

COMMON EDUCATIONAL WEB SPACE AS A NEW PARADIGM FOR INTERNATIONAL E-LEARNING

N. Andreeva*, N. Scerbakov**, A.Schukin*

* Institute of Computer Science and Technology
Peter the Great Saint-Petersburg Polytechnic University
St.Petersburg, Russia

**IICM, Graz University of Technology,
Graz, Austria
E-mail: nsherbak@iicm.tugraz.at

ABSTRACT

In this paper we present an innovative eLearning project being developed in collaboration by St.Petersburg Technical University and Graz University of Technology, Austria. The project aims to develop a common educational WEB space for distant teaching of different aspects of database technology. The method can be applied for teaching students which have different specialities and background knowledge. The method can be also easily adopted for so-called Life-Long-Learning (LLL), and can seriously amend a current practice of Massive Open Online Courses (MOOC).

The common educational WEB space consists of a number of Learning Management Systems (LMS) used by partner universities, reusable educational content structured as so-called SCORM modules, and a shared online database laboratory.

The previously mentioned functional components are seamlessly syndicated into an integrated system by means of a number of WEB services. For example, the laboratory is actually an Internet server with a DataBase Management System (DBMS), and a number of WEB service entries for integrating the laboratory into other eLearning components such as WEB Exams, Project developing space, and so on. The presentation provides a detailed technical specification of WEB services, and a reproduceable description of usage scenarios. The paper will present practical results of the pilot application of the system, discuss problems that were found out, and offer possible solutions.

ELEARNING AND EDUCATIONAL WEB SPACE

WEB space is normally perceived as a set of resources and services offered through the Internet. More specifically, educational WEB space is a Learning Management Systems (LMS), providing access to course specific materials, necessary communication, verification and data exchange facilities. Typically, students access written scripts or prerecorded video lectures, comment the text or movies, discuss the content using an online forum or chat. Data exchange facilities are necessary since students are supposed to perform some training actions and upload results for evaluation by teachers. Success of eLearning essentially depends on online monitoring and verification of students progress with the course material.

To implement a common educational web space we have to decide on the following aspects of such WEB sites:

- mutually recognized basic data structuring paradigm and functionality.
- data exchange standard that provides a seamless integration of data available from one site into the context of another WEB site possibly implemented in another operational environment (LMS).
- common way of evaluating students results with the course materials.

Content of a particular educational WEB site consists of the following components:

Announcements. Course announcements allow teachers to promptly announce important course events in a form of a sequence of textual documents that are sorted in a chronological order. We can say that a course announcement is a

teacher's blog devoted to a particular course. Similar to other blog applications, course announcements must be very easy to author, preferably any blog authoring tool can be applied to write such announcements. Since course announcements may contain very important information for course participants, and must be read before a particular date, there must be a variety of mechanisms that ensure delivering of announcements to students in time such as RSS (Really Simply Syndication), Atom Feed, News Groups, Emails, etc.

Curriculum description. Curriculum description is a menu of documents that describe different aspects of actual working with the course materials and tools. For example, typical curriculum documents may describe a course schedule, course content, important prerequisites, requirements for an examination, etc. Curriculum documents may be authored using any tools and uploaded in a form of HTML, PDF, MS Office documents, etc. The most preferable way of browsing such curriculum documents is a selection of documents from a list (menu). It should be especially noted that curriculum documents may be imported from another University Management System. For example, the description and calendar of all courses are available from a so-called TU Graz Online campus management system in TU Graz, and from a similar system "Portal of distant technologies" in St.Petersburg Polytechnic University.

Course library. Course library is a main learning component of any educational web site. The course library is a structured repository of training materials (files) that are available for students. Course documents can be of any format (HTML, PDF, PPT, movies, etc.). Normally, the course library is structured as a set of nested folders containing references to original documents.

The course library must provide special tools to simplify downloading of files by students, thus, any mass downloading browser plug-in can be used. Since functionality of the course library is similar to functionality of such well-known cloud services as DropBox, Google Drive and MS One Drive, interface to these web services is desired.

Course forum. The course forum is a primary asynchronous communication tool that allows to discuss course-related issues using an usual "Post/Reply" paradigm. The course forum should combine simple yet powerful text editing facilities, file uploading facilities and advanced data dissemination features such as "show new", RSS/Atom feed, email subscription, interface to news groups, etc.

Course tools. Each course web site must provide references to a number of course specific tools such as:

- online examination (course quiz);
- online project (course assignment).

A course quiz is an automatically selected list of predefined questions that must be answered by a particular student within a predefined time slot and possibly under monitoring by the teacher or course tutor. Course quizzes are evaluated automatically by the system or manually by the teacher and tutors. Students are getting examination points as a result of such evaluation. Quizzes are used as a part of final course examination or as a precondition for such examination.

A course assignment is usually a set of files developed by students on request of a teacher. Course assignments can be done individually or in collaboration with other student (working group). Course assignments can be also evaluated automatically by the system or manually by the teacher and tutors. Similar to a previous case, students get points for their course assignments.

