

PERSONALISED AND AUTOMATED TEST GENERATION AND ASSESSMENT

A. Andreatos
Div. of Computer Engineering & Information Science
Hellenic Air Force Academy

S. Katsoulis
Software Engineer
Greece

Abstract

Technological innovations provide powerful means for supporting education in various ways. One very important aspect of education is student assessment and feedback. In this paper we present a new tool called Exam Manager, suitable for automatic personalised test generation and assessment. The tool is user friendly, web based, and may be used in face-to-face, as well as, distance education. Exam Manager is applicable to education in general but it will unfold its full potential in engineering and science courses. Provision for adding metadata information to each question has been taken so that Exam Manager can support the instructor in composing personalised feedback reports for formative assessment.

Introduction

By the end of the 20th century there was a boom on the demand for distance learning as well as higher education. Instructors had to cope with new situations and heavy workload. Designers and Instructors of online courses have much more work in order to produce pedagogically correct results (Nurmela & Suominen, 2008).

It is estimated that distance learning grows 8–10 times faster than on-campus learning (Doukas & Andreatos, 2007). According to Burns, “up to 45 % of higher education enrolment is from adult learners, many of which sign-up for distance learning” (2006).

Assessment is strongly related to the learning/teaching process and plays a vital role in monitoring academic standards. Since instructors' time is limited, they turned to technology in order to find solutions. Thus, the revolution of e-learning lead indispensably to the evolution of Computer Aided Assessment (CAA) and e-assessment (Giurgiu, Bârsan, & Mosteanu, 2009). Today, technology-enhanced education is commonplace in many universities, both conventional and “open.” As a result, a

significant number of e-assessment systems have appeared and are being widely used. Examples of such systems are listed in Doukas & Andreatos (2007).

However, e-assessment tools available today have some important limitations (Andreatos & Doukas, 2006; Hatziapostolou & Paraskakis, 2009); most of them provide naïve types of questions which are not appropriate for higher education; what is of utmost importance, they do not unfold the potential of today's possibilities (Andreatos, 2009). A simple example will clarify this assertion. An old-fashioned way to examine programming languages in final exams was to have students write code on paper. A smarter way to do the same today with Java, for instance, is to have students write code on their laptop, compile it and debug it and finally incorporate it into an HTML page as an applet; then the instructor has simply to open the HTML pages and check the result. All the dirty work of revising the code has been moved to the students and the PCs (compiler).

Another issue of assessment is feedback; students want to have access to their feedback as soon as possible. However, feedback is not limited to just a plain score; students want and ought to know what they did wrong, what they have and what they have not learned, what they have to repeat, as well as, their strengths and weaknesses, if possible. A mark alone does not mean anything; two students getting the same mark should receive totally different (i.e., personalised) feedback. It is noteworthy that numerical assessment was introduced only in 1792 at the University of Cambridge, proposed by Prof. William Farish.

Timing is especially important in formative assessment because it enables learners to identify their strengths and weaknesses and guides them to the actions necessary in order to achieve the learning outcomes. Timely feedback allows students to recall how they addressed each question (Hatziapostolou & Paraskakis, 2009). Quality formative feedback must also be fair and directly related to the assessment criteria. Computer Aided Assessment helps instructors to provide timely feedback to students.

Computer-based training enables the evolution of novel concepts that instructors would have never contemplated of delivering a few years ago. Adaptable and adaptive learning environments (Brusilovsky, 1999), technology-enhanced learning environments (JISC, 2007), training platforms which automatically design courses that not only adapt to student capabilities but also dynamically adjust the contents, according to the students' performance during the progress of the course, are some examples. As a consequence, new types of tests are needed (Andreatos, 2009).

This paper describes an experimental e-assessment tool for higher education; in particular it refers to its architecture, usage, deployment, possibilities, advantages and pilot tests. We have called this tool Exam Manager for reasons that will become clear below.

