MINIMAL MARK-UP OF MULTIPLE CHOICE EXAMS USING XML

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

Peter Jerram
School of Management and Information Systems
University of Western Sydney, Australia
p.jerram@uws.edu.au

Abstract

We describe a minimal XML mark-up for multiple choice exams. In our system, exams may be generated at anytime, by choosing a subset of questions from a pool. Furthermore, the system randomises the order of the choices within each chosen question. Thus a student may sit the exam many times. Our first version of this system has been described elsewhere. In this paper, we discuss the limitations of our first system that led to our current work, and give a description of the new system features, including XML tags for supporting collaborative authoring.

Keywords

learning technology, XML, multiple choice exam, online assessment

Introduction

A popular idea in pedagogy is the concept of assessment for learning (Biggs, 1999). That is, student learning is structured via assessment tasks. However, 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. This problem is alleviated if students are given many chances to pass an assessment task, thus removing the stress caused by fear of failure.

Another popular idea is criterion-referenced assessment (Biggs, 1999). In this form of assessment, criteria are specified for a "pass" and all higher grades. Students are assessed on whether or not they have satisfied the criteria. Traditional university assessment practice is norm-referencing, where students accumulate a final mark, which then may be scaled to a desired distribution, and final grades are assigned according to the scaled mark. Norm-referencing leads to quibbles over marks, particularly when a student is close to a grade boundary. It discourages healthy skill-focused learning.

The above pedagogical issues led us to conceive of the following assessment regime, for a single subject with very large class numbers (more than 500 students), 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" (the boundary between these grades is significantly higher than 50%). 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 a small exam of multiple choice questions (MCQs).

Other Systems

Many existing web-based assessment systems, such as Blackboard, WebCT, and TopClass, support multiple choice testing. However, they do so in ways that do not easily lend themself to supporting the above assessment regime. Some of these existing web-based systems support an exam that is "hard encoded" as a static web page. This static approach results in an exam that should only be administered once to a group of students, otherwise a student may memorise the correct answers, perhaps after discussion with peers. Existing web-based systems only support a limited capacity to present a student with a random subset of questions drawn from a pool. Also, entry of questions into these systems is clumsy.

The IMS Global Consortium (2001) has developed a standard language for describing multiple choice exams, using XML (Maruyama, Tamura & Uramoto, 1999). This standard is very large, with the IMS white paper running to 124 pages, and caters for many variations on the basic multiple choice theme. For example, it provides for graphical input/output, for MCQs such as *"Click on the location of Australia in the following image of the world".* Instead of the IMS design philosophy, which attempts to anticipate all possible future uses of the mark-up language, we have opted for a bottom-up minimalist approach: we only introduce new tags when we have a need for them within our own exams.

Our Version One System

To support the assessment regime we outlined above, we first developed a web-based system that used a pool of potential questions, marked-up in XML. We have described our first system elsewhere (Lister & Jerram, 2001). In that system, a student may sit an exam at any time, by going to an invigilated PC laboratory dedicated to that purpose. The markup of a single question, from a pool of questions, is shown in Figure 1. The "Stem", "Key" and "Distractor" tags are self-explanatory, and reflect standard terminology from the literature on MCQs.

```
<Stem>What colour is the sky?</Stem>
<Key>Blue</Key>
<Distractor>Red</Distractor>
<Distractor>Green</Distractor>
<Distractor>Purple</Distractor>
<Distractor>Burnt Orange</Distractor>
```

Figure 1: An XML mark-up of a single multiple choice question, within a pool of questions

```
<?xml version="1.0"?>
<Pool>
<Question>
<MUTEX>Mark-up of single question goes here, using the tags shown in Figure 1.</MUTEX>
<MUTEX>Mark-up of single question goes here, using the tags shown in Figure 1.</MUTEX>
</Question>
<Question>
Mark-up of single question goes here, using the tags shown in Figure 1.
</Question>
</Pool>
```

Figure 2: A framework for a pool of multiple choice questions

The framework for a complete XML mark-up of a pool of questions is illustrated in Figure 2. The "MUTEX" tag is an abbreviation of "mutually exclusive". A <Question> and </Question> pair may contain pairs of <MUTEX> and </MUTEX> tags. When selecting questions from the pool, the system selects at most one mutually exclusive element within any <Question> and </Question> pair. Thus the pool in Figure 2 can generate only two different exam papers containing two questions.

