

# Mobile Performance Support System – Technical Solution

M.Milev<sup>1</sup>, D.Tokmakov<sup>1</sup>

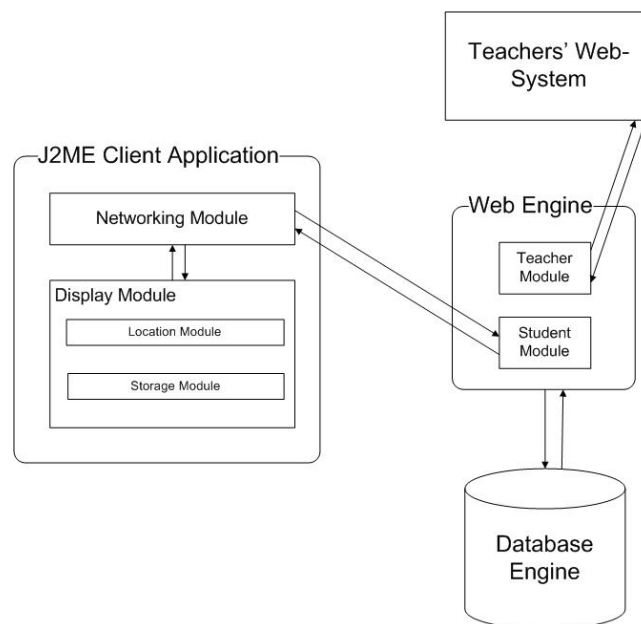
<sup>1</sup> ECIT Department, Plovdiv University „Paisii Hilendarski“, Plovdiv, Bulgaria

Students, trainees and trainers, we are all becoming more mobile - over half of all students now spend up to half of their time in mobility. Mobile devices are always available and can be used for a variety of learning functionality - providing access to content (both informational and instructional) and for communication and collaboration purposes. They can be used for formal or informal learning purposes as well as for performance support, i.e. for delivering information and support just-in-time and in context. That is why our organization decided to develop the Mobile Performance Support System.

**Keywords** mpss; ipss\_ee; learning; objects; educational; content; mobile; organization; java; j2me; students

## 1. Structure of the Mobile Performance Support System (mPSS)

The mPSS consists of three main, dependable from each other modules: a J2ME client application, which must be downloaded and installed to the client's (student's) phone; Database Engine – such as MySQL, for storing the educational data; a Web-Engine - such as Apache, for managing the data.



**Fig. 3** *mPSS modules*. Describes the main connections between the modules of the Mobile Performance Support System.

### 1.1 J2ME Client Application

J2ME client application is in form of a downloadable and installable JAR (Java ARchive) file. It could be downloaded from the main site of mPSS and installed to any mobile device, which supports J2ME CLDC 1.1 and MIDP 2.0 (or higher) standards. These two standards assure, that the phone has some basic and common features needed for the mPSS to correctly display the educational data (such as the J2ME MMAPI).

The J2ME client application can be divided in four modules:

#### 1.1.1 Location Module

The Location Module uses the main J2ME positioning libraries to obtain the device's coordinates (if the device has an integrated GPS module and/or has a valid connection with one). The specially defined criteria for this process may force the device to use other methods for locating, such as the

GSM/UMTS network. Of course this will only work if the corresponding mobile operator has allowed its BTS to transmit such data.

#### 1.1.2 Networking Module

The Networking Module can use every possible Internet connection for delivering the educational data – 2G, 2.5G and 3G, also old WAP services and or modern Wi-Fi/WiMAX services, of course only if the phone supports them. These services determines the type of the device the client is using and thus determines what type of content the student will receive upon studying. It is inappropriate to deliver rich multimedia content (such as video files and/or large voice files) to a phone with a low internet bandwidth and in the mean time isolate the modern devices from viewing such “interesting” educational data.

