get its object type. the methods (functions) will return a identifier for Deck to confirm. For example, the Text object, the returned object type value could be 0. Image object, it can return 1. The Interact object has the same problem also.

In the process of preparing the code, we should try our best to maintain the readability and maintainability of the code; we should minimize none-sense definition of the figures and identifiers. For example, the definition of the returned value 0 or 1, it is difficult to understand what it is. To solve this problem, we can use "define" function, identify the number with the symbol. For example, that we can define I_AM_TEXT as 0, I_AM_IMAGE as 1, thus we are able to increase readability of the code.

For the definition of the object, we can generate an INC file and always kept it, PHP files can always call it for generating the pages. It should be clear that what we have described objects above do not generate any pages; those are only the elements and methods which we can use to generate the pages. The real work of making pages are completed by those php by using the objects and method..

6 Summary

The aim of this thesis is to present the possible solutions to optimize an existing web shop system for mobile devices. In the first chapter, Author addressed several solutions for optimizing the existing web shop to meet the requirement of different mobile devices. It is recommended that this transformation must separate the content and the layout layer. Based on this approach, there are three solutions have been introduced for the webshop adaptation for mobile devices.

1. implementation of separate skin mechanism
2. dynamic CSS style sheets generation
3. dynamic webpage generation based on XML

The JSP pages, SMIL library, Tiles and CSS style sheet are enabling developer render the the layout and markup of the webshop into different sections based on the client's device capabilities. This has greatly improved the implementation of different presentation mechanism. This thesis has also proposed two different examples of the presentation layout for mobile browser: one approach is based on CSS media types, and another takes advantage of the tag library, these two approaches seemed to work fine. But one question is that the CSS 2.0 standard has not yet been widely supported by mobile device browsers, what we can hope is that in the future most of the mobile devices will support this standard. Further more , new extension of media types emerged in CSS 3.0 standard will be released and this new standard will improve the mobile we development dramatically. Since CSS media type is not yet available, implementing WALL tag will be another good alternative solution. It has been wildly believed that the interesting tiles abstraction methods could benefits the development of device independent mobile

website. Compare with CSS which allows us to hide certain blocks according the user preferences or device in formation, WALL has much more advantages, as it renders the whole website by creating new markups which are generated for certain device specifications sent by end user. As far as concerned, it is hard to predict which one of these two methods will be more popular in the future.

It is never enough that website only provides advanced navigation solutions which allow the customers comfortably surfing the internet by their mobile devices. Good customer information analysis is another crucial factor that enables the webshop making good business while the customer could be served better. The thesis has introduced one approach that could optimize the current website's customer data analysis. This method is called "social filtering" which collecting and comparing the user's navigation behavior and give the prompt recommendation on the site.

The real situation of the mobile devices is that people are not willing to access internet by mobile phone because the website content cannot display correctly on the mobile phone's screen. Some internet portals are offering WAP and XHTML formatted website which optimized for different mobile devices.

From the technical side, the thesis has been dealing with some terms such like UAProf and CC/PP which allow the developer retrieve the mobile phone information. Session and cookie sections described the general situation of the current standard, and it is still impossible to use session to store the information for example the product id because some type of mobile phone dose not supports this approach.

In the chapter of implementation details, there are three combination of solutions has been introduced. The first one is the Java and Struts Solution, this solution is taking the

advantages of tiles, jsp, wall and css technics together with powerfull Java platform which rendering the database site. All these methods have been explained in the earler chapter. But this solution is still using conventional markup languages to generate tiles according the user's device specification.

Second implementation is very interesting and wildly used in the mobile web. The solution is the combination of ACCESS-ASP-WML. Lively example: the mobile phone – based the movie theater ticks system which seamless cooperate together to provide very practical yet powerful solutions for mobile web shop. In this chapter, the author has been explain the whole system work flow, step by step introduce the meaning and function of different part of WML language, and how it can communicate with the ASP code, moreover the author has explained that how the data are passed through different WML cards, and why not using session or cookie.

The third solution is the combination of Apache->Mysql->PHP->WML making mobile website. PHP language is powerful in its server site database rendering, this is its best nature. And it can generate any client side language code such like XML, HTML,JAVASCRIPT and WML. This is enabling the developer taking the advantages of those features to develop advanced mobile web applications. Same like in the previous chapter. Author has introduced how php can connect to WML, which is by delare the wml in php header e.g. header("Content-type:text/vnd.wap.wml");

Furthermore, PHP is object-oriented programming languages. And WML contents many specific tags such like <card>,<p> and so on. Php could render those tags by classify them into different objects which contents attributes. By defining the constructor and creation function, php could easily and effectively manage the WML tags. With user's request from their device, php could easily collect the user device information, based on that PHP could use those objects render the webpage and database data.

In conclusion, mobile devices still has long way to go in order to compete with desktop in the aspects such as bandwidth limitation, computational power and display technology. Obviously desktop has absolute advantage in those aspects. Yet mobile devices has its own nice features such like it offers greater mobility, light weighted, easy to carry, and you can access internet anywhere at anytime if the above drawbacks could be solved. I certainly believe that mobile web application still has a lot of interesting things to improve. Mobile payment will be available widely in the future. It is no longer an imagination that credit card will disappeared in our daily life; instead one tiny mobile device will make our life much easier.

