Design and Implementation of a Web-Portal for Mobile Devices using WAP/XHTML

Objective

The objective of this practice is to familiarize students with the tools and techniques used for the creation of a WAP/XHTML site. During this practice, the students should learn how to use WEB programming languages; in particular for mobile terminals, and dynamic content.

Tools

The working environment is formed by:

- A Web Server configured for the deployment of WML, WMLScript and XHTML content where we install the designed application. The installation and configuration of a web server is out of the scope of this practice. We recommend the student to install an Apache Web Server in a Linux machine (it must be noted that in case of implementing the application with some specific technology another Web Server may be required. E.g. Apache Tomacat in case of using Java Servlets).
- A Windows computer where we install:
    - The tools to test the designed application:
        - In the case of a XHTML portal a web browser (e.g. Internet Explorer, Mozilla Firefox, Google Chrome) is enough to test the application. Although, if it is possible we recommend to test the application using a mobile phone with XHTML capabilities.
        - In the case of the WAP portal we need a WAP Browser Emulator. In this document we describe the Ericsson WapIDE and the Nokia Toolkit but there are other tools that can be also used for our purpose (WAPProof, WinWap Browse)
    - This computer can be also used to install the necessary development tools. The student is free of using any development tool for both cases the WAP and the XHTML portal (e.g. Xemacs, Eclipse, NetBeans, etc). In this document we describe the Nokia Mobile Internet Toolkit. This is a specific development kit useful to implement both WAP and XHTML portals.

Ericsson WapIDE

Please note that we do not recommend using this tool to program the WAP pages. However, its telephone-emulator/simulator is extremely versatile. We therefore recommend the use of the simulator once the practice is developed to test the application's performance on different emulated mobile phones.

The IDE (Integrated Development Environment) has a help system; please consult it, if you have any questions. In the following, only the main functionalities are going to be explained. The environment has three components: a WAP browser, an application designer tool and a PUSH initiator. The WAP versions with which this environment can work are version 1.1 and 1.2.1. We will focus on the WAP browser in our lab. The WAP browser allows to access WAP web pages as if a mobile phone were used. The interface simulates the Ericsson R520m mobile phone, although from *File->Load device* is possible to load other skins and behaviours. The mobile phones are described by a .dev file, the WAPIDE includes two phones, the R320s and R380s but more phones are available for download from the Ericsson web page.

As in a real mobile phone, the simulator needs a WAP Gateway. Is possible to select the desired Gateway using the *View->Settings menu.* There are several options to see details of the information provided by the browser. From *View->Trace*, a menu is opened with options to define the debug level in the browser.

In order to load a specific URL, it has to be specified in *File->Load URL*. There is an important option: *View->Settings->Cache->Disable Cache* , which can be used to disable the cache. This can be very useful while developing applications.

Some examples for WML and WMLScript can be downloaded from http://lm011.lab.it.uc3m.es/lablis/index.wml .

Nokia Internet Toolkit Development Enviroment

The Nokia Development Environment provides several tools as WAP browser, WAP Gateway and a Toolkit for the WAP applications development. The toolkit supports the creation of different types of content, as WML, WMLScript, XHTML-MP, MMS.... The toolkit includes a User Manual and inline help, which could be very useful while using the IDE. Using the *File->New* menu, is possible to create a file of whatever of the contents previously mentioned. While using the editor, the right click displays a list of possible actions, like file validator, and the possible tags of the language that can be used at this moment.

To see the final results of the application, the Nokia Web Browser can be launched from the environment: first launch the SDK Control Panel (Tools menu). From the Control Panel you should click on the green dot. After the browser is running use the Show button in the Toolkit main window. In the browser, in the Tools Menu is possible to launch a window with insightful diagnostics about the browser operation. This information can be useful for debugging. From the Tools menu is possible (*Tools->Settings*) to configure the WAP gateway, the cache memory etc.

Some examples on XHTML-MP can be found on http://lm011.lab.it.uc3m.es/lablis/ejxhtml.html.

Assignment - Practice Guide

During this practice, a WAP (or a XHTML) site which serves personalized content, is going to be build. This guide does not pretend to describe the site, but to include some interesting aspect that it should have. The practice's evaluation will take into account the originality of the idea, the content, as well as the usability and ease of use (from a Mobile Phone!). The site should be interesting for real users.

Personalization is fundamental for the success of a wireless web-portal, due to the constraints imposed in this type of devices. For personalization purposes, the server should store some information related with the authorized users of the site. As an example, for each user, the server should store a login, password and some kind of profile defining the services expected from the user. The main page should ask for a login and a password, both parameters should be processed by the server to check the authentication of the user. If the check is positive, the server should open a session for the user showing a menu dynamically build depending on the users' profile.

The server can use CGIs, Java Servlets, PHP scripts, or any other approach to serve dynamical pages, the program will treat the users' information creating dynamically a page using the profile, or forwarding the user to the login page if the authentication is not correct. There is no limitation in the possible content of the site; the only requirement is that the content should have some interest for some real group of users. As possible example: some content of interest for universitary students, as timetables, exam dates, lectures information, etc. could be the basis for an application In this case the personalization of the content could be only showing the information related with the lectures of the student. Another possible example is a website featuring on topic in detail.

The users' information has to be stored; this can be done by the use of files or databases. The easiest way, is by means of files (which we recommend) but higher performance can be achieved by using a database. There is no limitation in using open source for the site development; however, it is mandatory to indicate the source of the. A deep knowledge of the code and some adaptation to the particular site is necessary, too.

There is no limitation in the selected programming language used for the development of the site. One possibility is to develop the same site in WML and XHTML to compare and analyze the features/difficulties.

The site must be installed in the configured Web Server. A guide showing the topics while developing the portal, motivation, design and use should be delivered to the teachers. Conclusions about the technology used are welcome, but not mandatory. The information needed to test the site (URL, username, password, how to register, etc.), and the commented code should be included in the guide, which should be delivered to the instructor by email. Please test the developed application extensively, before handing it in fro grading.

Evaluation Criteria

- Please note that it is mandatory that the WAP/XHTML page includes on *dynamic* web pages, e.g. using a database or generated during the runtime of the application. Applications that use static content only will not be graded as sufficiently.
- Originality and technical merit of the proposed solution.
- Good design (usability on mobile phones by real users).
- Robustness (possible problems and errors should be handled).
- Complement the desired characteristics. E.g. consider time limits of sessions after which a reconnect should be enforced, an external webpage can be used for registration to keep the complexity of the application on reasonable level.
- Whatever extension or functionality added to the site and could be of interest.
- Quality of the user guide.

Notes

It is recommended the use of the Nokia Internet Toolkit due to its robustness and easy of use. It can be useful to use the Ericsson browser to see check the result on “real” mobile phones.

References

- *Programming Applications with the Wireless Application Protocol*. Steve Mann. John Wiley & Sons, 2000.
- *WAP – the wireless application protocol: writing applications for the mobile Internet. Sandeep Singhal. Addison-Wesley, 2001.*
- Understanding WAP : wireless applications, devices and services. Marcel van der Heijden, Marcus Taylor editors. Artech House, 2000.
- WMLClub: http://www.wmlclub.es.
- WAP Source Code: http://www.cellular.co.za/download_wap_source_code.htm.
- OMA: http://www.openmobilealliance.org.
- Ericsson: http://www.ericsson.com
- CGIC: http://www.boutell.com/cgic/ (biblioteca en C para programación CGI).
- Ericsson WAP IDE: http://www.ericsson.com/developer/
- NOKIA: http://www.forum.nokia.com
- Nokia Mobile Internet Toolkit