

Assessing Program Outcomes

Dr. Ali Farahani
National University
School of Engineering and Technology
Department of Computer Science and Information Systems
11255 North Torrey Pines Road
La Jolla, CA 92037

1. Abstract

Assessment of the degree program objectives and the individual course outcomes are critical components of an overall degree program review and evaluation. When assessment is in a form of a test, the administration, collection and the final aggregate report generation become somewhat time consuming and generally lags behind. In this paper we present web-based software that is primarily developed for automating the course assessment process. The assessment software automates the three phases of question generation, grading and report generation; thereby saving time, reducing errors and producing report summaries that can help in decision making. This application is written in PHP and uses MySQL as the backend database system.

Keywords: course assessment, software tool, student evaluation

2. Introduction

Course assessment is the process of gathering information that can help in the design, implementation, and ongoing evaluation of a specific course. Such assessments may use variety of methods to collect information that can help instructors make decisions about designing and/or improving a course. Some of the widely used methods are capstone course evaluation, pre-test/post-test evaluations, embedded course assessment and examination, portfolio evaluation, student exit interviews, external reviewer evaluation, and even alumni and employer surveys [1]. A course assessment can take place at different stages. Some instructors use assessment early on at the beginning of the course to determine student level and needs, while other instructors periodically assess the course in progress to determine how well the course is meeting its objectives and how well students are achieving the learning outcomes of the course. Both of these approaches systematize the process of getting useful and timely feedback from students. Most instructors though assess the impact of their course and student learning at the conclusion of the course. In this case, assessments are often used to evaluate the effectiveness of the course as well as to identify needs for further change when the course is taught again [2].

This paper discusses the design and development of a database driven, web-based application for generating, correcting and reporting course outcome assessment tests. The application can easily be used for either pre/post assessments or as a course embedded assessment.

3. The need for an online assessment application

Faculty in many universities have tested and used a wide range of assessment methods to determine whether students were attaining prescribed educational goals. In larger universities with many concurrent sections of the same course, the logistics of distributing the assessment test and then collecting the individual student course grades is a hassle in it itself. Combine this with the complex task of comprehensive data analysis and report generation and soon you realize the extent of the challenge and time commitment for completing this task. This scenario can become scarier, and often does, when you consider that in many universities the lower division courses are often taught by part time adjuncts who may not have been trained on the assessment process well and tend to lag behind and require clarifications and periodic reminders.

The web based assessment application can simplify the course outcome assessment process and the data analysis and report generation part of it. Students are simply guided to a web page, where they take the assessment test online. The test is automatically graded and the student score's is returned. Many of the critical data components and attributes of the tests are stored in a backend database for later use in queries.

4. Database schema for the assessment application

The backend relational database system supporting the assessment application has 4 main tables, Course, Objective, Question, and SLOAT. The relationship among these tables is shown in figure 1.

The overall design of the tables was based on the hierarchical relationship among the main entities of the application. In our design a course is allowed to have one or more objectives. Associated with each course objective is one or more multiple choice question(s). Multiple choice questions have 4 options for possible answers. An assessment test for a given course consists of a collection of questions, one question for each objective in the course.

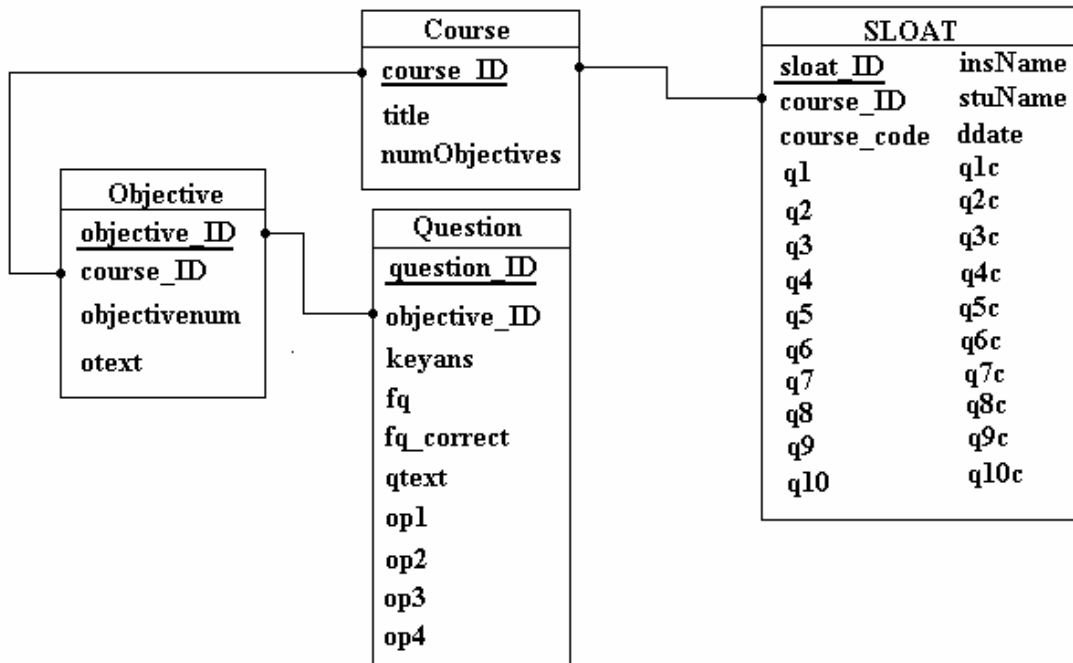


Figure 1 - Database schema for the assessment application

Some of the important attributes tracked by the application are number of objectives in each course (numObjectives), the actual text for the objectives (otext) and questions (qtext), the count on number of appearances of a given question (fq) in assessment tests, and the number of times individual questions was answered correctly (fq_correct). The SLOAT table stores an instances of an assessment test as a record, it is assumed that each course can have up to maximum of 10 outcomes, an assessment test will have a question for each outcome, therefore making 10 the maximum number of questions in each assessment test.