#### 1.1.3 Display Module

The Display Module is actually the main J2ME application, which encapsulates all the other modules to build the client software. It offers the user a simple way for navigation and cares for the right displaying of the content. Upon loading of this module it checks what kind of multimedia formats the device supports, thus determining what kind of content the device can display to the learner. This module and the Networking Module are the two main criteria for determining the phone. The two possibilities are: “Advanced” (modern) device or “Unadvanced” (elder) phone. These two categories determine, what kind of content will be transferred and shown to the client.

#### 1.1.4 Storage Module

The Storage Module is a simple module based on the Record Management System (RMS) of J2ME, which stores important user data for further usage. The coordinates, the network bandwidth, usernames and passwords are such important data and in order to speed up loading time, save some time and/or network costs from the user, this data must be stored locally on the device.

### 1.2 Database Engine

The mPSS system uses the most famous database engine – MySQL. Its ease of use, high level of security and no costs has made this engine to one of the best “data stations” in the world. Everything in the mPSS system is stored in the database – from the organization of the system, through the educational data and until the multimedia files. The whole architecture of the educational system is represented as tables in the database engine, which ensures the right organization of the system. Tasks, Modules and Courses are stored in separate tables, which contain only the most important information about them. The multimedia files, which enhance the educational data, are stored in a separate database table, thus allowing fast and easy access to them for every user of the system. Other tables are used for determining country and special for the given coordinates tasks.

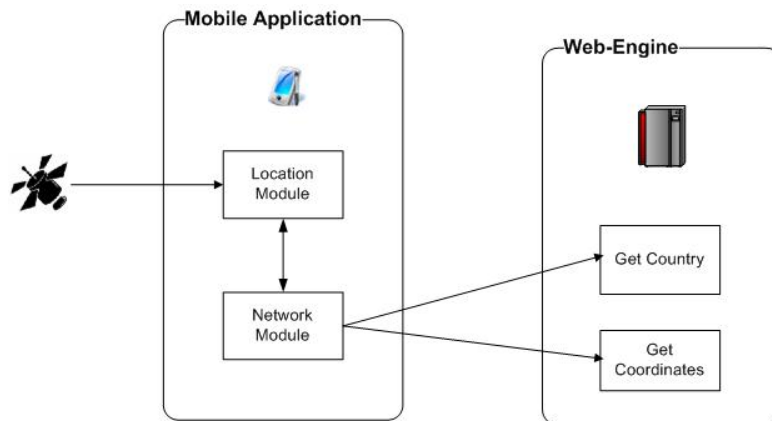
### 1.3 Web-Engine

The third module of the system is the Web-Engine. Our system uses the free Apache web server, which plays the role of “glue” for the whole system. Teachers use a specially developed web system for importing and editing educational content to the database system. Mobile clients, such as the students, download and install a specially developed J2ME application to their phones, which using its Networking Module uses the Web-Engine to retrieve educational data and/or determine tasks only for the given country and coordinates. The Web-Engine-side of the mPSS system is developed in PHP, so it allows using other web servers, which support this scripting language. Its easy way for developing software allows fast upgrades of the solutions, features and services, and easy maintenance of the current system.

## 2. The positioning system

Upon loading the J2ME client application for the first time it invokes a process, which tries to determine the device coordinates using the Global Positioning System (GPS). This process starts with a procedure, which tries to load the positioning libraries of J2ME. Because not all mobile devices support these, in case they are missing the process terminates immediately without any warning or error and also disables all features, which are related to positioning. There are no warnings or errors, since the application can continue its normal work also without

these features. After this initialization the GPS device of the phone is woken up and a request is sent to determine the coordinates. For one minute the application process sleeps. This one minute is needed for the device to startup and take the coordinates. This long period should provide enough time even if the device is next to high buildings or in areas, where the GPS signal is not so strong. Coordinates cannot be determined inside of buildings or other closed areas. If after a minute the position of the device is not recognized all features of the application, which require positioning will be turned off, again without any warnings. If the GPS receiver of the device has successfully determined the coordinates, they are stored in the application memory for later use, using the Storage Module.



