

mDIPSEIL v2

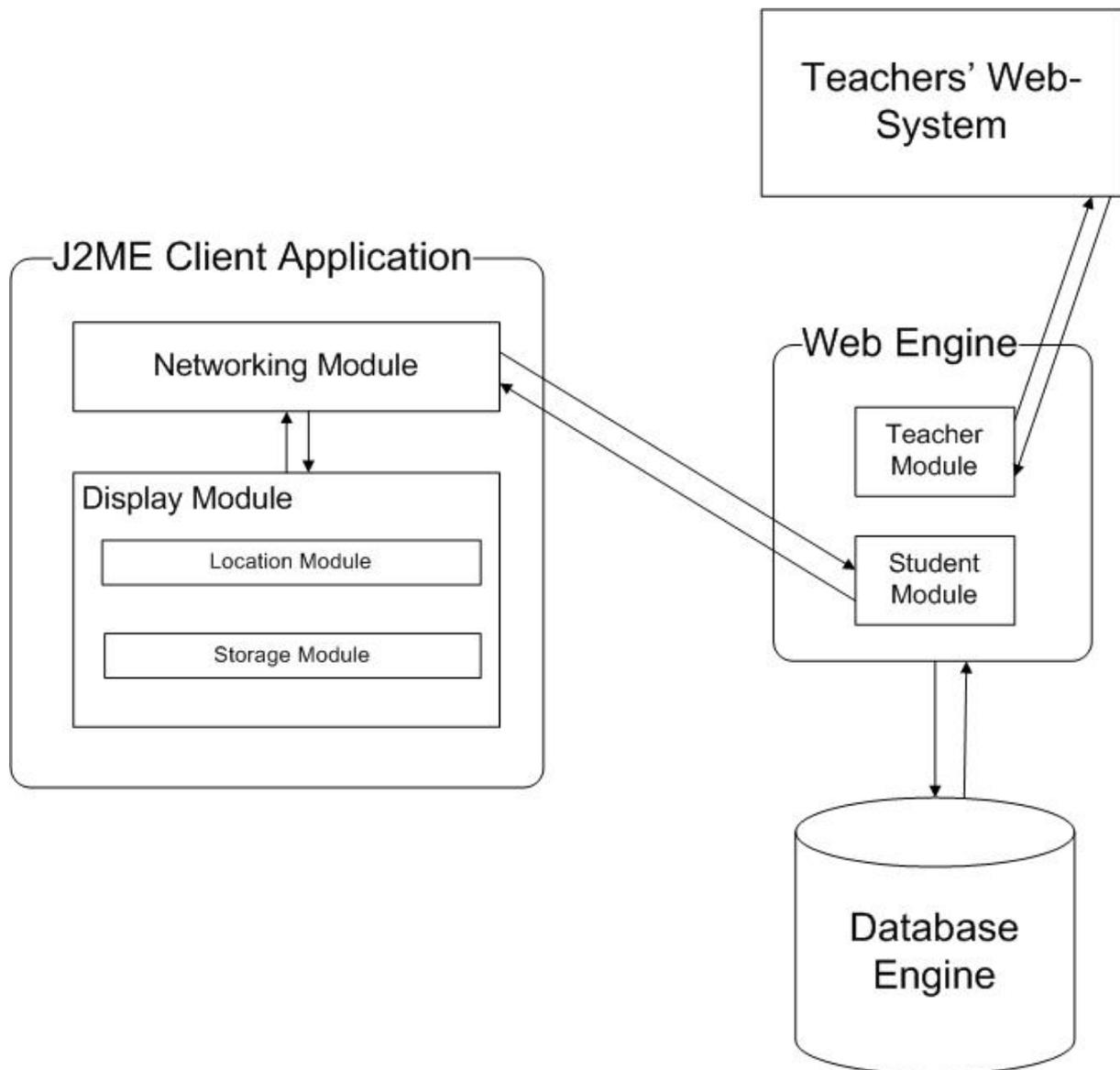
Local System Design

Table of Contents:

- Main Modules of the mDIPSEIL v2 system
- Determine the Device Position using GPS
- Usage of the GPS Coordinates in the System
- Structure of the Educational Content
- User Roles
- Integration with DIPSEILv2
- Developing Educational Content by Teachers
- Viewing and Studying Educational Content by Students
- SCORM
- IPSS_EE Learning Objects

Main Modules of the mDIPSEIL v2 system

The mDIPSEIL v2 Mobile Learning Management System (MLMS) 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.