6. Technology selection for the implementation of the system

Building of any web based application with database connectivity requires the selection of a few products such as a database management system and a program or a scripting language to communicate with the database. We have chosen open source MySQL 5.0, a very fast, robust system as our relational database management system. This system is easy to configure and learn. We use PHP 5.0 for the server side scripting. PHP is efficient with native connections available to many database systems [3]. PHP is free and its syntax is based on other programming languages, primarily C. PHP was designed for use on the web, consequently it has many built-in functions for performing many useful web related tasks.

6. The overall structure of the application

The database tables Course, Objective and Question were populated by storing multiple insert commands in plain text in a .sql file and then executing the commands using *source <filename>*, which executes commands read from a file. Currently the creation and

development of an intuitive, user friendly interface for the purpose of populating the database tables is underway. Initially all tables, except the SLOAT table, are populated and have valid records.

To take an assessment test, students are directed to a web page, where they enter their class code, instructor name and other key information for storage in the database, see figure 2.

Standard Learning Outcomes Assessment Test (SLOAT)

Enter your name and the class code in the space provided and click on the submit button.

Enter your name:	<input type="text" value="Ali Farahani"/>
Enter instructor name:	<input type="text" value="M. Moztarebi"/>
Enter course number (i.e. CST242):	<input type="text" value="CST208"/>
Enter class code (i.e w8w95):	<input type="text" value="w451"/>
Enter date (yyyy-mm-dd): (2003-10-17)	<input type="text" value="2005-3-25"/>
	<input type="button" value="submit"/>

Figure 2 – Start page

Once the submit button on this page is clicked, the information on this page is sent to the *generatetest.php* file on the web server for further processing and eventual generation of a randomly selected set of questions. In fact, for each objective in the course, one question is randomly chosen from the set of questions specifically created to address that objective. So, it is expected that students in a class will be taking different tests. A sample assessment test is shown in figure 3.

The submission of the assessment test activates the *correcttest.php* file that resides on the server. The PHP's SESSION command is used to pass critical information from the *generatetest.sql* file to *correcttest.php*. The answer key (anskey) to each question is stored in the Question table, therefore the assessment test is automatically corrected and a short report is immediately forwarded back to each student. Data pertinent to the assessment test such as the questions, student responses, and the instructor name are stored in the SLOAT table for later access and retrieval.

cst208 Assessment Test
Answer all questions and submit once

Q1---> Differentiate the function $f(x)=1/x$

- $-x^{-2}$
- 0
- $\ln(x)$
- x

Q2---> Integrate the function $f(x)=e^{2x}$

- $2e^x$
- $2e^{2x}$
- $\ln(x) \cdot e^{2x}$
- $2x$

Q3---> Find the critical point(s) of $f(x)=x^2-x$

- $x=1$
- $x=0, x=1$
- $x=2, x=0$
- $x=1/2$

Figure 3 – A sample assessment test

The assessment application is capable of generating an aggregate report on the performance of the students on an assessment test. Currently a simple web page similar to that shown in figure 2 allows an instructor to enter the class number and code to activate the *querysloat.php* file, which is responsible for constructing a comprehensive report about the assessment test by making multiple queries from the database.

7. The course outcome assessment report

The SLOAT table is the primary source of data for generating an assessment report. Currently the report consists of a tabulated result of the students' performances that includes the percentage of correct responses to each question. This report can assist in measuring the overall student achievement of the individual course outcome. For example, the sample report in figure 4 indicates that only 25% of students correctly answered question three on the assessment test which is assessing the third course outcome.

Course Learning Outcomes Result:cst208--ww28			
Student Name	q1	q2	q3
Student 1	1	1	0
Student 2	0	0	1
Student 3	1	1	0
Student 4	0	1	0
Total	2	3	1
	50.0%	75.0%	25.0%

Figure 4 - A portion of an assessment report

A second, more detailed, table is generated by the assessment report that displays a list of course outcomes and the corresponding questions appearing on the assessment tests, see figure 5. This table provides information on the cumulative frequency of the appearances of individual questions and the number of times they were correctly answered, not shown here. This allows for the comparison of the results with those from other classes

objective	Question
Be able to differentiate functions	Differentiate the function $f(x)=x$
	Differentiate the function $f(x)=1/x$
	Differentiate the function $f(x)=5x^2$
Be able to integrate functions	Integrate the function $f(x)=2x$
	Integrate the function $f(x)=e^{(2x)}$
	Integrate the function $f(x)=1/x$
Apply derivative techniques in curve sketching	Find the critical point(s) of $f(x)=x^2-x$

Figure 5 – outcomes and their corresponding questions

8. Conclusion

The assessment application presented in this article is an accurate and easy to use tool for automating the course outcome assessment. Course outcome assessment will most likely become a mandatory component of course administration and tools such as the one covered in this article can save time and improve the accuracy of the results. Furthermore, the timely report generation can greatly help to guide decision making.

9. References

1. Outcomes Assessment.
<http://www.wisc.edu/provost/assess/manual/manual2.html> (March 2005)
2. William Peirec. Course Assessment Handbook.
<http://academic.pg.cc.md.us/aoac/AssessmentHandbook.pdf> (February 2003)
3. Luke Welling, Laura Thomson. *PHP and MySQL Web Development*. Indianapolis, Sams Publishing.