**Fig. 2** Workflow of the positioning system. This figure describes the main workflow of the positioning system in the J2ME mPSS application. Location Module receives data from the GPS satellites, sends it to the network module, which may use it to filter available tasks according to country and/or device location.

If there is already a determined position in the memory, there are two features of mPSS, which can use it. The first one hides all tasks, which aren't determined for these coordinates, or said another way – shows only tasks which are configured by the teacher for them. For example if a student needs to study a task about some building, and the teacher has properly configured the task for the coordinates of this building, the student may go next to it, and use the described feature to “isolate” only this task. This is a great feature for developing location based tasks. After the learner selects this feature from the task menu in the J2ME application, it connects to the Web-Engine using HTTP and passes it its coordinates. In the Web-Engine there is a module called “Get Coordinates”, whose only function is to search the whole database and “isolate” the tasks for the given coordinates. The new results are again passed to the client application using HTTP.

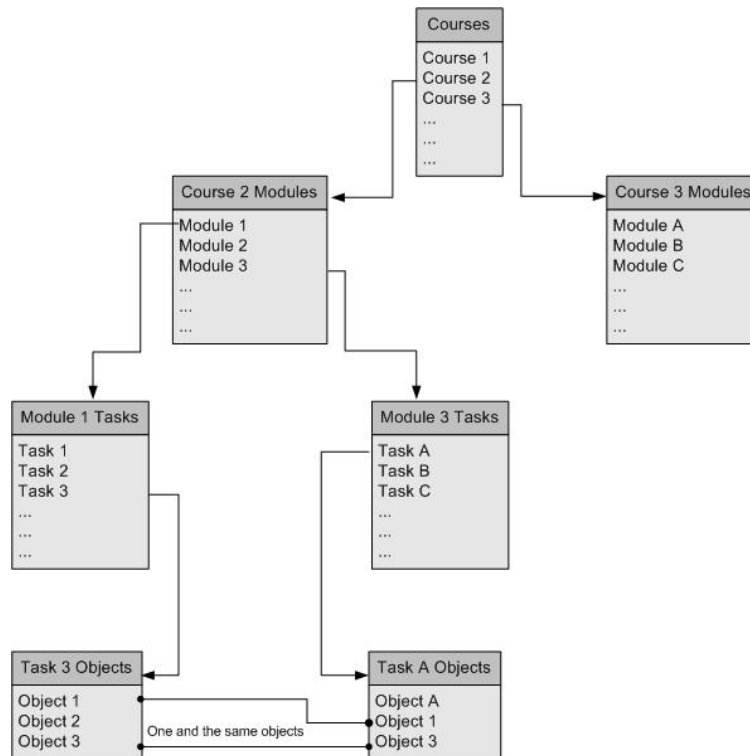
The second feature of the client application, which uses the GPS coordinates shows only tasks which are specified for the given country. Or other way said, the student may hide all tasks, which are in other languages (of course if properly configured by the teachers). Again after selecting this feature the client application passes its coordinates to the Web-Engine, which this time executes its “Get Country” module. It searches the database for countries which cover this position and after that shows only tasks, which are configured for it. For example a person finds himself in Spain and he wants to see only courses in Spanish. By selecting this second feature of the application he will get only list with all tasks specified for Spain. Of course if there are some tasks in English or German for example, but again configured by the teachers for Spain, they will also be shown.

### 3. Structure

#### 3.1 Educational Content

The mPSS system has the standard IPSS-EE structure: courses, containing modules, containing tasks. In this new system a new level was integrated in this structure, called “Objects”. Objects are the building elements of the tasks. Objects can be: text, picture, sound or video. One task may have as many objects as the teacher want. One object can be used not only in one task, but in as many as there are in the system, not only in the same module or course, but also in others and by other teachers. This brings the greatest power of the system – the sharing of objects. For example if one tutor has created a Module for “Electronic Design” and another wants to develop a task for “Operational Amplifiers”, he may use objects from the module. This has two positive sides: if the student has already made a task from the module, he will find the same objects in the new task and thus it will be easier for him to study them; the sharing of objects will save time and resources of the tutor, and saves

also the time of the student, because the system is not overloaded with one and the same information, so it needs less time to search and use its database. For security reasons objects cannot be edited or deleted while they are assigned to a task.