The 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 mDIPSEIL v2 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 mDIPSEIL v2 system to correctly display the educational data (such as the J2ME MMAPI).

The J2ME client application can be divided in three four modules:

1. **Location Module** – its task is to locate the client's mobile device using the Global Positioning System (GPS);
2. **Networking Module** – its task is to manage the downloading of educational data;
3. **Display Module** – this module should properly display the educational data to the student and provide him with tools for easy navigation through the different task parts;
4. **Storage Module** – this module stores important user data on the phone (using the J2ME RMS technology) for later use.

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.

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.

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.

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.

The two most important modules – the Display Module and the Networking Module, cannot run without each other. The other two modules are optional and may only enhance the interaction with the educational content. For example the client application needs constantly to download this educational content from our servers (so it constantly needs networking). If the positioning libraries of J2ME are not supported by the given device, then the Location Module won't work, but this won't stop the application from working. The student won't just be able to use some extended features of the tasks (such as showing only tasks corresponding to the given coordinates). This modularized technology allows our application to cover much more devices – from elder ones (such as the SonyEricsson K750i) to most modern ones (such as the newest Nokia N-Series or HTC PDAs).

Database Engine

The mDIPSEIL v2 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 mDIPSEIL v2 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.

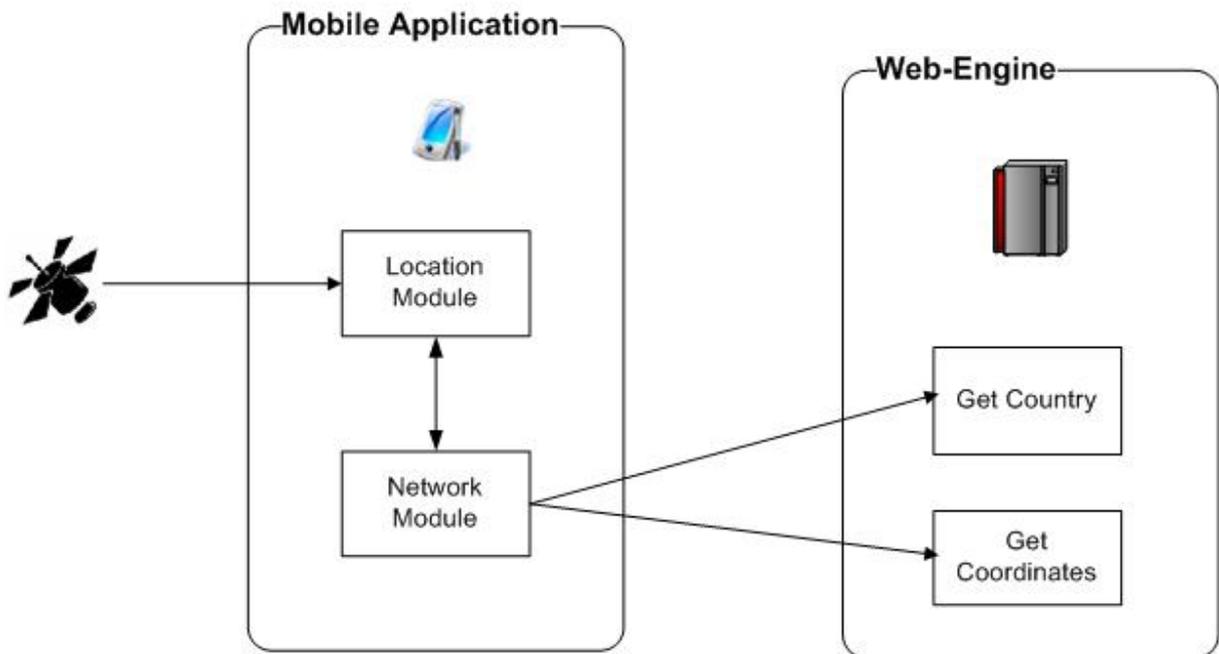
Web-Engine

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 mDIPSEIL v2 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.

