Using Mobile Devices for Information Retrieval in the Academic Environment

Ileana Stănescu, Răzvan Zota, Raluca Stănculescu, Laura Stănescu, Emil Stănescu, *Member IEEE*

***Abstract* — In this paper we set out to explore the various usages of wireless technology, through web services and WAP. The necessity of this paper is given by the continuing interest of web developers to add WAP functionality and support to their web sites, and the versatility of web services. For the practical part of the paper, we will provide as examples two applications, both developed for the academic environment. The first one uses web services and enables users to consult their schedule from mobile devices, on many different criteria. The second one leverages the WAP protocol to provide students with the opportunity to check their schedule and have access to information about professors and subjects.**

***Keywords* — Web Services, WAP, WML, mobile phones, PDA.**

I. Introduction

IN the academic environment, students possess many communication related devices, such as mobile phones or Personal Digital Assistants (PDAs). Students are permanently on the move and in permanent interaction so there is great demand for message interchange. It would also be useful to have access to information about their schedule or on the availability of their professors for consulting hours. Following up, a few models of information are analyzed pertaining to a user from the academic environment, that has at his disposal mobile terminals with some communication and processing facilities, such as a mobile phone or a PDA. Further on, the architecture of a student and professor information system over the schedule and other events is described. Respective of the mobile device that the user is using, he/she can use the browser of the mobile phone to access WML files, using the WAP protocol, or he/she can upload a user application that can be of the JavaME type or Microsoft Compact .NET framework type, application that accesses Web services that reside on the application server. Because of memory and computation power constraints, the implementation of client applications for web services can relay access to much bigger computation resources.

Wireless applications accessed from a mobile phone, provide the user with the information that he wouldn't otherwise have access to on the move. Let's assume that the user (a student or professor) is a hurry, and does not have the time to swing by to check his faculty's notice board, or he/she is just plain ignorant about his/hers professors' email, office no, etc. A mobile phone could go a long way in any such situations. The user could access via WAP, his/hers personal page application and retrieve the so needed information.

Concerning web services, interoperability is a solution for the correlation of information from different information systems, implemented independently on different platforms, and for unitary implementation of some services that can be utilized by different clients independently of the platform that these clients' applications have been developed on.

II. Communication technologies

The Wireless_Application_Protocol (WAP) is an open international standard for applications that use wireless communication. Its principal application is to enable access to the Internet from a mobile phone or PDA [1,3].

A. WAP, WML, XHTML technologies

WAP sites are websites written in, or dynamically converted to, WML (Wireless Markup Language) and accessed via the WAP browser

Latest WAP Forum's specifications are known as WAP 2.0. The most important architectural components of WAP 2.0 are following [2]:

- Protocol Stack Support – WAP 2.0 adds to WAP 1.* stack support for TCP, TLS and HTTP, providing a connectivity model on a broader range of networks and wireless bearers.
- WAP Application Environment – Includes markup languages viewed by "WAP Browsers". WAP 1.* application environment uses WML (Wireless Markup Language) as a conformant XML language. WAP Browser, for the WAP 2.0 Application Environment has evolved to embrace developing standards for Internet browser markup language that permits wireless devices to utilize existing Internet

Ileana Stănescu is with the National Institute for R&D in Informatics, ICI, 8-10 Mareşal Averescu Blvd, 011455 Bucharest, Romania, (e-mail: ileanas@ici.ro).

Razvan Zota is with the Faculty of Cybernetics, Statistics and Economic Information Technology of Bucharest, Romania, (e-mail: zota@ase.ro).

Raluca Stanculescu is with the Faculty of Cybernetics, Statistics and Economic Information Technology of Bucharest, Romania, (e-mail: raluca.stanculescu@gmail.com).

Laura Stanescu is with the Faculty of Cybernetics, Statistics and Economic Information Technology of Bucharest, Romania, (e-mail: arual@yahoo.com).

Emil Stănescu is with the National Institute for R&D in Informatics, ICI,, 8-10 Mareşal Averescu Blvd, 011455 Bucharest, Romania, (e-mail: stanescu@ici.ro).

technologies. XHTML MP (XHTML Mobile Profile) is the markup language of WAP 2.0. XHTML MP is a subset of XHTML, which is the combination of HTML and XML. eXtensible HyperText Markup Language (XHTML) has been developed by the W3C to replace and enhance the currently used HTML language.

- Additional Services and Capabilities –With WAP 2.0, there is a considerable increase in the number of features available to developers, operators and users.

