

# AN ONLINE LABORATORY IN WEB BASED TRAINING

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

WEB based learning today is being widely used in almost every educational facility. In the last years all conceivable use cases that need to be carried out in such teaching mode have been defined and the features of a modern educational web site needs to offer are very clear. One of the most important aspects of such distant learning is a verification of students' progress with the course materials. In this paper we analyze a number of practical scenarios for checking students' knowledge and progress with course materials. More specifically, we discuss so-called "course quiz" and "course diary" paradigms. *Course quiz* is an automatically selected list of predefined questions that must be answered by a particular student within a predefined time slot. If we speak about web based

courses on databases, a typical quiz question might be: "consider the following database schema and define a query ..." .

*Course diary* is a list of tasks that are supposed to be accomplished by students throughout the course. Definition of tasks may significantly vary, but generally each task requires performing a certain element of a database application design, for example, developing a database schema, proving that database schema is in particular normal form, implementing a database transaction, etc.

We discuss a particular implementation of so-called web laboratory to check students knowledge on databases. We define such a web laboratory as a cloud service - set of resources and services offered through the Internet. Simply stated, the web 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 training components. The laboratory provides a possibility to build and use database applications in remote mode via a number of web services. In other words, the laboratory delivers special services for building database applications from a particular data center, the services can be used by other web client sites located throughout the world. The communication between the cloud server and clients is carried out using SOAP web services.

Our goal in this research work is to describe the architecture, and specify web services necessary for actual usage of the laboratory.

## **Course quiz**

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 (see Fig.1). Quizzes are used as a part of final course examination or as a precondition for such examination.

Normally, there are two more or less independent tools that are involved in a quiz implementation. We distinguish a quiz authoring tool and run-time quiz environment.

A quiz authoring tool supports:

- definition of individual questions;
- definition of a selected type of user input;
- definition of templates for automatic answer evaluation;
- definition of a pool of questions where individual questions are combined in a special way to provide random selection of questions for each student.

Question 8 : "Data Sub-Language SQL (1)" [4 Points]

Customer				Transaction				Product		
C#	Cname	Ccity	Cphone	C#	P#	Date	QNT	P#	Pname	Price
1	Codd	London	2263035	1	1	26.01	20	1	CPU	1000
2	Martin	Paris	5555910	1	2	23.01	30	2	VDU	1200
3	Deen	London	2234391	2	1	26.01	25			
				3	2	29.01	20			

Please define an SQL Query: "Get names of customers who bought the product CPU" using nested SQL blocks.

Answer:

```
SELECT Cname FROM Customer WHERE C# IN
(SELECT C# FROM Transaction WHERE P# IN
(SELECT P# FROM Product WHERE Pname = "CPU"))
```

Result: 2  
Well done.

Fig.1 A question from a course quiz

Definition of individual questions is almost identical to authoring small multimedia documents, we can simply see the process as an HTML authoring (see Fig.2).

Examination Editor  
quiz\_lv506255

Help Jump to Room Refresh Edit Question Duplicate SCORM New Question Cut

Examination: "Databases (706.004/706.104)"

- Topic: "Introduction to Databases"
- Topic: "Basic Relational DM"
- Topic: "Data Updating and Normalization"
- Topic: "Relational Algebra (1)"
- Topic: "Relational Algebra (2)"
- Topic: "Relational Calculus (1)"
- Topic: "Relational Calculus (2)"
- Question: "Question 7.1"
- Question: "Question 7.7"
- Question: "Question 7.2"
- Question: "Question 7.3"
- Question: "Question 7.9"
- Question: "Question 7.4"
- Question: "Question 7.10"
- Question: "Question 7.5"
- Question: "Question 7.11"
- Question: "Question 7.6"
- Question: "Question 7.12"
- Topic: "Data Sub-Language SQL (1)"
- Topic: "Data Sub-Language SQL (2)"
- Topic: "Data Sub-Language QBE"
- Examination: "Databases (Demo Test)"