Determine the Device Position using GPS

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.



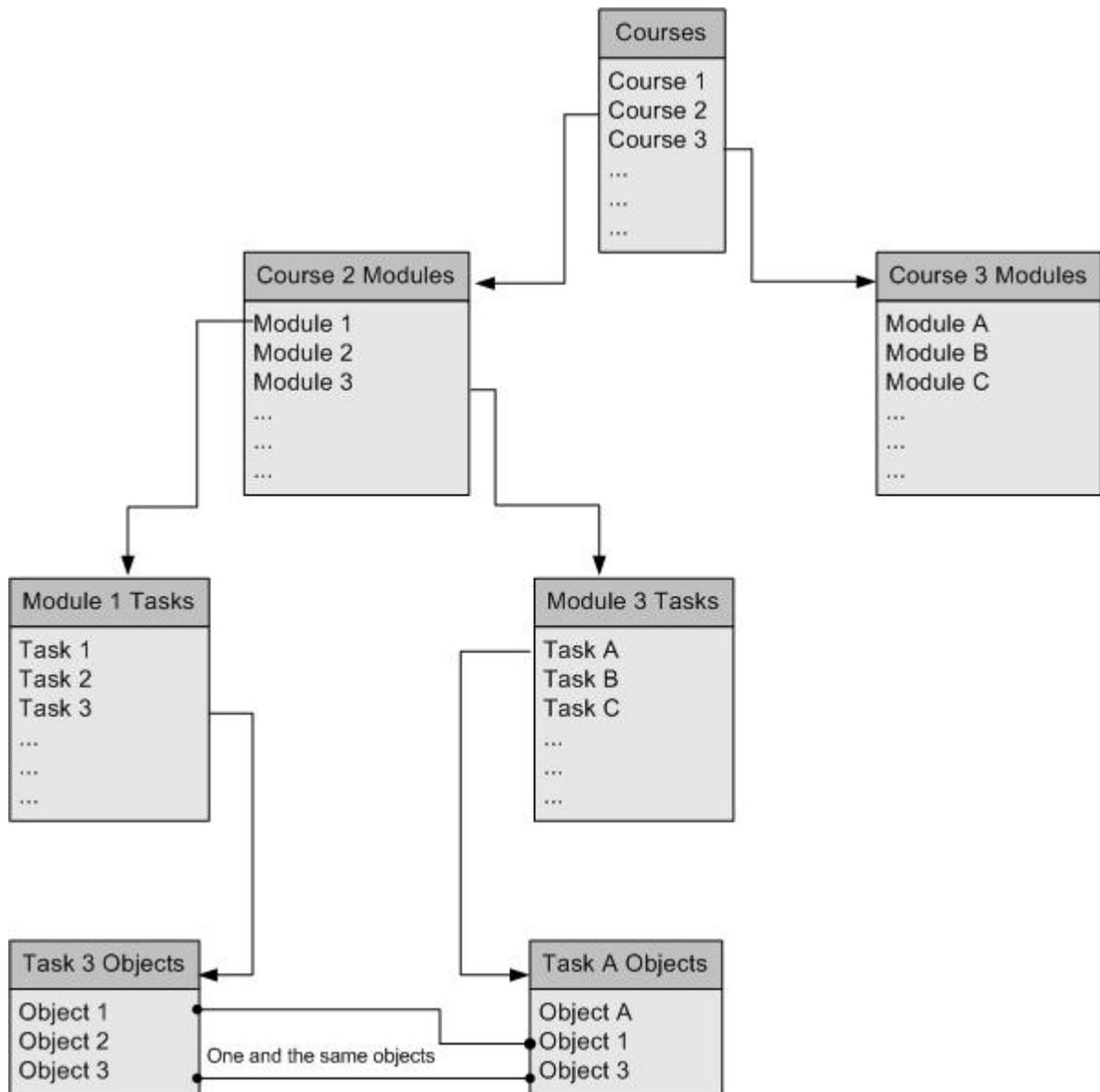
Usage of the GPS Coordinates in the System

If there is already a determined position in the memory, there are two features of mDIPSEIL v2, 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.

Structure of the Educational Content



As DIPSEILv2 the mDIPSEIL v2 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.