Tool Description

The main advantages of Exam Manager are:

- It supports all known types of problems (Andreatos & Doukas, 2006), as well as future types due to its generic object-oriented and modular architecture.
- It automatically generates personalised exams and the corresponding solutions, based on students' ID and random numbers.
- It automatically grades the submitted exams and provides statistics instantly.
- It is web-based and supports both conventional and distance education.
- It is user-friendly, easy to use and provides a GUI (Graphical User Interface); no programming skills are necessary.
- It supports all majors, but it demonstrates unparalleled performance in engineering and science problems, when combined with specific software related to each course.

For simple use, the instructor can use the GUI provided. If custom engineering problems are required, then the user will have to write custom evaluation code. However, the difficulty will be comparable to that of writing code in Matlab, i.e., relatively easy. The proposed architecture will enable the implementation of any type of question, known or new, with any kind of evaluation method; it is limited only by the instructor's programming skills and imagination.

The system uses HTML forms for providing the web interface and presenting the tests, PHP for generation and processing of the exams and MySQL for the data storage needs.

Pilot Application

The pilot experiment of the prototype will take place in class at the Hellenic Air Force Academy, using the instructor's PC as a web server; students will access the system through their network-capable laptops. Three parametric computational problems with different sets of parameters will be used, leading to different results. Additionally, the problems will be shuffled by the system and mixed with simple questions, so that each student faces a different exam.

Description of the Process

Instructors teaching a subject for a series of years have authored and accumulated a pool of assessment material such as questions and problems. Exam Manager allows instructors to categorise and enter all this stuff into the system and then, based on that, edit their exams. The allowed operations are presented below:

Question Definition Process

This process involves the Instructor selecting the type of question and defining the parameters to be used with each question, by entering a definition of the question as a series of fields and rules to be used (see Figure 1).

This stage is optional, as the main system will contain a basic set of question types. If the Instructor chooses to do so, they can include arbitrary logic and evaluation code or even interface to another heterogeneous system for further evaluation. This may be done by using standard programming techniques, and always adhering to the provided Object Oriented structure of a question.

Figure 1: Definition of a new question in Exam Manager

Define a new question or problem

Title of question or problem:

Presentation of question or problem:

At the Casablanca Conference in January 1943, the Allies agreed to:

Path:

Type: Randomize order of choices: Choices are multi-selectable:

Multiple choice items:	Weight of answer:	Feedback for answers:
An invasion of Italy in the spring, invasic	100	Correct answer!
The division of Poland.	0	Incorrect, make sure you study more
Fight until the world was made "safe for	10	Too imprecise, make sure you study mor
The creation of a United Nations organiz	0	No, this was accomplished later in the...

In order to assist the composition of the personalised feedback, Exam Manager provides an additional metadata field for each question, to be filled in by the instructor. This metadata may be associated with the taxonomy of the specific question/ problem (according to Bloom, for instance), qualities assessed with it, prerequisite knowledge and/or skills needed in order to answer it, common mistakes or misunderstandings associated with it, etc.

Although this stage is optional, it is recommended, since, based on this metadata as well as class statistics, Exam Manager will provide useful hints to aid the Instructor form personalised assessment.

User Definition Process

This process involves the Instructor entering the users of the exam (students), so that it is possible to provide a means of logging in and identification for the Exam Definition, Exam Execution and Exam Evaluation Processes. The Instructor will be able to save this list of users for future use.

Exam Definition Process

During this process the Instructor uses previously defined individual questions to combine them together into an individual exam. This is done by selecting a number of questions and their respective types to comprise a full exam (Figure 2).

Figure 2: Definition of a new exam in Exam Manager

Define a new exam

The screenshot shows the 'Define a new exam' interface. The 'Title of exam' field contains 'Modern history and WW2 exam for 6th grade'. The 'Class' dropdown is set to '6th Grade'. The 'Exam duration' is set to '60'. The 'Randomize order of questions or problems' checkbox is unchecked. The 'Exam question or problem items' table is as follows:

Exam question or problem items:	Marking weight of item:
History question 1 (Modern History)	30
History question 2 (WW2)	30
History question 3 (Description of circumstances during V	40

At the bottom of the interface are five buttons: 'Add Item', 'Remove Selected Item(s)', 'Clear List', 'Add', and 'Remove'.