Modify existing Question [Picture HTML Editor Math Editor]  
Title: [Question 7.2]

```
<CENTER>
<IMG SRC="rdlq5_01.gif" border=0>
<IMG SRC="rdlq5_02.gif" border=0>
<IMG SRC="rdlq5_03.gif" border=0>
</CENTER><BR>
Query: "Find a total number of customers who bought the product VDU."
<P>
Please select a correct definition of the query
<OL>
<LI> Range Customer C; Range Transaction T Some; Range Product P Some;<BR>
Get A(Count(C.C#));(C.C#=T.C# & T.P#=P.P# & P.Pname=VDU);
<LI> Range Transaction T; Range Product P SOME;<BR>
Get A(Count(T.C#));(T.P#=P.P# & P.Pname=VDU);
<LI> Range Customer C; Range Transaction T Some; Range Product P Some;<BR>
Get A(Count(T.P#));(C.C#=T.C# & T.P#=P.P# & P.Pname=VDU);
</OL>
```

Submit Delete Structured Answer Cancel

Define Question (Type: Selection)

1  2  3  4 Size: 4 Set Size

Check Set as Correct

Close

Fig.2 Authoring a course quiz

Selected type of user input is normally selected from a list of all supported HTML input elements such as text area, text field, select, checkboxes, radio buttons. As soon as the type of user input is selected, some additional parameters may be defined. For example, a teacher may select a radio button input element, define a number of alternatives and a legend (brief text) for each alternative.

Run-time quiz environment is the second component of an online quiz application. This component is responsible (see Fig.1) for:

- selecting a set of questions for a particular student;
- visualizing the questions;
- gathering and evaluating user answers;
- syndicating students results and exporting the results in a form suitable for a teacher.

### **Evaluating quiz answers.**

Automatic evaluation of user answers is one of the most desirable features of any e-learning platform. The algorithm of automatic evaluation essentially depends on a selected type of user input. The algorithm may be very simple if the answer is expected in a form of checkboxes or radio buttons. In this case, the teacher simply "inputs" a correct answer to the system, and the system compares two strings - the teacher's answer and a particular student's answer. In case of numeric input, the task is a bit more complex - teacher defines upper and low limits for the input value, and the system checks that the provided value matches the interval. In case of short textual inputs a so-called regular expressions may be used to evaluate user answers in many cases. Obviously, not a single of the above mentioned methods, can be used to evaluate user answers containing fragments of source text in general, and SQL queries in particular.

It seems that the only way of automatic evaluation of such fragments is execution of sources using a relevant software environment. Thus, if we speak about programming in C++, there must be a C++ compiler installed, additionally there must be a number of test inputs (test cases) to run the compiled module.

In the case of database assignments, the situation is even more complex. Any query definition (SQL) can be executed if:

- an instance of DBMS is installed on the server;
- a special database (repository of data plus database schema) was previously created;
- a test database content (a set of test table rows) was added to the database.

On first glance, automatic evaluation of SQL queries can be done via simple comparison of output of a particular SQL query and a predefined correct output provided by a teacher. In reality, the situation is not so straight forward. In SQL, there might be a number of different query definitions that produce identical results.

To distinguish different questions, every question gets a unique ID that is used to attach further parameters to the question. To implement an automatic evaluation of answers, the following parameters must be defined for each question.

- a database schema and test content, this definition may be shared by a number or even all questions in a particular question pool;
- maximum number of points that can be achieved;
- a definition of the SQL fragments that is expected as a correct result.