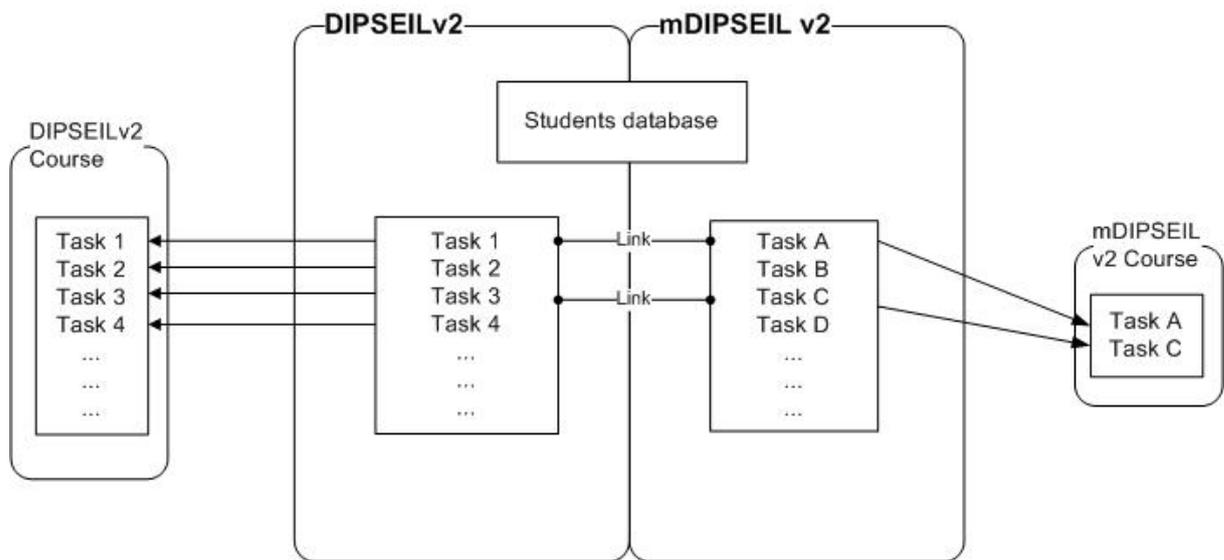
User Roles

In mDIPSEIL v2 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. Using this Web-System they may create and/or delete courses, modules, tasks and objects, assign objects to tasks or remove assignments, but only for tasks created by them. They could also specify, how objects are used inside of tasks. For proper view of educational content teachers must also specify coordinates for the tasks and/or country for the task. Tasks in mDIPSEIL v2 must be attached to tasks from DIPSEILv2, or else students won't be able to view them.

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 from the original DIPSEIL v2 system. Once logged in, they may view all tasks, which they have chosen or use the special positioning features to show only specific tasks. If for example a student wants to see courses made in Germany and he is somewhere in this country, he may choose the option – “Hide tasks from other countries”, which will automatically activate the “Get Country” module of the Web-Engine. Students are presented with the educational content using the standard IPSS-EE Learning objects – the simply organized menu in the task screen allows them to browse through “Task Description”, “Task-specific Training” and so on. According to the specification of their phone and its features they may view advanced content (such as video or sound), or else they will only see text and images.