COMMON EDUCATIONAL WEB SPACE

On the first glance, the idea of common education WEB space may be implemented as a remote access for students from one university to a LMS residing on a server of another university. Unfortunately, this mechanism cannot be used in practice, courses offered by each university have significant variations in content and especially legal restrictions for access to the course materials, online examinations, doing practical assignments online, etc. Simply stated, teachers are creative people and must have a necessary freedom to offer students teaching materials and training curriculums that they find suitable for learning the course subject. From the legal perspective, all educational web sites must match a number of restrictions that are specific for a certain country and/or university.

For example, in Austria students cannot be forced to upload their assignments onto a foreign server that is not a subject of Austrian law. Similarly, all the discussion contributions (posts) must reside on a server where university authorities have a full control.

Thus, a common educational WEB space is a set of WEB sites collaborating via export/import functionality, web services and commonly accepted protocols. We distinguish two levels of such WEB sites:

Basic level of that space is a huge collection of reusable teaching resources (modules) and a number of so-called laboratories. Modules from such basic level may be reused on an upper level that is called *user level*, via insert and clone operations (see below). Laboratories are active components that receive student's

submissions (can be answers from a particular quiz or components of students assignments) and evaluates such submissions. All the communication to laboratories is done via web services. The most obvious way of using such laboratories is evaluation of programming assignments where students are requested to implement a programming component that is automatically compiled, run and evaluated by the laboratory. In this particular project, "Fortran Programming", "C programming", "Java Programming", "MathCAD" and "Database" laboratories were implemented.

The *user level* may be seen as a number of LMS accessible by students. Servers on the user level belong to particular universities and are built for the students of this university. The servers functionality on the user level confirm to internal regulations of the university.

STRUCTURING EDUCATIONAL CONTENT

Courseware, according to our approach, consists of addressable components called modules. A module does nothing but encapsulates other components called members therefore, together with some internal structure (i.e. navigational topology). This structure is in fact a link structure expressing the relationships or associations between members. A member is either a document or another module. One member of a module is chosen and designated as its head (the starting point for navigation).

Additionally, a module may have an associated document called its label (see Figure 1) to provide contents synopsis. If a label is missing then the head is used instead.

We may think of a module as an opaque container: if we are outside the container, its members will not be visible to us. To see what's inside it, we must enter it. Of course, we can only be inside one container - the current container - at any given time. But once inside, we will be able to visit its members by navigating its link topology (see Fig. 1).

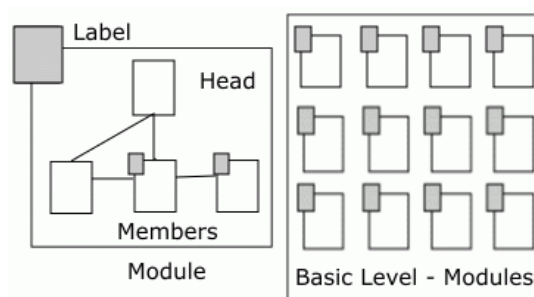


Fig.1 Internal data structuring on a basic level

A member we visit in the current container may be a document or another module. If a document, it will be visualized in some appropriate way - typically involving the presentation of the document's media objects on the computer display and/or sound system. If a module, its label (which is a document, by definition) will be visualized. The most recently visited member of the current container is the current member. Access to other members from the current member is determined by the container's internal navigational tools.

Note that visiting (i.e. navigating to) a module does not enter it, even though its label is visualized. The point of the label, as mentioned earlier, is to present a synopsis of the module to allow us to decide whether or not we want to enter it.

To enter it, we must explicitly use the Zoom-In operation. Zooming into a module makes it the new current container, its head the new current member; all links emanating from the current member become accessible.

Figure 2 illustrates the Zoom-In operation. Prior to zooming in, "A" is the current container and "B" the current member. Applying the zoom-in operation on "B" makes it the current container, its head the current member, and all the links emanating from the head accessible. Navigation after zooming in is restricted to the links in the current container, or zooming into the current member (if it is a module) or zooming out of the current collection.

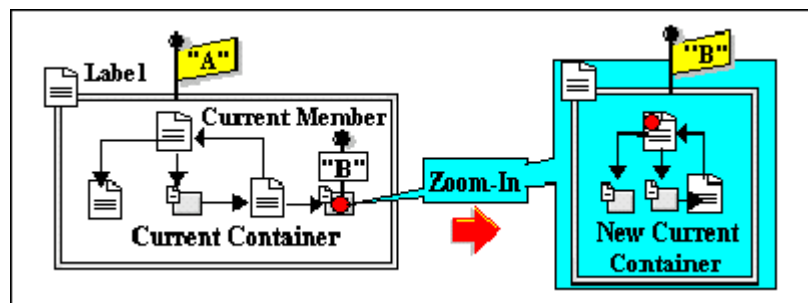


Fig.2 Operation Zoom-In

The Zoom-Out operation is of course the inverse of Zoom-In. Its effect is to restore the current container and the current member to the state immediately prior to the most recent zoom-in. Thus, a zoom-out operation for the situation in Figure 2 will reinstate "A" as the current container and "B" as the current member.

