Mobile Learning and Testing with Java-enabled Phones

Rosen S. Ivanov[1], Mihail B. Momchedjikov[2]

Abstract – **The widespread use of mobile phones among students suppose possibility to develop mobile learning tools. The paper presents an approach to integrating mobile phones into traditional learning and testing process, which combines the advantages and benefits of both. The main problems, corresponding to mobile testing tools for Java-enabled phones development and methods for their solution is described.**

Keywords – **Mobile learning, M-learning, Java midlets.**

I. INTRODUCTION

The term "mobile learning" (m-learning) refers to the use of mobile and handheld devices (mobile phones, smartphones and PDAs) in learning process. Mobile phones are much more reasonably priced than desktops, and therefore represent a less expensive method of accessing the Internet services like the m-learning portals.

Mobile phones are becoming very popular within the communities of university students. Nearly 100% of students have a mobile phone [1] (Italy - 94%, Finland and Sweden - 92% , UK, France and Spain - 82%). The widespread use of mobile phones among students supposes possibility to use this technology to improve the motivation of and communication with students.

Mobile learning is an additional channel to communicate with students and to deliver learning and testing content. The main advantages and benefits of the m-learning are [2]:

- can be used anywhere and anytime;
- can be used of these students who do not take all the courses for the academic year;
- increases motivation to self-learning;
- can be used of peoples with disabilities.

When the m-learning tools are developed the focus will be pedagogical - technologies, standards and specifications in mobile communications used to the creation of the mobile learning tools will contribute to successful learning for the students. The rules for the creation of successful learning tools for distance education are well known by now and must be applied to the new mobile learning tools. E-learning technology is the state of art for distance learning in Europe [5-8]. M-learning is the next generation of learning.

[1] R. Ivanov is with the Department of Computer Systems and Technologies, TU of Gabrovo, BULGARIA, E-mail: rs-soft@ieee.org.

[2] M. Monchedgikov is with the Department of Padio-engineering, TU of Sofia, BULGARIA.

In traditional learning scenarios, for example blackboard or a projector is used for the presentation of the learning material. This scenarios offer many advantages - if there are questions or unclear points each student can immediately ask the teacher and receive an answer. Vice versa, a teacher can test knowledge by asking a questions. Using up-to-date mobile phones with integrated Java Virtual Machine (JVM), high speed mobile data communication (including streaming) and multimedia processing (audio, video) capabilities, mentioned above learning scenarios can be simulated and enhanced [3].

To use mobile phones as a learning tools they need to support application-development technologies such as:

- Wireless Application Protocol (WAP) which allows data communication between WAP-enabled mobile phones and Internet services;
- Java 2 Micro Edition (J2ME) which is a general platform for programming embedded devices;
- Microsoft .NET framework which includes C# language as an alternative to Java;
- NTT DoCoMo i-mode.

Using WAP we can develop Wireless Markup Language (WML) applications. WML is a tag-based document language. The main disadvantages of the WML-based applications are: limited memory size of WAP browser, limited type of objects that can be displayed (text and images), cannot integrate programming logic in WML code, increased number of request to application server etc.

The real mobile applications are written in C++, C# or Java. The mobile device could download Over-The-Air and run such applications. The advantages of J2ME applications (midlets) are as follows:

- lower price of Java-enabled mobile phones in comparison with PDAs and smartphones with operating systems like Symbian and Microsoft Mobile 2003;
- Java applications are platform-independent;
- allows development of the fully Java-based client-server applications;
- Java security model provides a very restricted environment in which to run untrusted midlets;
- Java Development Kit 1.2 allows development of the signed midlets.