B. Web Service technologies

Web Services are based on XML, classified as an extended language, because it aims to facilitate the common use of data between different systems.

Web services standards provide a façade type interface for the development environment of mobile applications that is based on a server component and a client one. This integration of web service technologies with object oriented programming techniques, and with the structure and communication based on the web raises special development problems for mobile devices such as Smart Phones and PDAs.

III. THE ARCHITECTURE OF THE STUDENT INFORMATION SYSTEM

The architecture of the software system frames into a multi-level architecture type, with the following levels: data administration level, business level, client level.

The information system is formed of two base software

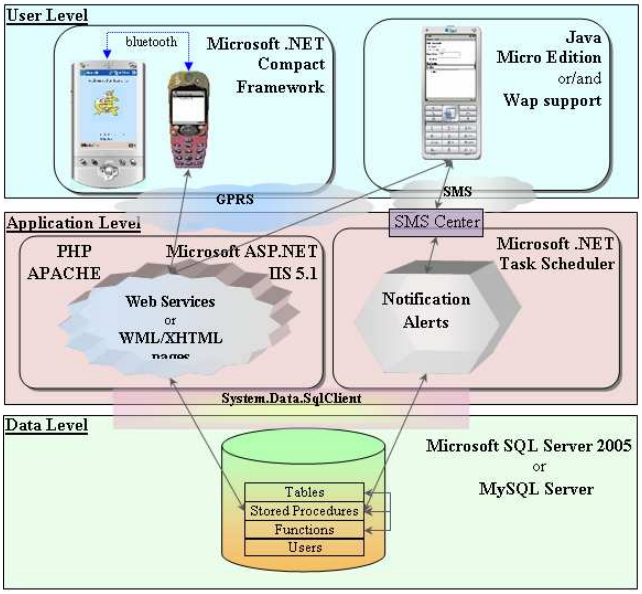


Figure 1 System Information Architecture

applications. One is based on the client level use of some *JavaME* or *Microsoft Compact Framework* modules, corresponding to the equipment type. At the application level, on the server, there are web services that deliver useful content to the user. These services are accesed by the client applications fore-cited.

The second application uses, on the client side, only WAP browser facilities. On the server side the PHP, WML, WMLScript files are developed in correlation with an Apache web server. Another possibility would be that the server side is developed on a Microsoft platform using the IIS Web server and ASP.NET for the new generation of user content. The two applications basically, have the data level formed of shared data bases, developed using Microsoft SQL Server. Nevertheless, for experimentation reasons, the MySQL data server was used for the WAP software application.

The information system also provides transmission support for SMS messages, that permits user notification about certain events or it gives the possibility to navigate on WAP type Internet links to obtain the neccesary information.

Figure 1 presents the schema of the information system architecture.

A. The functions of the system

The main functions of the system are the following:

- Handling information about the users: specific to the profile (the group for students, department, title for professors) or common to all profiles (first name, last name, CNP, etc.);
- Handling information regarding the educational institution: faculties, departments, classes within the departments, groups within the faculties;
- Handling elements of the schedule: week-day, hour and minute to start (begining time), hour and minute to stop (ending time), room, professors, class, group, subgroup, type (course, seminar, lab), scheduling (weekly, even week, odd week);
- Authentication using the mobile phone or PDA;
- Visualisation of the user's profile(s): student profile and/or professor profile;
- Examine the schedule using different criteria;
- Information about the academic situation (grades obtained during the academic year, credit count);
- Sending messages to users.

IV. WEB SERVICES BASED INFORMATION SOFTWARE APPLICATION

The first application example uses Web services and enables users to consult their schedule from mobile devices, on many different criteria. The example describes how the application was created on a simple PC (the application architecture, technologies used in implementation), the software used for its development and the upload to the mobile phone.

The application contains four modules which are assigned on the levels of the architecture in the following way:

- at the data level: a module which is implemented as a data base of the *Microsoft SQL server, 2005 Express Edition* version type.
- at the business level: a module which contains two web services with similar functionalities,

but with different access possibilities. This module is implemented using a Microsoft *Internet Information Server 5.1* web server.

- at the client level: two modules which implement the functions of the client interface for two types of mobile devices: *Pocket-PC 2003* (PK-PC) and mobile phone (MP).

A. The description of the web service

Two Web services have been developed into Microsoft Visual Studio 2005 Web Service project types, using C#. This medium permits great automatization of the defining and development of a Web service, specific to the common infrastructure described by the used XML standards.

