Design for Web-Based On-Demand Multiple Choice Exams Using XML

Raymond Lister
Faculty of Information Technology
University of Technology, Sydney
raymond@it.uts.edu.au

Peter Jerram
School of Computing and I.T.
University of Western Sydney
p.jerram@uws.edu.au

Abstract

We describe a web-based system for the delivery and marking of multiple-choice questions. While other web systems exist that deliver such exams, those systems require either the manual mark-up of static exam questions, or generate different versions of an exam with limited variability. Such approaches result in exams that should only be administered once to a specific student. Our system is more general in how it generates different versions of an exam, using XML.

1. Introduction

A popular idea in pedagogy is the concept of assessment for learning. where student learning is structured via the assessment tasks. Ideally, students should be able to work at their own pace. Another popular idea is criterion-referenced assessment, where criteria are specified for a "pass" and all higher grades. Most university assessment is norm-referenced, where students receive a percentage score, and arbitrary boundaries determine a student's grade. A small number of large assessment tasks, and final exams in particular, lead to stress and an unhealthy approach to learning, where the goal of students is to pass rather than to learn.

The above pedagogical issues led us to conceive of the following assessment regime, for a single subject with very large class numbers (typically 500), occurring early in a university degree program. Students are assessed by small regular assessment tasks. Students work at their own pace. Each task is graded as either "satisfactory" or "not yet satisfactory". Students with the latter grade need to repeat the task until the "satisfactory" grade is achieved. An overall-passing grade for the subject is awarded when a student attains a "satisfactory" grade for all tasks comprising a core set. Higher grades are awarded if a student completes further optional tasks.

Given the marking-intensive nature of the above assessment regime, each assessment task is an exam of multiple choice questions. Many existing web-based assessment systems support multiple-choice testing (e.g. Blackboard, WebCT, and TopClass). However, they do so in a way that does not easily lend itself to supporting the above assessment regime. In some existing web-based systems, an exam is a static web page. Such exams should only be administered once to each student. In other web-based system, static questions are selected at random from a pool. While an improvement on completely static exams, such exams are still limited in their usability, and require intensive invigilation to prevent students from plagiarism.

To support our above assessment regime, where it is expected that students will routinely repeat exams, we have developed a system that uses a pool of potential questions, marked-up in XML. Whenever a student requests an exam, a subset of questions are chosen at random from the pool. Furthermore, the system also randomises the order of the choices within each chosen question. Thus students may sit the exam many times, until a minimum acceptable score is achieved, without intensive invigilation or manual intervention from the examiner.

2. The XML Mark-up of an Exam

The complete XML mark-up of an exam is illustrated in Figure 1. This particular example is very small, containing only four potential questions. In practise there should be many more potential questions.

The integer between the <Size> and </Size> tags specifies the number of questions that should be drawn from this pool to create an exam page for a particular student. In Figure 1, two questions are to be chosen.

The "Instruction", "RightAnswer" and "WrongAnswer" tags are self-explanatory. Together, they define a single potential multiple choice question. For each question, five choices are presented to a student, one of which is the right answer. However, the XML script may contain a larger number of wrong answers, from which four are chosen at random.

The "MUTEX" in one of the tags is an abbreviation of "mutually exclusive". A <Question> and </Question> pair contain one or more pairs of <MUTEX> and </MUTEX>. When selecting questions from the pool, the system selects at most one mutually exclusive element within any <Question> and </Question> pair. Thus the XML script in Figure 1 can generate four different exams, each exam containing two questions.

3. The Delivery System

The student initiates the generation of an exam by pointing his/her browser at a given URL. The web server passes the request to a Java™ Servlet. The servlet reads the XML file, and selects a random subset of questions. For each question, the right answer and four randomly selected wrong answers are placed into random order. The correct answers are saved to a back-end "Session DB" and the exam is sent to the student. By clicking on a "Submit" button, the student sends his/her selected answers to another servlet, which retrieves the correct answers from the Session DB. The student's mark is recorded in another database, and is also sent back to the student. Incorrectly answered questions are indicated, with the correct answer highlighted.

3.1. Security and Other Issues

The greatest security risk is one student impersonating another student. If student A is struggling with a particular assessment item, his friend student B, who has already passed that item, might sit the exam for student A.

To avoid this security problem, the system was designed so that the exams would be conducted in a room with an invigilator (who is not required to have knowledge of the subject under examination). The invigilator may demand photo-identification. When satisfied, the invigilator enters a password into a special field on the student's exam page. The grading system will not accept the student's submission without the password. An attractive feature of this simple approach is that there is plenty of time for the invigilator to verify the identity of students, as the students work on their exam questions. There is not a rush at the beginning or end of the session to verify identities.

Our aim was to build a simple yet flexible system. The approach to security is one example of that. Another example is the task of establishing whether a student is eligible to sit a particular test at a particular time, which is left to the invigilator, who provides authorisation via their password. Students frequently have valid reasons for sitting a weekly exam outside their normal allotted time, such as illness. Policing this issue is best left to the invigilator, rather than complicating the delivery system.

Acknowledgements:

The Java™ Servlets implementation of the online system was done by Mark Burgess, Kin Keung Kwan, Shi Sui Ling, Meng Tze Liu, and David Nguon. Java is trademark of Sun Microsystems.

```
<?xml version="1.0"?>
<Test>
<Size>2</Size>
<Question>
<MUTEX>
<Instruction>The unsigned binary
number 10101110 is the base 10
number: </Instruction>
<RightAnswer>174</RightAnswer>
<WrongAnswer>172</WrongAnswer>
<WrongAnswer>168</WrongAnswer>
<WrongAnswer>166</WrongAnswer>
<WrongAnswer>142</WrongAnswer>
<WrongAnswer>178</WrongAnswer>
</MUTEX>
<MUTEX>
<Instruction>The unsigned binary
number 10100110 is the base 10
number: </Instruction>
<RightAnswer>166</RightAnswer>
<WrongAnswer>174</WrongAnswer>
<WrongAnswer>172</WrongAnswer>
<WrongAnswer>168</WrongAnswer>
<WrongAnswer>142</WrongAnswer>
</MUTEX>
</Question>
<Question>
<Instruction>The base 10 number 189
is the unsigned binary number:
</Instruction>
<RightAnswer>10111101</RightAnswer>
<WrongAnswer>10111111</WrongAnswer>
<WrongAnswer>10011111</WrongAnswer>
<WrongAnswer>10100101</WrongAnswer>
<WrongAnswer>10110001</WrongAnswer>
</Question>
<Question>
<Instruction>The base 10 number 124
is the unsigned binary number:
</Instruction>
<RightAnswer>1111100</RightAnswer>
<WrongAnswer>1101100</WrongAnswer>
<WrongAnswer>1101110</WrongAnswer>
<WrongAnswer>1110110</WrongAnswer>
<WrongAnswer>1110010</WrongAnswer>
</Question>
</Test>
```

Figure 1. An XML mark-up of a small multiple-choice exam.