Exam Execution Process

The Instructor initiates the Exam by setting the status of a particular exam to “running.” This will make the Exam GUI accessible to the Students. Then, the process of the students actually taking the exam, takes place. They will first need to log into the system, using their individual user information. Then, the students will be presented with personalised versions of the exam defined in the Exam Definition Process. Thus, all students will be presented with an exam testing for the same skills and knowledge, however each producing a different numeric result.

Exam Evaluation Process

This process will be automated, in that when the user submits their exam, the system will instantaneously evaluate and grade each test, based on the Test Definition Process. Thus, after test submission, the results will be instantly available to the Examiner for further processing. Once either all Students in the list have committed their Exam answers, or the Instructor chooses to do so, the Exam status is set to “evaluated.” Access to the Exam GUI is again restricted.

Reports

As already mentioned, Exam Manager generates hints for each student automatically, based on the metadata associated with exam problems and the class statistics. Thus, this process involves the instructor using a simple interface to retrieve the exam results and

feedback hints for each individual exam for further manual processing, that is, writing a personalised feedback report.

Technologies Used in the Project

The project is implemented using standard Open Source technologies. These are namely the PHP pre-processing language, MySQL database server and a web server such as Apache. This will enable users to install and use the application in either an Intranet or Internet situation, at no cost. The simplest way to accomplish this would be by installing standard Open Source software, such as a Linux server with PHP/MySQL installed.

In addition, by using the PHP language, we have been able to implement a dynamic code execution feature, which enables us to provide the innovations presented below.

While using PHP as an interpreter for our application, we are also using additional Open Source Libraries, in order to have a more defined and manageable structure of our codebase. Namely, the VCL (Visual Component Library) for PHP library (<http://www.qadram.com/vcl4php/index.php>), which is a PHP framework designed to provide visual design capabilities. VCL for PHP is an Open Source project licensed under an LGPL licence.

Innovations and Premises for Designing the Application

Although there are other e-assessment applications in this software 'ecosystem' such as "e-Xaminer" (Andreatos & Doukas, 2006) and "e-exams" (Fragidis, Mardiris, Mizas, Simeonidis, & Chatzis, 2004), we believe that our design and implementation can introduce a number of features that will prove to provide more flexibility and power. This was the primary reason justifying the design and development of yet another e-assessment application. Thus, the innovations included are there to provide a more dynamic range to the application and execution of exams.

It is important to note here that our intention is not to replace the current standard methods of presenting various tests or problems, such as multiple choice types of questions, but rather to augment this standard capability with more dynamic and flexible ways to test and evaluate, if and when this is required. In other words, it is our intention that an examiner with low or medium computer knowledge will still be able to use the system, but for the more advanced examiners, Exam Manager will provide a framework with great and innovative capabilities.

More specifically, in addition to preparing, executing and evaluating standard types of tests, Exam Manager also provides the following:

- Flexibility in the way that tests or problems are presented, evaluated. This is accomplished by allowing the examiner to define their own HTML - based presentation of problems by entering HTML code directly as the description of the test or problem presented to the students.

- Flexibility in the type of test or problem that can be presented. This is accomplished, by allowing the examiner to enter their own fields using a specific format and their own evaluation code, which is only limited by the capabilities of the PHP language.
- Randomisation of parameters, such that each test or problem presented to the student, will test the same skills, but be different to all other tests given during the same exam. This is accomplished, by including a parametrised way of defining input parameters, such as constants and input values to functions etc.
- Randomisation of question order, so that each electronic “test sheet” has a different appearance. This is accomplished by allowing the examiner to define whether the tests will be presented in a randomised order or not.