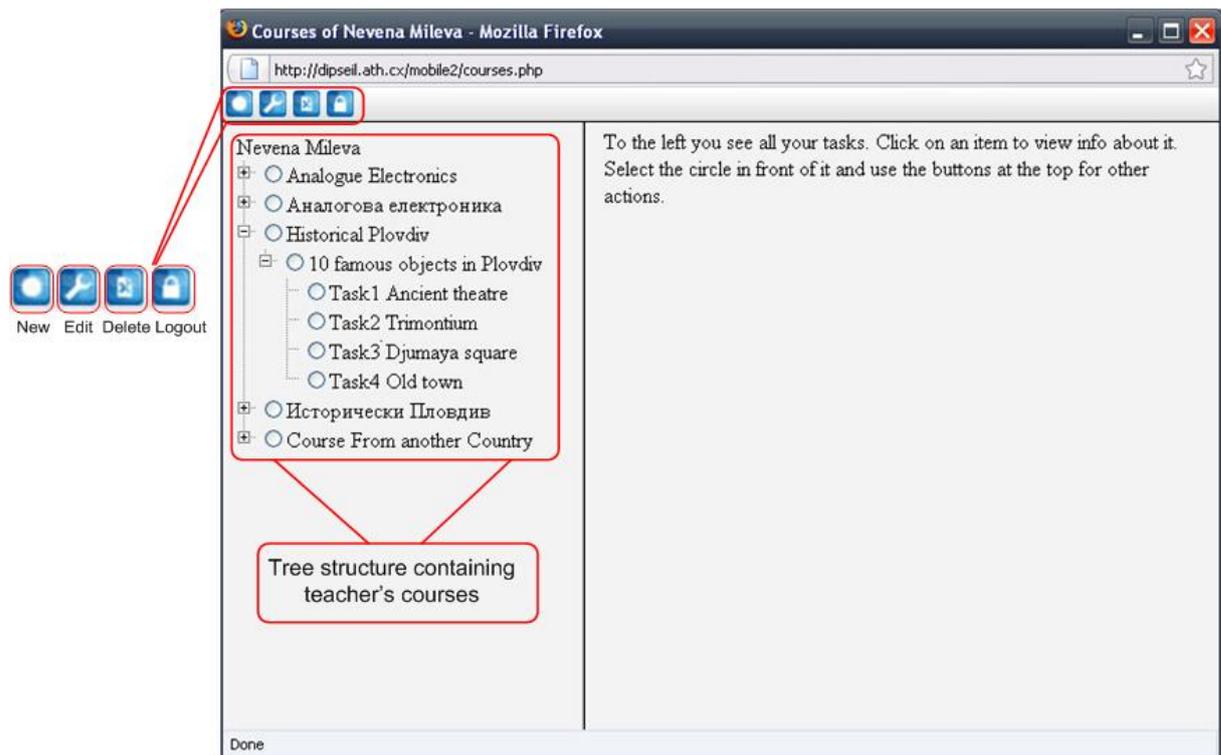
Integration with DIPSEILv2

mDIPSEIL v2 is closely integrated with DIPSEILv2. In order students to be able to enter the mobile system, they must have a valid registration to the web-based system. mDIPSEIL v2 does not have its own database with students. In the DIPSEILv2 system, students normally build their own courses from all available tasks on the system. This is stored in the database. The course of the student in the mobile system is automatically built from all tasks in mDIPSEIL v2, which are “attached” or linked with the selected in the DIPSEILv2. For example if we have selected four tasks in DIPSEILv2 and three of them have attached mobile tasks, then only these three are shown to the learner in the mobile system. If there are no attached tasks, then none are shown. To which DIPSEILv2 task, the mobile task is attached is controlled by the teacher.



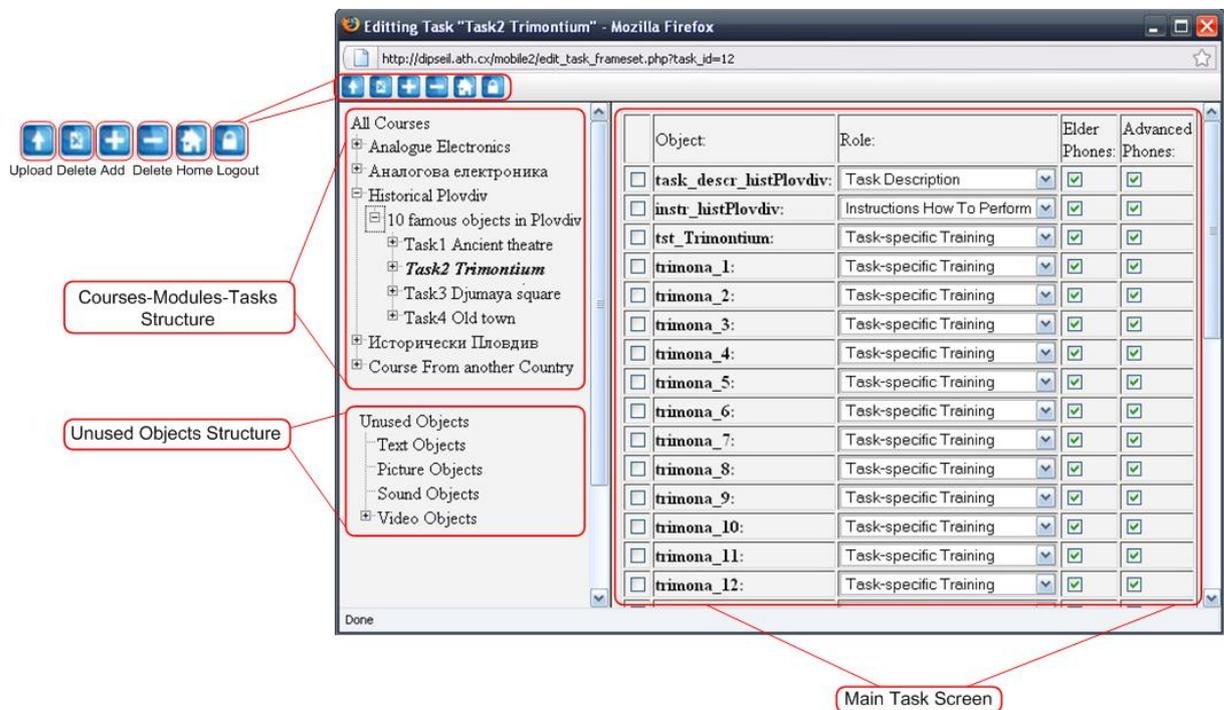
Developing Educational Content by Teachers