As a student formulates an answer (provides a source query code), the text is automatically submitted to the laboratory via a web-service, executed and automatically evaluated using the following algorithm:

- if the source SQL fragment cannot be run by the DBMS, get an error message from the DBMS, and return 0 points + error message.
- if the source SQL fragment returns a result from the database:
  - if the result does not match the predefined correct result, return 0 points + invalid result;
  - if the result match the predefined correct result,
    - calculate a distance between the SQL fragment and a correct query definition using Levenshtein Metric,
    - normalize the distance for 0 - 100 percents, where 100% means that the two query definitions are absolutely identical, and 0% means that the strings have nothing in common,
    - return a rounded value of  $[\text{max.number of points}] * [\text{Levenshtein Distance}] / 100$ ;

## **Course diary**

To understand the course diary paradigm, just imagine that a student wants to record the progress with a particular course in a form of brief weekly text chunks published regularly (say, once a week). Imagine further that a teacher prescribes to students dates when and in which form such diary entries must be done. For example, a teacher may request students to answer a couple of questions at the end of each teaching week or after finishing a certain topic. Moreover, the teacher may request students to make diary entries in a form of files that must be produced as home works and uploaded to the server. Thus, we can say that a teacher defines a number of tasks that must be fulfilled by students throughout the course. Speaking about a "database" course, we can see that the tasks may significantly vary, but generally each task requires performing a certain element of a database application design, for example, developing a database schema, proving that database schema is in particular normal form, implementing a database transaction, etc.

**TU Graz** WBT Course Diary "*Internet-based Information Systems*" Administrator: *Nikolai Scerbakov*

Back LogOut Help Refresh New Action Evaluate Export/Print Preferences

**Block 01: Databases and WWW** [Edit](#)

Practical Work:

- Install a mySQL Database management system and an HTTP server on your local computer.
- Define a database schema, create a database with a test content, and define a SQL query.
- Develop an HTML document, and publish it on your HTTP server.

Reporting Results:

- Make an SQL dump of your database (File 1), copy paste a protocol of running your SQL query including output from DBMS (File 2).
- Access the previously created HTML file (File 3) on your server with any browser, and make a screendump of the browser screen (File 4).
- Make a "Block\_01.zip" out of these four files and upload it here.

File:    ◆ **Uploaded File**

Further Comments:

**Evaluation: 4 Points - Well done**

● From 20/03/2016 00:01 Till 11/04/2016 23:59  
Done:08/04/2016 09:11

Fig.3 Course diary

### Evaluating course diary.

Evaluating course diary is a very tedious and time-consuming teacher task. The teacher must download 100s of files on a local computer, execute them, evaluate sources and results, etc. At the same time, automatic evaluation of such diary entries does not seem to be feasible. To help teachers in performing their duties we offer a concept of automatic generation of a report for evaluation. Simply stated, the teacher evaluates not a student's diary entry that may consist of a number of files of different formats and meanings, but a so-called evaluation report that automatically build by the system, and consists of texts taken from original files, results of compiling/running original files, quantitative characteristics of original files - size, number of works, distance to a template, etc. To produce an evaluation report a special work-flow script must be defined for each diary entry. The work-flow script is a list of specific actions that are selected from a set of predefined actions. The actions are performed on files submitted by students. The actions might be:

- unzipping of a submitted archive;
- sending a particular file to an online laboratory via a web service and including results into the report;
- searching for a particular fragment in an original file and inserting it into the report;
- calculating a number of words in the original file and inserting the result into the report;

- calculating a distance between an original file and a template file and inserting the result into the report;
- calculating a distance between a result received from the online laboratory, and a predefined result and inserting the result into the report.

Thus, as a student uploads a new diary entry, an evaluation report is automatically generated using a predefined work-flow script. The teacher can simply access the evaluation report that contains all the information necessary for evaluation of the entry.