**Fig. 3** Structure of educational content. Courses contain tasks, containing modules, containing tasks. Tasks are built from objects (text, pictures, sound and video) and objects can be shared throughout tasks.

### 3.2 User Structure

In mPSS there are two roles of persons – students and teachers. Teachers use a specially developed Web-System, which directly communicates with the Web-Engine, to upload educational content to the system. Students are actually the direct user of educational content through their mobile devices and thus the system is specially developed for them. In order to use the system, they must download and install a JAR file to their phones. Students are identified with unique e-mail and password. Once logged in, they may view all tasks, which they have chosen to study or use the special positioning features to show only specific tasks.

## 4. IPSS-EE Learning Objects

IPSS\_EE Learning Objects fits Wiley’s definition, “any digital resource that can be reused to support learning”. IPSS\_EE\_LOs are certainly designed to support learning. More specifically, they support learning in an performance-centered environment by providing necessary materials in an easy-to-access format. IPSS\_EE\_LOs are reusable. Most immediately, they will be reused project after project for the duration of each MBA cohort. Designers use subsets of the objects in other IPSS\_EE, the students use the learning object library to create own courses. IPSS\_EE\_LOs are digital. In fact, because of the distributed nature of the program and our desire for easy maintenance, our learning objects are exclusively Web-based. IPSS\_EE\_LO provides a combination of the following capabilities: reference information about a task or closely related set of tasks; task-specific training; expert advice about a task; automated tools for task performance.

Content structure answers to the performance support systems requirements – the content is structured to provide individualized online access to the full range of information, guidance, advice, data, images, tools and software to permit the user to perform a task with a minimum of support and intervention by others. IPSS\_EE\_LO consists:

- reference information about a task or closely related set of tasks;
- task-specific training;

- expert advice about a task;
- instructions how to perform the task
- instructions how to use the software.

**Acknowledgements** The support by CONTSSENS CONTSSENS (Agreement number 2007-1968) is gratefully acknowledged.

## References

1. Brown, J., Collins, A. & Duguid, P. (2002). *Situated cognition and the culture of learning*. Retrieve February, 2 2002 from: <http://www.ilt.columbia.edu/ilt/papers/JohnBrown.html>
2. Copley, J. (2007). Audio and video podcasts of lectures for camous-based students: production and evaluation of student use. *Innovation in Education and Teaching International*, 44 (4), 387-399.
3. Gaskel, A. (2007). Open and distance learning on the move: mobile learning, crossing borders, rethinking roles (Editorial). *Open Learning*. V. 22 (3), 197-200.
4. Gery G. (2002) Performance support – driving change. In *The ASTD e-learning handbook. Best practices, strategies, and case studies for an emerging field* (Rossett, A., ed.), pp. 24-37, McGraw-Hill, New York.
5. Greenberg J. & Dickelman G. (2002) Distributed cognition: a foundation for performance support. In *The ASTD e-learning handbook. Best practices, strategies, and case studies for an emerging field* ( Rossett, A., ed.), pp. 303-313, McGraw-Hill, New York.
6. Kirton, M. (2003) ‘Adaption – Innovation in the context of diversity and change’. London:
7. Lee, M. & Chan, A. (2007). Pervasive, lifestyle-integrated mobile learning for distance learners: an analysis and unexpected results from a podcasting study. *Open Learning*, 22, (3), 201-218.
8. Mayer, R. (2005). Introduction to multimedia learning. In In R. Mayer (Ed.), *The Cambridge handbook of multimedia learning*, pp 1-16. New York: Cambridge University Press.