Navigation may thus be viewed as occurring in and between "planes". Link navigation within a module is in one plane, while the zoom-in and zoom-out operations are orthogonal to this and allow a jump to a lower or higher plane.

Note also that a module may be a member of more than one other module, i.e. modules may be re-used in different contexts.

Building hypermedia courseware typically follows a bottom-up approach. First, multimedia documents are created. They then become building blocks for complex modules that constitute the basic level of the system. The modules are encoded using a well-known SCORM (Sharable Content Object Reference Model) format.

RE-USING EDUCATIONAL CONTENT ON UPPER LEVEL

Assuming that documents and modules on a basic level have been created, a teacher may reuse the content to incorporate it into a particular curriculum and, hence, make it available for students. Re-using is carried out via two different mechanisms - inserting elements from basic level into modules on the user level and cloning modules from basic level.

Inserting elements is done using the following scenario. A teacher creates a new module, assigns it a unique name, selects a document or an existing module from a basic level as its head, and optionally selects a document as its label. Once a module has been created, its members - other modules or documents from the basic level, can be inserted, modified or removed.

Cloning modules may be seen as selection of most suitable modules from a basic level and creating instances of such modules on the user level. Such instances of modules can be further edited as it was described above.

USING DATABASE LABORATORY

In this chapter we describe a couple of scenarios for actual usage of the "laboratory" component in teaching different aspects of database technology. The laboratory is an Internet server with a DataBase Management System (DBMS) installation, and a number of WEB service entries for integrating the laboratory into other eLearning components.

Evaluating quiz answers.

As it was mentioned above, a course quiz is an automatically selected list of predefined questions that must be answered by a particular student. Thus, the teacher must formulate a set of questions, and possibly define templates for automatic evaluation of the user answers. In case of a database course, a most common situation is if a teacher requests to define an SQL query with a reference to a particular database schema. Obviously, it would be very useful to automatically run the query, and compare results with a predefined DBMS output to evaluate the student's answer. To accomplish this, teacher defines via special WEB services:

- a database schema and test content for a special database application;
- a number of SQL queries, each query is recorded along with an expected correct query definition.

As a student formulates an answer (provides a source query code), it is automatically submitted to the laboratory via a web-service, executed and the output is compared to a test output. All the results are returned to the teacher.

Working with database assignments.

In this case, a student or a working group of students gets a named space in the laboratory where the following actions can be performed via a web-service:

- a database schema may be defined and compiled;
- test database content may be inserted into the database;
- a number of queries can be formulated and executed in the context of the previously defined database.

Teachers may request a working protocol of a particular group. The protocol includes all the sources defined by the students and the results of execution of the queries (see Fig.3). Thus, the students assignment may be easily evaluated. This mode is primary used for doing students projects - sample database applications.

The screenshot displays the 'Check SQL' web application interface. At the top, it identifies the user as 'TU Graz WBT Master' and the administrator as 'Nikolai Scerbakov'. The interface includes a navigation menu with options like 'LogOut', 'Help', 'New Project', 'All Projects', 'Subscription Manager', 'All Files', and 'Preferences'. The main content area is divided into two sections. On the left, a 'Report' window shows a SQL query: 'select * from customer LEFT OUTER JOIN transaction ON (customer'. Below the query is a table with the following data:

| cname | ccity | cphone | cid | cid | pid | tdate | tqnt |
|---------|--------|--------|-----|-----|-----|------------|------|
| Nick | Graz | 112233 | 44 | 44 | 30 | 2003-09-30 | 1 |
| Nick | Graz | 112233 | 44 | 44 | 31 | 2003-10-01 | 2 |
| Nick | Graz | 112233 | 44 | 44 | 32 | 2003-10-02 | 3 |
| Denis | Wien | 332211 | 45 | 45 | 30 | 2003-09-30 | 10 |
| Denis | Wien | 332211 | 45 | 45 | 35 | 2003-10-02 | 10 |
| Hermann | London | 111111 | 46 | 46 | 32 | 2003-09-30 | 2 |

On the right, a 'Locker' window titled 'cccccccccc' displays a list of files. The first file is 'example.sql - 06 03 2010 14:41', and the second is 'all.sql - 13 02 2013 12:40'. Each file entry includes buttons for 'Evaluation', 'Comment', 'Report', and 'Remove'.

Fig.3 Application of a database laboratory

REFERENCES:

- [1] M.Ebner, N.Scerbakov, H. Maurer "New Features for eLearning in Higher Education for Civil Engineering", Springer Journal of Universal Science and Technology of Learning, Vol. 1, No. 1, pp. 93 – 106 (2006)
- [2] A.Scerbakov, N.Scerbakov, M.Ebner "Using Cloud Services in a Modern Learning Management System", Journal of Computing and Information Technology 03/2015; 23(1):75-86. (2015)
- [3] N.Scerbakov "TU Graz Teach-Center", Available: <http://tugtc.tugraz.at/wbtmaster/welcome.html> (2015)