The method of generating evaluation reports significantly simplified the teachers work on evaluation of course diaries and gave a positive effect on student satisfaction with the evaluation (see below).

```

block01.html
<!doctype html>
<html class="no-js" lang="">
  <head>
    <meta charset="utf-8">
    <title>Modern Information Systems - Block01</title>
  </head>
  <body>
    <h1>Modern Information Systems - Block01</h1>
    <h2>File created by Oskar Bechtold</h2>
  </body>
</html>

mis_block01_2015-10-14.sql
# *****
# Sequel Pro SQL dump
# Version 4499
#
# http://www.sequelpro.com/
# https://github.com/sequelpro/sequelpro
#
# Host: localhost (MySQL 5.5.42)
# Datenbank: mis_block01
# Erstellt am: 2015-10-14 18:13:36 +0000
  
```

Fig.4 Automatically generated report on a student's diary entry.

## Conclusion

The online laboratory was practically implemented and actively used for the "database" courses at the Graz University for Technology, Graz, Austria in the years 2015 and 2016. More specifically, a quiz on databases (10 questions) was done by 350 - 400 students each year. The quiz consists of 6 multi-choice questions, and 4 free-text SQL questions. In former times, multi-choice questions were automatically evaluated by the system, free-text questions were manually evaluated by teachers and tutors. The average work load for the evaluation of a single questionnaire was assessed as about 21 minutes. For 350 students it was about 130 working hours. Since the SQL queries were mainly evaluated manually, there was a high number of students' requests for additional inspection of their answers and results. About 30 students asked for additional inspections each year,

and unfortunately, in 10% of cases of additional inspection, the manual evaluation needed to be corrected. With the introduction of the online laboratory, the work load required for evaluating the quizzes, dramatically went down, thus, we needed an average of two minutes to evaluate a particular course quiz, or about 12 working hours altogether. What is perhaps even more important, there were no requests for additional inspection of evaluations. Students saw the results of practical execution of their queries, definition of correct SQL queries, and could understand easily why they lost points for a particular wrong answer. A very similar picture can be reported in comparison of manual and semi-automatic evaluation of course diaries.

### REFERENCES:

- [1] Bolliger, DU., Wasilik, O., "Factors influencing faculty satisfaction with online teaching and learning in higher education", *Distance Education* 30 (1), 103-116 (2014)
- [2] Martín-Blas, T., The role of new technologies in the learning process: Moodle as a teaching tool in Physics, A Serrano-Fernández, *Computers & Education* 52 (1), 35-44 164 (2009)
- [3] Liu, SH., Liao, HL., Pratt, JA., Impact of media richness and flow on e-learning technology acceptance, *Computers & Education* 52 (3), 599-607 162 (2009)
- [4] Ozkan, S., Kosele, R., Multi-dimensional students' evaluation of e-learning systems in the higher education context: An empirical investigation, *Computers & Education* 53 (4), 1285-1296 145 2009
- [5] Lonn, S., Teasley, SD., Saving time or innovating practice: Investigating perceptions and uses of Learning Management Systems, *Computers & Education* 53 (3), 686-694 145 2009
- [6] Dietinger T. & Maurer H. "GENTLE – General Network Training and Learning Environment", Proc. of ED-MEDIA98 / ED-TELECOM 98, Freiburg, pp. 274–280 (1998).
- [7] Ebner, M., Scerbakov, N., Maurer, H. "New Features for eLearning in Higher Education for Civil Engineering", *Journal of Universal Science and Technology of Learning*, Vol. 1, No. 1, pp. 93 – 106 (2006)
- [8] Ebner, M., Scerbakov, N., Tsang, P., Holzinger, A. (2011). EduPunks and Learning Management Systems – Conflict or Chance? - in: *Proceedings of International Conference on Hybrid Learning IHCL 2011 Springer Lecture Notes in Computer Sciences LNCS 6837*. (2011), S. 224 – 238
- [9] Scerbakov, A., Ebner, M., Scerbakov, N. (2015). Using Cloud Services in a Modern Learning Management System. *Journal of Computing and Information Technology* 03/2015; 23(1):75-86.
- [10] N.Scerbakov, TU Graz Teach-Center, Aviable: <http://tugtc.tugraz.at/wbtmaster/welcome.html> [13.05.2015]