Microsoft technologies offer the *DataSet class* for the data interchange, structured corresponding to a relational data base, class that facilitates very much the information handling, once received by the Web service client. The only problem is that the *DataSet* structures (dynamically created by the adaptors that read data from other data sources) are not so easily accessible from client applications developed on other platforms. That is the reason why we developed two web services: one with methods that return only *DataSet* objects and that will be referred only from the application for the *PocketPC*, and another service that returns simple data types, that can be referred from the *PocketPC* application, as well as from the application for the mobile phone developed in *J2ME*.

B. The schedule querying application from the Pocket-PC using a Web service

The module intended for the PK-PC client was developed in the Microsoft .NET Compact Framework 2.0 technology, in the Visual Studio 2005 IDE environment, using C# language. This client application will consume the created web service.

The client application will send a request to the Web service, that will return a result to the application from the mobile device, using a XML/SOAP interface over the HTTP protocol.

The mobile application offers the following functions:

User's authentication after inserting the account and password information. There can be two categories of users – students and professors – and corresponding to the category to which the user pertains, they can have access to certain menu features.

Visualization of the user's profile and utilization of the data corresponding to this profile for viewing the schedule. For example, a user can go to two faculties, and from the form in which is described the profile can choose the group for which can view the schedule.

Visualization of the schedule. The user can choose to visualize the schedule depending on several parameters. Thus, the user can choose one or more parameters between the following:

- the starting point from which begins the visualization of the schedule,
- the faculty at which the user studies/teaches,
- the group no., which belongs to the faculty which was filled out previously,
- the professor's name, if the student wants to find out the schedule of a certain teacher,
- the class name,
- the room no. in which the class takes place.

1) The consumption of the Web service from the Pocket-PC

The application includes two web services as was mentioned above.

After the creation of a web reference, the referred web service is available to the client through a class named *Service* (the name of the class from the ASP.NET project that has implemented the web service) from the namespace *denumire_referintaWeb* (in our case *serviciu_web_ASE* and *serviciu_web_ASEms*).

2) Running the Application

The first step in running the application is the user

Figure 2 User's options

authentication. (See figure 2).

The menu options will be activated corresponding to the user type (student or professor). The user can be student and teacher at the same time, in the case of master post-graduates and doctorate post-graduates that teach in the academic institution.

After authentication, the user can view his profile and consult his schedule using certain parameters.

By choosing the Orar (schedule) option from the menu, a form will open which will contain two *TabPage elements* (with the displayed text *Parametri*, respectivelly *Orar*) of an object of *TabControl* type from the form. When the form is displayed the active page is the one with *Parameters*.

In this form, different parameters can be introduced which will help generate the schedule. If the *Profesor* parameter is introduced with the name (or part of it), then the professor's schedule can be visualised, or professors with the respective name.

3) Presentation of the schedule consulting operation on the PDA

With the page change from *Parametri* to *Orar* (by touching the text *Orar*) a series of events ensue. Two events are treated in the application: the *Validating* event attached to the page with the text *Parametri* and the *SelectedIndexChanged* event attached to the control *tabControl1*. In the procedure that treats the first event the validation of the faculty field is achieved. (Other validations can also be implemented). In the second procedure that treats the second event, a test is performed to see if the activated page is *Orar* (schedule), in which case the request for the schedule is called.

Figure 3 shows a schedule from Monday, with classes that start later than 13:03, for group no. 1071 of the faculty

with the CSIE abbreviation.

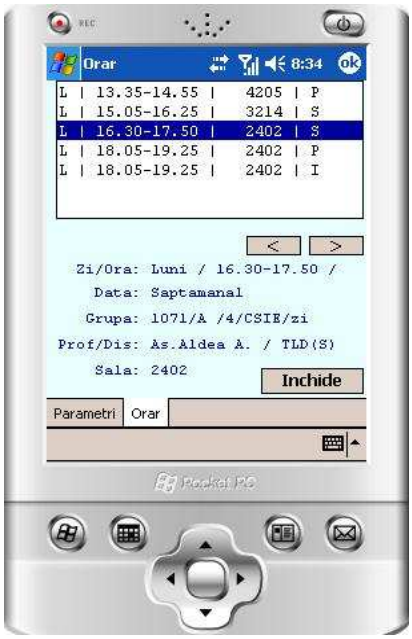


Figure 3 Student's schedule details

The schedule is listed in a list (a *ListBox* type control) with simple selection. At the selection of a list element, the details of the element schedule from the selected position will be displayed in the lower side of the screen. Using the buttons „<" and ">" , one can navigate from a day to the other so that the schedule can be visualised.