With the new mDIPSEIL v2 system creating, editing and viewing tasks has never been easier. Teachers are the persons, who create tasks. A valid registration with an e-mail and password is needed, in order a teacher to login the system. The system administrator must be contacted, if a new teacher should be added. After the teacher has entered the system, he sees a tree view with all his courses, modules and tasks. On the top of the window, there is a toolbar, which offers all the possible functionalities on this screen – add or remove an element from the structure, edit a task or logout. The first step for the new teacher is to create new courses, modules and tasks. Basically on this screen he creates the main structure of his courses. After clicking the “Create” button, on the right of the window a creation form will be shown. Step-by-step he must enter: what structure type he wants to create (course, module or task), then type the title of the element. In case he chooses module, he must point a course to be added to, and if he chooses task, he must select course and module. The elements of the tree on the left have radio buttons in front of them – by selecting one of the element and clicking on the “Delete” button, this element will be deleted. Deleting a whole course will also delete all its modules and tasks. The third button from the toolbar is the “Edit task” button. Only tasks could be edited and trying to edit another element will result in an error.



Choosing to edit a task will transfer the user to another screen – the “Edit Task” screen. Again to the right is the main working screen, to the left is the structure screen and on the top there is a toolbar. The structure screen this time shows two structures – the first one contains all courses of all teachers in the mDIPSEIL v2 system, but not only showing courses, modules and tasks, but also the objects. The task, which is currently being edited is written with big bold font. The second structure shows a tree of all unused objects, sorted in four groups – text, picture, sound and video. The toolbar offers several options – upload/delete object, add/remove object to/from task, go back and logout. At the first moment the main working screen shows the “Main Task Screen” – it contains information of all objects added to the task, how are they used, coordinates of the task, country of the task and information to which DIPSEILv2 task this one is added. These are all the options needed to successfully build a new task. By selecting “Upload object” button, a new form is shown to the right, offering a step-by-step procedure for uploading a new object. First the teacher should select what type of object – Text or File (picture, sound, video), after that point the file or type the text in. After uploading the object, it will be shown in the “Unused Objects” structure to the left. From there it could be deleted, or added to a task. This is done by selecting the object and the clicking on either “Delete Object” or “Add to Task” button. If an object is added to a task it cannot be deleted anymore until removed from the task and it will also be removed from “Unused Objects” and be attached to the “Courses-Modules-Tasks structure”. It will also be shown in the “Main Task Screen”. From there the teacher should select its role – “Task Description”, “Task-specific Training”, “Reference Information” or “Instructions how to Perform”. Also for what type of device it should be used must be chosen. Removing an object from the task is done by selecting it in the “Main Task Screen” and then clicking on