System Architecture

As mentioned earlier, the application will make use of a MySQL Database server. The Server will be used to host a Database which will be used for storing retrieving and managing the data pertaining to the system.

The key feature implemented in this structure, is the table “examtypes” which instead of containing a reference to a static question or problem type such as a multiple choice question, is that instead, it contains code. The fields “inputfields”, “resultsfields”, “evaluators” and “description” are open ended and thus, they can be used to define a much greater range of problems and evaluation methods.

The code implementation consists of several PHP source files which implement the peripheral functions, such as defining a user, an exam type etc. There are in essence, two ways of defining a problem (examtype record):

- An automated way, which allows the definition of standard types, via the use of a wizard like interface. This will provide a simple way to define problems or questions of various usual types, without the need to write any dynamic code. In the background, the necessary code is generated automatically.
- The second way to define problems is accomplished, by allowing the examiner to enter custom field definitions, custom evaluation code, and a custom description. Scenarios, where this method can be used to greatly enhance a testing and evaluation procedure will be presented in the next section. The code used is PHP/HTML itself, and the system will store this custom code, in order to use it at the right instances to provide this additional functionality.

The implementation of the code, as already mentioned, makes use of the “VCL for PHP” framework; thus, our system takes advantage of some of the features provided by “VCL for PHP.” Some of these features that are part of “VCL for PHP” are the built-in internationalisation, the enhanced user interface experience and the use of simple-to-use but powerful User Interface components and controls.

Finally, the evaluation of the exams is enhanced by the automated process of evaluation and marking of each exam. And because of its open ended architecture, it is possible to evaluate types of problems and questions that would have not been possible to evaluate otherwise in an automated way.

Advanced Applications of Exam Manager

The system, as previously explained, is open ended, in terms of the definition of problem or question types. The system, because of its design, is capable of running arbitrary PHP code but by extension, it is also capable of interfacing and executing and/or utilising third party programs, scripts and services. Thus, one can start to imagine that the presentation and evaluation steps are not confined within strict static parameters. More specifically, Exam Manager is able to do the following due to its design:

- Definition of an unlimited number of parameters, both as input fields (i.e. problem constants or parameters) and output fields (i.e. number of answers to a problem).
- Definition of any type of problem presentation, via the use of PHP and HTML code. This will allow the rendering of practically any type of information that may be required for testing purposes.
- Definition of evaluation code of arbitrary complexity, limited only by the limitations of PHP and its interfacing capabilities.

Some examples follow on how this feature can be used to implement advanced exams.

- Since it is possible to execute system scripts and programs, it is then possible for the system evaluator to execute a system program to obtain some result on a given input. For example, a compiler program, such as the Java language compiler, can be made to attempt the compilation of a given source code and obtain the compiler results. It can then evaluate the results by the use of custom written parsing code.
- PHP has inbuilt interfacing/networking capabilities. Thus, it is possible to have PHP interface with software such as Matlab, Mathematica, Octave etc., to either use as a presentation aid and/or evaluation tool. All that is needed is the required interfacing code.

- Because of the aforementioned capabilities of PHP to interface with various external software applications and services, it is possible to link the evaluation and/or presentation of a problem or question with external Web Services such as the Wolfram Alpha computational engine (www.wolframalpha.com), which is capable of intelligent combination and calculation of input data. Thus, one could use this external web service as a “computation engine” or a data mining aid, in order to greatly enhance the effectiveness of an advanced exam.
- Likewise, since PHP is available to its entirety to the examiner, it is possible to even evaluate types of questions and tests that would never have been possible otherwise, in an automated way. Examples of this type of evaluation, include the automatic spelling checking of text, the use of Artificial Intelligence for evaluation of the content of a given text, the cross referencing of information from other sources, etc.