C. The application for schedule consulting from a mobile phone using a web service

The *Java 2 Micro Edition* (*J2ME*) web service specification is a standardization attempt of the programmatic access at web services from the client applications *J2ME*.

Developed in the *Java Community Process* under the name of *JSR 172*, the *J2ME Web Services API* (*WSA*) extends *Java 2 Micro Edition* platform in order to support web services.[4]

The *J2ME* web service implementation offers two optional packages based on XML that can be used to develop Java web services for mobile devices, such as cellular phones or PDAs: one for remote call services (*Java API for RPC* based on XML or *JAX-RPC*) and one for XML parsing (*Java API* for XML processing or *JAXP*). The packages are independent one of the other. The APIs interact in order to provide access to the web services through profiles from *Connected Device Configuration* (*CDC*), or from *Connected Limited Device Configurations* (*CLDC*). [5]

The module destined for the *TM* client was developed in *NetBeans 5.5* for the *SUN Java* (*TM*) *Wireless Toolkit* platform 2.5.1, in the *Java* language.

Next, the most important screens of the application are presented:

-in figure 4.a is the authentication screen (web service call, authentication)

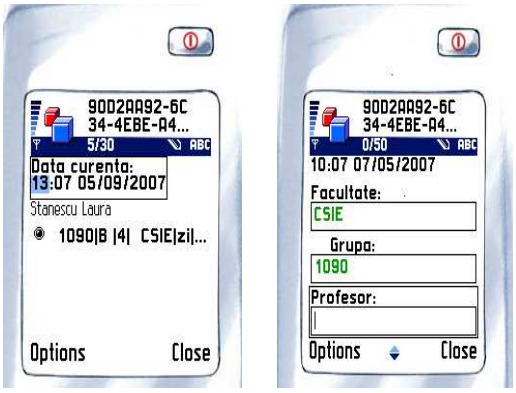
-in figure 4.b is the correct authentication screen (with the user identifier in the title).

a) **b)**

Figure 4 User authentication

-in figure 5.a is the student profile information screen (web service call, the *getStudent* operation)

-in figure 5.b is the schedule interrogation parameter screen (Data (date), Facultate (faculty), Grupa(group) fields – automatically filled).



a) **b)**

Figure 5 Student profile and interrogation

-in figure 6.a) is the selection screen of a schedule element for the visualization of details)

a) **b)**

Figure 6 Schedule details

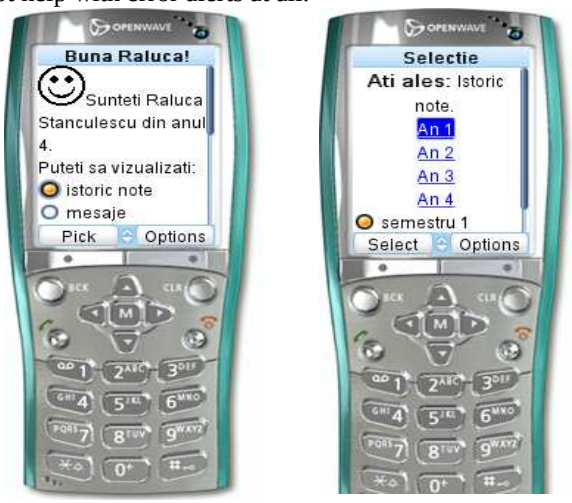
-in figure 6.b) is the Schedule details screen (title, professor's name and the subject are in the abridged form).

V. WAP Based Software Information Application

The second application of information system has the purpose of putting to good use the WAP protocol and providing both students and professors with much needed information. In this section we describe how the application was developed on a PC and how it was set up on a server. A WAP enabled phone is necessary, that implements the WAP 1.* or WAP 2.0 standard.