the “Remove from Task” button on the toolbar. Removing an object from all tasks, which it is assigned to, will remove it from the “Courses-Modules-Tasks structure” and again attach it to the “Unused Objects”. From this point it can again be deleted. After finishing with the role selecting, the teacher should attach the task to a task from the DIPSEILv2 system. For this purpose a tree with all available courses in DIPSEILv2 is presented and selecting one task from it will link the new mobile task. The last obligatory step is to choose the country, for which this task should be shown. If none is selected, it will be tagged as “Unknown Land” and will stay always hidden if the client uses the “Get Country” function from his mobile device. If the teacher wants to specify this task for some position, he should also enter the coordinates of this position. He must define four points, drawing a rectangular around this position on the map, so if the student “enters” this rectangular he will be shown this task. Coordinates may be taken from Google Earth or Google Maps. They are always in pairs, so there are four pairs of coordinates.



Viewing and Studying Educational Content by Students

In order to view the newly developed tasks, the student must download the mDIPSEILv2.jar file from the main site of the system. That is an installable Java Archive file – it must be transferred to the device using a computer-to-device connection like USB cable or Bluetooth. Another way to do this is directly download the file to the phone using its web browser and some kind of data connection. Once the file is downloaded and installed, a new icon will show up in the “Applications” folder of the phone. At the first time it is executed, it will start its initialization process – determine type of phone and locate the device using GPS. Is the phone advanced or not, depends on two criteria – first it must support the 3GP video format, and second its internet connection should be

faster than 80 kbps. For the first one the Multimedia libraries of J2ME (J2MMAPI) are invoked and they determine what kind of formats the device supports. For the second a specially created file from our server is downloaded. Its fix size of 500kb and the time needed to be downloaded from the device determine the device's bandwidth. If the criteria are (UDOVLETVOREN) then the phone is tagged as "Advanced". The bandwidth of the device is stored using the "Storage Module" to the device's memory, so the next time the application is loaded it won't need to download the file again (thus saving some money). The second step of the initialization process is to determine the phone's coordinates. The process sleeps for a minute and tries to get the position – this delay should give enough time for more accurate positioning. The coordinates are then also stored to the device memory. After this initialization process the user is shown the login screen. There he should type his e-mail and password and try entering the system. Other options on this screen are to view the "Important Data" or "Clear the Important Data". Under "Important Data" the coordinates and the internet bandwidth are meant. By selecting "Clear the Important Data" everything in the memory is cleared and a new initialization process is started. After the student has successfully logged in to the system, he is presented with all tasks available to him. These are the mobile tasks, which are linked to the DIPSEILv2 tasks, he has chosen to study. By choosing "Learn Selected" he will start studying the selected task. Other options on this screen are "Show tasks for coordinates" and "Hide tasks from other countries". The first one will invoke the Web-Engine's "Get Coordinates" module and after a processing time all tasks, which are not developed for the current coordinates will be hidden. The second one will invoke the Web-Engine's "Get Country" module, which will hide all tasks, which are not developed for the current country.

If for example a student must study a task for a famous city gallery, he must first go to next to the building. After that he should select "Clear Important Data", in order the old coordinates to be cleared and new ones to be determined. The second step should be on the "Task List Screen" to choose "Show tasks for coordinates". If properly configured only the task for the museum should be shown on the screen, and the student can start studying it.

SCORM

Sharable Content Object Reference Model (SCORM) is a collection of standards and specifications for web-based e-learning. It defines communications between client side content and a host system called the run-time environment (commonly a function of a learning management system). SCORM also defines how content may be packaged into a transferable ZIP file.

SCORM is a specification of the Advanced Distributed Learning (ADL) Initiative, which comes out of the Office of the United States Secretary of Defense.

SCORM 2004 introduces a complex idea called sequencing, which is a set of rules that specify the order in which a learner may experience content objects. The standard uses XML, and it is based on the results of work done by AICC (CBT), IMS Global, IEEE, and Ariadne.

SCORM will be integrated in the mDIPSEIL v2 system in late 2008. The most important part from a task is the “Task-specific training”, that’s why it will be packaged as a SCORM ZIP file ready to be transferred to another SCORM-supported system. Every object, which is referenced as part of the “Task-specific training” will be included in the archive.

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.

Reference information describes the task that the user has to perform. This reference information supports the user by making immediately available information, which (s)he previously had to memorize or look for in a book or a manual. The reference section allows the user to learn more deeply about a given task and is always available for her/him to read and *provides the theory behind the task it supports*.