Limitations of Our Version One System

After completing this first system, however, we came to see that it had been a mistake to base the system around the concept of testing students in a dedicated laboratory. PC laboratories are an expensive resource. A single class of several hundred students could easily monopolise an entire lab.

We have come to the conclusion that the multiple choice exams should be run in a conventional class room, with students sitting at desks, using pencil and paper. A student presents them self to a clerk who runs the exam room. The clerk gives the student an exam paper, from a pile of such papers, which were generated (in the same randomised manner as in our first system) and printed earlier. Each paper bears a unique serial code. After attempting the paper, the student presents their answers to the clerk, who marks the paper against a marking scheme generated specifically for that particular paper. The clerk may then record the student's mark in a spreadsheet. The clerk need not have any training in the discipline being tested. Not only is this approach less expensive to run, it is probably more secure than our earlier web-based system, as it is not vulnerable to hackers.

Also, our conception of the role of XML has changed. Initially, the web was the focus of our project. XML was peripheral; it was a convenient way of marking-up MCQs. We have come to see XML as central to the project, as the language for "business-to-business" (i.e. academic to academic) transfer of an assessment resource. These changes in our thinking require the introduction of new tags, which we describe below.

Statistical Data

Over time, the steady improvement of a pool of questions will be driven, in part, by statistical analysis of the questions. For example, analysts commonly look at two groups of students: those whose performance on a set of questions placed them in the upper quartile of overall class performance, and those whose performance placed them in the lower quartile (Ebel, 1991). A single well constructed MCQ should be answered correctly by a large percentage of the upper quartile students, but should also be answered incorrectly by a large proportion of the lower quartile students. Such performance data should be included in the XML file. Figure 3 illustrates the tags we introduced to capture statistics.

```
<Stats>
<Where>Applied Epistemology Dept, Birdsville University</Where>
<When>Autumn Semester, 2001</When>
<UpQuart>80</UpQuart>
<LowQuart>20</LowQuart>
</Stats>
```

Figure 3: XML mark-up for statistical data on each key and each distractor

Version Control and Multi-Authoring

We expect that a pool of questions will be generated as a sharing "copyleft" (Dalziel, 2000) collaboration among academics, in different institutions, over multi-year time frames. This introduces two related problems. The first problem is the preservation of statistical data. A change to a question could render invalid all the historical statistical data about that question. The second problem is recognition of authorship: how does one distinguish between a small change to an old question, and the creation of a substantially new question? Academic sensibilities about authorship must be respected.

A <VERSION> tag may safely address both problems. An academic could change an existing question, but leave the original question in the XML file, and group the old question and the new question within <VERSION> tags. It would be understood that the first question within <VERSION> tags is the most current version. The function of this new <VERSION> tag is similar to the existing <MUTEX> tag.

Open Design Issues

Assessment-for-learning implies feedback. A student should be told why their choice for a particular question was wrong. Such information could be provided in the XML, for each distractor of each question. Its use, however, is problematic. The options presented to the student are a randomized version of the distractors in the XML file, so relating a student's answer to the XML is not straightforward.

Some MCQs will require formatting features to be added to our mark-up, such as facilities to ensure that elements of a stem/key/distractor align in a column. In a primitive and clumsy way, this can already be done, by embedding HTML tags in a MCQ, and then printing out the exam paper via a web browser.

References

Biggs, J. (1999). *Teaching for Quality Learning at University.* Buckingham, England: Society for Research into Higher Education and Open University.

Dalziel, J. (2000). *Integrating Computer Assisted Assessment with Textbooks and Question Banks.* [Online]. Available: http://www.lboro.ac.uk/service/ltd/flicaa/conf2000/pdfs/dalzielj.pdf [10th September 2001].

Ebel, R. (1991). *Essentials of Educational Measurement.* Englewood Cliffs, NJ: Prentice Hall Inc.

IMS Global Learning Consortium (2001). *IMS Question & Test Interoperability QTI Lite Specification.* [Online]. Available: http://www.imsglobal.org/question/qtilite03.html [10th September 2001].

Lister, R. & Jerram, P. (2001) Design for Web-Based On-Demand Multiple Choice Exams Using *XML.* In proceedings of *IEEE International Conference on Advanced Learning Technologies.* (pp. 383-384). Madison: USA.

Maruyama, H.K., Tamura, K. & Uramoto, N. (1999). *XML and Java.* Reading, MA: Addison-Wesley.