Future Enhancements

A number of enhancements could be added to the system, in order to improve the user experience. Some of them are listed below:

- The inclusion of more wizards for generation of exams. While all the standard question or problem types will be available to the examiner who does not want to use the open ended way of defining their tests, in the future, the system can be enhanced so that it includes more automatic test types as part of its user friendly way of defining them. These new test types could be provided in terms of “packages” of exam types, where each package could serve a particular educational theme or sector. Examples could be: Textual evaluation exam types, Mathematical Exam types, Science exam types, Computer Science exam types, etc.
- User interface enhancements such as themes, etc.
- Custom translations.
- Inclusion of templates for easier design of advanced exam types.
- Provision of personalised feedback hints to the instructor so that he/she can prepare personalised feedback reports to the students.

Conclusion

Assessment is an indispensable task of the educational process but it often becomes boring, tedious and time consuming for instructors; moreover, it might prove to be unfair due to the human factor (Giurgiu et al., 2009). Computer Aided Assessment aims at making this process fast and fair, while saving time and effort to the instructors, so they can occupy themselves with other important tasks, such as designing active learning activities and smart assessment questions.

Using the system proposed in this paper, Exam Manager, the instructor can offer exams and feedback fully electronically through a web-based interface, either in local settings such as a classroom or a PC lab, or remotely, and get results and statistics instantly; hence, it is appropriate for any type of education.

One possible drawback of e-assessment is that it is impersonal, in that it hides the students' performance (in terms of knowledge and understanding of the educational material) from the instructor. Exam Manager assists instructors in composing personalised feedback comments to the students by providing metadata associated with each question or problem, as well as, statistics for the whole class performance. This is particularly important in formative assessment.

It is our belief that the Exam Manager described in this document can enrich the effectiveness and appropriateness of any exam environment, because of its innate ability to be open-ended. The power of such a system used by a properly planned exam policy could raise the status of electronic testing and evaluation to the next level.

References

- Andreatos, A. (2009). Electronic exams for the 21st century. In N. Mastorakis, V. Mladenov, & V. Kontargyri (Eds.), *Proceedings of the 1st European Computing Conference*, Volume 1, Chapter 38 (pp. 383–390). New York: Springer.
- Andreatos, A. S., & Doukas, N. T. (2006). The 'e-Xaminer' approach: A proposed electronic examination system. *WSEAS Transactions on Advances in Engineering Education*, 3(5), 431–438.
- Brusilovsky, P. (1999). Adaptive and intelligent technologies for Web-based education. In C. Rollinger & C. Peylo (Eds.), *Special issue on Intelligent Systems and Teleteaching. Knstliche Intelligenz*, 4, 19–25.
- Burns, E. (2006). *Continuing education drives distance-learning enrolment*. Retrieved May 25, 2006, from www.clickz.com/stats/sectors/education/article.php/3605321.
- Doukas, N., & Andreatos, A. (2007). Advancing electronic assessment. *International Journal of Computers, Communications & Control*, 2(1), 56–65.
- Fragidis, L., Mardiris, V., Mizas, Ch., Simeonidis, S., & Chatzis, V. (2004). Electronic examinations "e-Exams" using Open Source Software. *WSEAS Transactions on Information Science and Applications*, 5(1), 1225–1230.

- Giurgiu, L., Bârsan, G., & Mosteanu, D. (2009). Educational issues in computer based assessment. In D. Remenyi (Ed.), *The 8th European Conference on e-Learning* (pp. 228–235). Bari, Italy: The University of Bari.
- Hatziapostolou, Th., & Paraskakis, I. (2009). Effective provision of formative feedback in an e-learning environment. In D. Remenyi (Ed.), *The 8th European Conference on e-Learning* (pp. 250–258). Bari, Italy: The University of Bari.
- JISC. (2007). *Technology Enhanced Learning Environments, JISC e-Learning Programme* (online). JISC Publication. Retrieved May 22, 2009, from <http://www.jisc.ac.uk/whatwedo/themes/elearning/tele.aspx>.
- Nurmela, S., & Suominen, R. (2008). *Become an eTeacher in a week*. Turku, Finland: Center for Extension Studies of Turku University.