II. M-LEARNING SYSTEM ARCHITECTURE

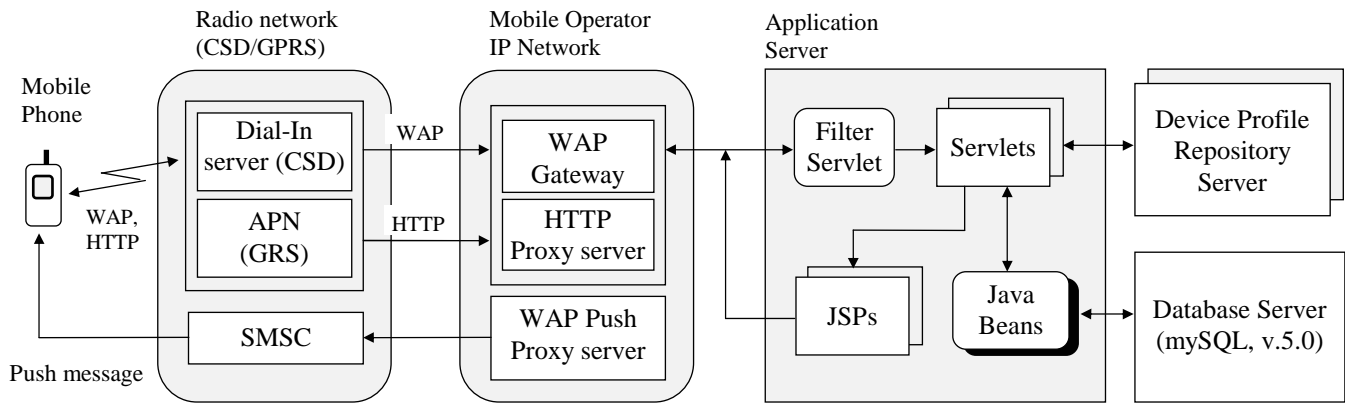We use fully Java-based client-server communication model architecture shown in Fig.1.

Fig. 1. M-learning system architecture

Three layer architecture is used:
- Layer 1 - Client;
- Layer 2 - Application server;
- Layer 3 - Database server.

The mobile terminal (client) should support WAP and J2ME technologies. Tomcat Servlet Engine is used as a WEB application server. Tomcat is an open source project available from the Apache Software Foundation [9] that supports Servlets v.2.2, Java Server Pages (JSP) v.1.1 and Java Beans technologies. The open source database mySQL [10] is deployed as the server database.

The client-server communication flow is as follows:
- A client asking for particular service submits a request (WAP if WML browser generates request or HTTP if Java midlet generates request) to the application server;
- The client's request is retransmitted to the WAP Gateway or HTTP Proxy Server via Dial-In server if CSD bearer (9.6kbps data rate) is used or Access Point Name (APN) if GSPS bearer (52kbps data rate) is used;
- The WAP Gateway transforms WAP request into an HTTP request and sends it to the Application Server;
- The application server replies to the request by returning an answer: WML code if request is from WML browser and eXtensible Markup Language (XMS) if request is from m-learning Java tool;
- The answer is transmitted to the WAP Gateway or HTTP proxy server;
- WAP Gateway or HTTP Proxy sends back the content of answer to the client.

A. *M-learning midlet downloading*

M-learning midlet can be downloaded and stored permanently into Flash File System (FFS) of the mobile terminal if following conditions are fulfilled:
- mobile phone supports J2ME technology, Mobile Information Device Profile (MIDP) v.1.0 or 2.0;
- the client was registered to the m-learning system as an user;
- Java Application Management System (ASM) could be

able to download through OTA application with size of 45KB;
- The memory for Record management System (RMS) must be at least 25KB.

The registration to the m-learning service is realized from an HTML form. The user must enter your name, family and personal number. If the registration is successful the user receives an identification number (user ID) that holds information for tests which user could download.

When application server receives a blind request from mobile terminal it returns a WML deck with registration form as shown in Fig. 2.
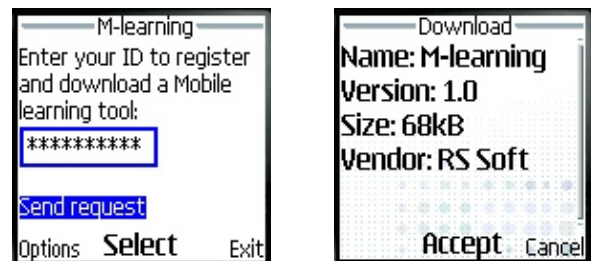

Fig. 2. Registration to the m-learning service

The application server checks entered user identification number (user ID) and prepare an answer:
- Error message if user ID is not valid, request was sent from mobile phone emulator, mobile phone cannot support Java technology or cannot download Java archive with required size;
- Java Archive Descriptor (JAD) file.

The content of JAD file is generated dynamically, because different versions of m-learning application are used:
- MIDP 1.0 version with application-specific XML parser;
- MIDP 1.0 version with Enhydra's XML-RPC package;
- MIDP 2.0 version with Enhydra's XML-RPC package;

When WAP browser receives JAD file Java AMS is started. Java AMS tries to download content of Java archive file (JAR) into download buffer in SRAM. After successful downloading

application is stored into the FFS. The URL to JAR file is described in JAD file as shown in Fig. 3.

```
HTTP/1.1 200 OK
Content-Type: text/vnd.sun.j2me.app-descriptor
Content-Length: 394
Date: Fri, 25 Feb 2005 06:53:05 GMT
Server: Apache Coyote/1.0

MIDlet-1: M-learning, ,mlearning.Mlearning
MIDlet-Jar-Size:45326
MIDlet-Jar-URL:
http://rs.tugab.bg/mlearning/jars/MIDP1A.jar
...
User-ID: GH56KLN809
Mobile-ID: SGHC100G
```