A. Developing the Application

We followed a client-server architecture supported by standard Internet protocols. This technology allows building dynamic Web pages. They are built on the server from non-static information stored in a database. This information is previously supplied by the system administrator or student/professor input (in the case of messages). The application was developed using an *Apache* server with client-side support for *WML* content and server-side support for *PHP*. Using the *WML* standard we can develop pages that can be dynamically generated with *PHP* and served by *Apache*. A *MySQL* database is used to obtain information about students, professors, subjects, etc. The bundle of files with the *.wml* extension is uploaded onto the server configured to serve *WML* content. The *Opera* browser was used to test the application, being one of the best *WML* renderers. For the purpose of this presentation the *Openware* phone simulator is used. Its phone-like shape creates the best image possible of how the application will look on a mobile phone. Also, *Nokia Mobile Toolkit* was used to write the code, although it does not help with error alerts at all.



a) b)

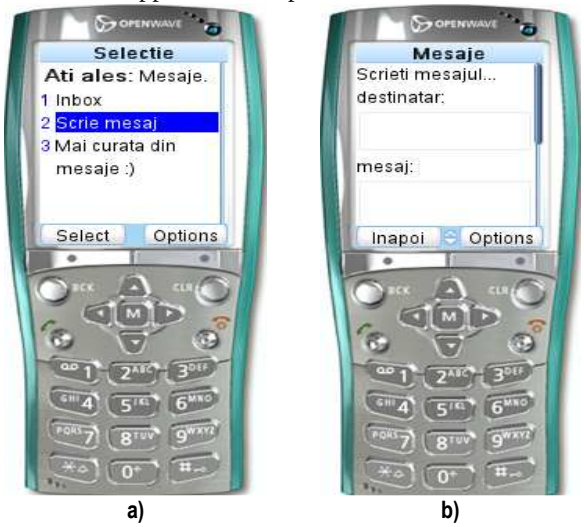
Figure 7 a) Options presented at login /b) Grade options

B. Setting up the application environment

The installation of the *Apache* server is needed on the server (the 2.2.4 version was used) and of the *PHP* module (*PHP* version 5). Also installation of *MySql Server 4.1* is needed. This choice of development software is very popular amongst web developers, and it can be successfully applied to wireless *WML*-programming as well.

C. Running the Application

The application caters for two types of users: a professor and a student. Both of them have different options and possibilities on the application menu. Access is granted after authentication. The accounts are predefined and are outside of the application scope.



a) b)

Figure 8 a) Message options/b) new message

In case of wrong authentication, for whatever reason, the application warns the user of the mistake made and requests the correction of the entered data.

Once the user has accessed his account, the options provided to the student are the following as shown in the figure 7. A similar menu is available to the professor.

The used images presented in the application are *WBMP* pictures, that can be integrated in the *WML* page.

a) b)

Figure 9 a) Search for information/ b) Inbox

The application provides support for a correct authentication. In addition to this, the password can be modified, as a security measure. The message options provide *INBOX*, message writing, message deletion possibilities. Also, the professor user can send "spam" messages to a whole year, series or group of students. This way, the professor can advertise oncoming classes for the next semester or year, for example. A student can check for grades from the current year or the previous ones as

shown in figure 7-b). He, also, can send or receive messages (figures 8 and 9).

Both users can check their schedule, an option that can prove very useful if they are on the move.

VI. Conclusion

The paper presents an information system for the academic environment. Two approaches have been considered for designing, developing and testing of software applications, in concordance with the different features of devices and environments of the mobile user. The first application uses *Web services* to supply the information to Web services consuming mobile clients such as *Pocket PC*s and *mobile phones.* The second software application of the system uses WAP and WML technologies to make a point about the simple implementation of a very useful program that leverages the wireless web.

As a future development of the system we will integrate the mentioned two applications and we will develop a notification/alerts module that will incorporate push technology to initiate notification messages from the server which can send SMS messages to users so that they to connect to the information server.

Further on, the presented applications will be developed and put together into a “Integrated system for the mobile user” that will be tested in the academic environment.

Life as a student would be much simplified if mobile access to information would be provided.

Acknowledgment

This paper was supported by the Romanian Ministry of Education and Research, under the research program CEEX no. 80/10.10.2005 SIUM.

References

[1] Damon HoughLand, Khurram Zafar – Essential WAP for Web Professionals, 2001 Prentice Hall PTR

[2] Wireless Application Protocol WAP 2.0 Technical White Paper www.wapforum.org, January 2002

[3] WAP, http://en.wikipedia.org/wiki/Wireless_Application_Protocol, accessed September 2007

[4] C. Enrique Ortiz, (2004), Introduction to J2ME Web Services, http://developers.sun.com/mobility/apis/articles/wsa/

[5] Jim White, Turn Your J2ME Mobile Devices into Web Service Clients, http://www.devx.com/wireless/Article/28046, 2005.