Task-specific training reduces preliminary training by helping the user to learn while performing the task. This type of *training is learner-centred* because the learner asks for help when he needs it to perform a task, and the help gives him the specific information that (s)he requests.

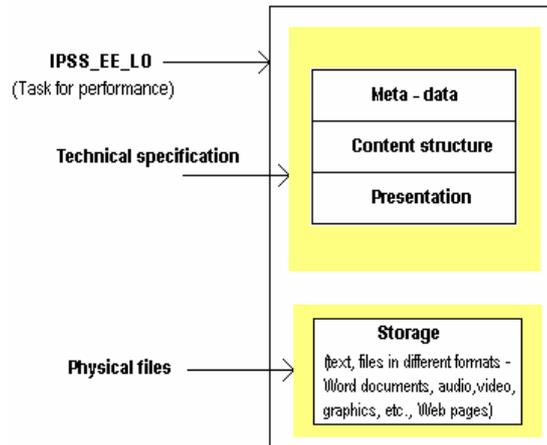
Educational performance support systems contain specific advice on performing tasks and it is its greatest advantage. The advice is usually provided by an *expert system*.

Automated tools for task performance are most helpful when a supported task involves the use of *specific software*.

The self-test provides an opportunity for the students to measure their own understanding of the content of the learning object. The students may revisit the

test/reflective activity as many times as they want. The test doesn't involve any interaction with the facilitators. The tests use computer-scored multiple choice items.

IPSS_EE_LO Model specifies content structure, meta-data, storage and presentation of IPSS_EE_LO.



Meta-data are used to describe IPSS_EE_LO. They give the basic information for the learning objects and additional elements like learning object indexing. Metadata helps the system to archive and locate valuable learning data and to select useful knowledge packages for the learners, based on the meta description. Elements of IPSS_EE_LO Model meta-data with examples:

Element title	Example
<i>Topic</i>	Task title (aims_work_name)
<i>Description</i>	Task description, file with task description
<i>Identifier</i>	Task identifier (aims_work_id)
<i>Source</i>	Which Module, which IPSS_EE (modul_id, system_id)
<i>Creator</i>	Who creates the IPSS_EE that contents the task
<i>Resources</i>	Task content (modul_work, modul_work_file, modul_description, modul_file, books, hyperlink, instruction_text, instruction_file, softuer_text, softuer_file, softuer_name, softuer_exe)

<i>Test</i>	Task element (test_id, aims_work_id, name_query, text_query, file_query, answer1, answer2, answer3, answer, answer_points)
<i>References</i>	Task element (inf_resurs_id)
<i>Example</i>	Task element (example_id, aims_work_id, example_file)
<i>Solutions</i>	Task performance (trainee_id, aims_work_id, trainee_name, solution_text, solution_file, a_date)
<i>Problems</i>	Expert advices (problem_id, modul_id, problem_number, problem_text, symptoms, solutions)

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.

Reference information describes the task that the user has to perform. This reference information supports the user by making immediately available information, which (s)he previously had to memorize or look for in a book or a manual. The reference section allows the user to learn more deeply about a given task and is always available for her/him to read and *provides the theory behind the task it supports*.

Task-specific training reduces preliminary training by helping the user to learn while performing the task. This type of *training is learner-centred* because the learner asks for help when he needs it to perform a task, and the help gives him the specific information that (s)he requests.

IPSS_EE_LO contains specific advice on performing tasks. The advice is usually provided by an *expert system*. Instructions how to perform and how to use the software are automated tools for task performance – they are most helpful when a supported task involves the use of *specific software*.

IPSS_EE_LO presentation allows the learner to perform the task using the text fields, executive files, simulation activities etc. Links provide access to all IPSS_EE_LO elements, defined in the content structure, that help the learner to perform the task – task-specific training, instructions, reference information, software for task performance and expert

advices for problems and trouble shootings. Each of these parts is presented in a new window, opened on a working environment. The rationale behind this is:

- to allow learners to access the elements of IPSS_EE_LO quickly;
- learners generally see themselves as on a work place;
- the learner doesn't lose the assumption that the main goal is to perform the task and the links are used to increase the quality of this performance.