Fig. 3. Content of JAD file (Samsung C100)

B. Mobile phone profile recognition

Mobile phones have different range of input and output capabilities, network connectivity, Java MIPD and Connected Limited Device Configuration (CLDC) versions etc. The User Agent Profile (UAProf) extends WAP 2.0 to enable Application server to recognize hardware and software properties of the mobile terminal. UAProf is an XML format document which is published on a Public Repository Server (see Fig. 1). If mobile phone supports UAProf specification the field "Profile" in HTTP request header holds URI to this document. In opposite application server analyzes field "Accept" to determine a part of phone properties.

We use Apache XSTL (eXtensible Stylesheet Language Transformations) processor to parse UAProf XML document to plain ASCII text. The required information (Phone Vendor and Model, Number of soft keys, Screen size, Bits per pixel, Java platform, Java packages, Browser version, Security support, WAP version, HTML version, XHTML version, Cc/pp accept, Maximum WML deck size, Push accept, Push message size, Player version, RTSP method capable etc.) is stored in UAProf database.

B. Client-server message format

Every MIDP compatible device must support HTTP. For maximum compatibility HTTP should be used as the transport protocol between client and server. J2ME platform does not offer standardized mechanisms for message format. In proposed application an XML-based message format is implemented. XML is used as an industry standard format. XML allows realizing any data exchange between the client and server. The main advantage of XML is that it is standardized and thus highly portable.

The MIDP environment does not provide integrated support for XML parsing and generating. We use two versions of XML message formatting:

- XML-RPC package from Enhydra [11];
- Application-specific XML parser and generator.

XML-RPC is an extremely lightweight messaging protocol enabling the execution of remote procedure over the network through HTTP. The client places an XML message through HTTP POST method that the application server parses. This results in a local procedure that returns the response also in the form of the XML message. XML-RPC enables data types to be specified and transmitted as parameters in platform-neutral approach. At the server side we use XML-RPC support from Apache Software Foundation [12], which can be applied together with the Tomcat Servlet Engine.

One of disadvantages of XML-based packages is their size. XML-RPC package from Enhydra occupies 24KB of program memory. It is important to take this into account when developing applications for mobile phones, since some of them limit size of downloaded Java application. In these cases we use application-specific XML parser and generator that occupy only 12KB of memory. In this case, at the server side Document Object Model (DOM) parser is used.

The content of XML message depends on type of request or answer. For example, when a new test must be downloaded following parameter are sent to server: User-ID; Mobile-ID; Password; Language; Subject; Test type (self test or exam); difficulty level and Checksum. The content of the generated test contains following parameters: Test-ID; Number of questions; Type, Time and Body for each question and Checksum.

Four types of questions are supported:
- type 1 - one-choice question;
- type 2 - multi-choice question;
- type 3 - missing word(s);
- type 4 - error correction.

The "Body" parameter contains question's data and tags that control visualization of the information. M-learning midlet micro-browser analyzes tags and displays information according to tag's names:
- <t> Text </t>;
- <a> Text - answer;
- <m> Text </m> or <m/> - missing word(s);
- <i> URI </i> - URI to image file;
- <A> URI </A> - URI to audio file;
- <v> URI </v> - URI to video file.

III. EXPERIMENTS

When midlet is started the main menu of application has been displayed as shown in Fig. 4.

Fig. 4. Menu of the application (Nokia 6230)

The combination of low-level MIDP User Interface (UI) Application Programming Interface (API) and high-level UI API is used to control information drawn on the screen. The high-level UI API is portable across all MIDP-enabled phones and low-level UI API gives complete access to the screen and

the input events. The main menu includes following items:

- Tests - allows new test downloading, start, continue or delete the test;
- Statistics - information for student's grades;
- Calendar - includes information such as the dates of exams, calendar of classes and news;
- Config - allows setting of the password, language, difficulty level and time interval in seconds after that next question is displayed;
- Teachers - allows searching for selected teacher and sending a message (E-mail or SMS) to him;
- Help - includes useful information for test downloading and navigation in menu.

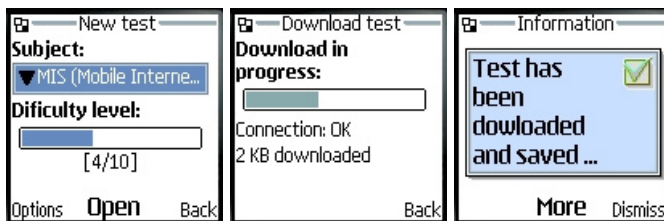The process of the new test downloading is shown in Fig. 5.