7 LIST OF SOURCE

Books

Openwave Technology, 2002, Push Backward Compatibility Delivering WAP Push Functionality to Devices, USA California.

William Routt, DEC 2002, Wireless Markup Language (WML) Scripting and Programming using WML, cHTML, and xHTML.

Ray Rischpater, May 2001, Wireless Web Development with PHP and WAP

Li chunguang, June 2001, Manual for WAP development.

Xiang quan , Jan 1 WML and WAP mobile web development using WML, WAP manual

Electronic Resources:

The WAP Forum. Wireless Application Protocol WAP 2.0: Technical White Paper. RetrievedAugust, 2005 from http://www.wapforum.org/what/WAPWhite Paper1.pdf.

R. Jones. Creating Web Content for Mobile Phone Browsers. Retrieved October 29, 2004, from http://www.oreillynet.com/pub/a/wireless/2004/02/06/mobile browsing/.

My Guidelines for Mobile Development Switcher, a useful tool to tell mobile devices from web browsers programmatically. from http://www.passani.it/switcher/

The WURFL, de-facto standard for figuring out device capabilities programmatically Retrieved from http://wurfl.sourceforge.net/java/tutorial.php

The WALL, Create content for mobile using HTML-like tags, and support WML, XHTML and Compact-HTML in one shot! From http://wurfl.sourcef orge.net/jav a/tutorial.php

PHP FUCNTION FOR MAKING ELEMENTS

```
function Make_Element($deck)
    {
    if ($this->attribute & TEXT_BOLD)
    echo "<b>\n";

     if ($this->attribute & TEXT_UNDERLINE)
    echo "<u>\n";

    if ($this->attribute & TEXT_ITALIC)
    echo "<i>\n";

    if ($this->attribute & TEXT_BIG)
    echo "<big>\n";

    if ($this->attribute & TEXT_SMALL)
    echo "<small>\n";

    if ($this->text)
    printf("%s\n", convert_character($this->text));

    if ($this->attribute & TEXT_SMALL)
    echo "</small>\n";

    if ($this->attribute & TEXT_BIG)
     echo "</big>\n";

    if ($this->attribute & TEXT_ITALIC)
    echo "</i>\n";

    if ($this->attribute & TEXT_UNDERLINE)
    echo "</u>\n";

    if ($this->attribute & TEXT_BOLD)
     echo "</b>\n";

    $br_command = "<br/>\n";
    for ($i=0; $i<$this->br_count; $i++)
    echo $br_command;
    }
```

SMIL Example:

```
<smil xmlns="http://www.w3.org/2001/SMIL20/">
      <head>
          <layout>
               <root-layout width="320" height="480" />
               <region id="a" top="5" bottom="100" />
              <region id="b" top="200" bottom="280" />
          </layout>
       </head>
       <body>
                <text region="a" src="text.html"/>
                <text region="b" src="additional_text.html"/>
       </body>
</smil>
```

SWITCH STATEMENT

```
<smil:switch>
        <html:img
                src="bigColorImages/logo.gif"
               title="Company logo"
               alt="logo"
               smil:systemScreenSize="768X1024"
          />

         <html:img
               src="smallMonoImages/logo.gif"
               title="Company logo"
              alt="logo"
              smil:systemScreenSize="160X160"
          />
        <html:img
                src="defaultImages/logo.gif"
                title="Company logo"
                D.2. THE SWITCH ELEMENT 74
               alt="logo"
          />
</smil:switch>
```

8 Appendix

WML example:

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

<card id="card1" ontimer="#card2" title="Tookit Demo">
        <timer value="50"/>
          <p aligh="center">
                </br></br></br>
                <big>
                      <!--Write your card implementation here.-->
                      Welcome to....
                </big>
        </p>
</card>

<card id-"card2" ontimer="#card 3"title="Toolkit Demo">
       <timer value="50"/>
           <p align="center">
                   <br/><br/>
               <b>
                    The Nokia<br/>
               </b>
                    Wireless Application Protocol
                  </u>
            </p>
</card>

<card id="card3"title="Toolkit Demo">
    <p align="center">
        <br/><br/><br/>
      <big>
          <i>
            Toolkit
          </i>
       </big>
   </p>
</card>
</xml>
```

L. Passani and A. Trasatti. WURFL and WALL. Retrieved Mai 5, 2005 from http://wurfl.sourceforge.net/.

W3C. Media Queries, W3C Candidate Recommendation 8 July 2002. Retrieved in August, 2005 from http://www.w3.org/TR/css3-mediaqueries/.

Device Independence Working Group W3C. Authoring Techniques for Device Independence. Retrieved Mai 5, 2005 from http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/.

SYMMWorking Group W3C. Synchronized Multimedia Integration Language (SMIL 2.0). RetrievedAugust, 2005 from http://www.w3.org/TR/2005/REC-SMIL2-20050107/.