Fig. 5. New test downloading (Nokia 6230)

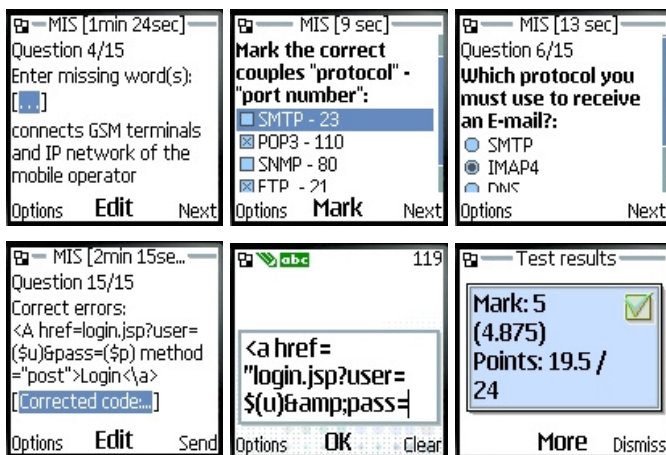Different types of questions are shown in Fig. 6.

Fig. 6. Types of questions and final result (Nokia 6230)
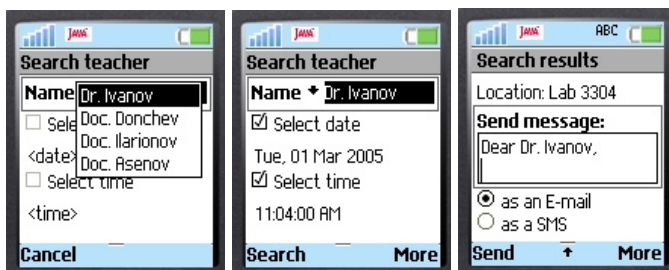
In Fig. 7 teacher searching procedure is shown.

Fig. 7. Teacher searching (Sony Ericsson Z500)

IV. CONCLUSION

The challenge for universities now is to develop didactic tools for mobile devices and to integrate them into their learning strategies. For this reason it is important to develop mobile learning tools which complement the traditional process of learning. Despite the limited possibilities of Java-enabled mobile phones, it is possible to design learning tools like interactive tests.

Mobile learning allows students freely to decide on when to study during a day or a week. The WAP 2.0 and MIDP 2.0 Push technologies (see Fig. 1) could be a way of giving important information to the students (via SMS Center), such as their grades, responses to their assignments etc.

The small size, weight and multimedia capabilities of the mobile phones make them particularly effective for users with disabilities: customize screen colors, text size and fonts; generate vibrating or visual alerts; realize multi-modal interface. Text-To-Speech (TTS) technology can assist users who have visual impairments and find it difficult to read the screen. Voice-To-Command technology can aids users who may find it difficult to type or use the keyboard.

REFERENCES

[1] J. Attewel, C. Savill-Smith, "Mobile learning and social inclusion: focusing on Learners and Learning", A book of papers "Learning with mobile devices: Research and Development", pp.3-11, ISBN 1-85338-833-5, 2004.

[2] M. Perry at al., "Dealing with Mobility: Understanding Access Anytime, Anywhere", ACM Transactions on Computer-Human Interaction, Vol.8, No.4, pp.323-347, Dec. 2001.

[3] M. Mayorga-Toledano, A. Fernandez-Morales, "Learning tools for Java0-enabled phones: an application for actual studies", A book of papers "Learning with mobile devices: Research and Development", pp.95-98, ISBN 1-85338-833-5, 2004.

[4] M-learning project, http://www.m-learning.org

[5] Project "Leonardo Da Vinci", http://europa.eu.int/comm/education/programmes/leonardo/leonardo_en.html

[6] Project "ISOPIA's m-Learning", http://www.isopia.com/main.html

[7] Project m-learning ResultsMART, http://www.learningcitizen.net/mlearning/mlearning.shtml

[8] Project "Aictc's m-Learning", http://www.aictc.com/moblieComputing/mLearning.htm

[9] Web site of the Jakarta Tomcat project, http://jakarta.apache.org/tomcat/index.html

[10] Web site of the MySQL open source database project, http://www.mysql.com

[11] Web site of the Enhydra KXML-RPC project, http://kxmlrpc.enhydra.org

[12] Web site of the XML-RPC project of the Apache Software Foundation, http://ws.apache.org/xmlrpc