TOWARDS AUTOMATED TESTING

Ingrid M Rewitzky

Towards Automated Testing

Ingrid M Rewitzky

Department of Mathematics and Applied Mathematics
University of Cape Town
South Africa
Tel: +27(0)21 6503206
Fax:+27(0)21 6502334
Email: rewitzky@maths.uct.ac.za

Abstract

This paper investigates an approach for automated testing. A system for generating questions and tests is described, and a case study is given where such questions and tests were used for continuous assessment in a first-year mathematics course. The very positive feedback from the students and staff indicates that this approach to continuous assessment is an excellent way of helping students master the course material and keep up to date, while simultaneously minimizing the time spent by lecturers in setting and marking tests.

Introduction

Many instructors strive to find an innovative way of assessing students that will help students of all abilities master their course material. The ideal is to achieve this without an unreasonable amount of effort on the part of the instructor. The main objective of developing an automated testing system is to arrive at such an approach for assessment.

The idea underlying the automated testing system described here is due to Jansen (1995-1998). It has several unique features which address some shortcomings of more traditional multiple choice questions and tests (described by, for example, Laurillard, 1993). In particular,

- questions and their possible answers are carefully structured so that the correct answers are not easily guessed by a process of elimination;
- there is an individualized test facility for generating a different test for each student and thereby overcoming the problems of copying among students;
- there is a computer generated analysis of student responses that provides staff with guidance on where students are having the most problems.

The system consists of two parts: one for generating questions and one for generating tests. In this paper we describe each, and give a case study of how the automated testing system was used for continuous assessment of first-year mathematics students at the University of Cape Town.

Generating questions

Any (mathematics) question written in LaTeX (or LATEX 2e) format can be encoded and added to the database. So the task of finding suitable problems is not too onerous even with little experience with the automated testing system

Suppose we wish to add the following (easy) question to the database:

> The base of a triangle has length 25cm. Its altitude is 12 cm. Which of the following is its area?
>
> (1) 120 (2) 180 (3) 150 (4) 160 (5) 140

This question may be parametrized and typed in LaTeX as follows:

> The base of a triangle has length pram(p)cm. Its altitude is pram(q)cm. Which of the following is its area?

The numeric answer, in terms of the parameters p and q, may be encoded as,

ncans = (p*q)/2.

When p and q are given the values 25 and 12 respectively, the above instance of the parametrized question with five answer options (the correct answer and four distractors) is generated. The distractors are chosen to ensure that the correct answer cannot be guessed by a process of elimination. The user may exercise some control over this process when setting a question. In particular, eight distractors may be encoded, in terms of the parameters of the question, so as to include, for example, answers corresponding to incorrect methods for answering the question. In either case the choice of four distractors and the position of the correct answer is automated.

The reason for parametrizing a question is to facilitate generation of variations thereof (which, as will be seen later, are useful for generating a different test for each student in a course). Associated with each parameter of a question is a user-defined list of possible values. Each variation of a question (with its choice of answers) corresponds to a selection of values for each parameter in the question. The selection is automated and may be subject to user-specified constraints on the relationships among the parameters. The position of the correct answer is not the same for all variations of a question.

Before adding a question (and its variations) to the database, various correctness checks are done to ensure, among other things, that the LaTeX is correct and the generated answer (for user-defined values of the parameters) is correct.

A question (and its variations) is given an identifying number and is categorized in subtopics within topics, allowing for various searching and updating facilities. New topics and subtopics may be added when updating the database with new questions. The level of difficulty of questions is also recorded as easy, moderate or difficulty.

Multi-part questions, questions including diagrams, and questions with non-numeric answers (such as equations, formulae, etc) may also be generated. For a question with a non-numeric answer eight distractors must be encoded when the question is set. There is a web page facility for maintaining the database and for ensuring an effective peer review process of its questions.

Generating test

Tests are composed by the automated testing system from questions in the database. There are two types of tests: multiple choice and free response. For each it is possible to generate a single test paper, to be written by all students, or individualized test papers, a different one for each student. The individualized test facility is useful when large classes necessitate running several test sessions. The single test facility allows instructors to draw from a database of questions thereby saving time when setting tests with free response questions.

By way of example, we consider the process of setting and marking a batch of individualized multiple choice quizzes (or MCQs for short). Typically questions of comparable standard are grouped together in (up to five) *choice pools*. The number of choice pools, the number of questions in each and the number of questions to be chosen from each is user-specified. An MCQ consists of exactly one variation of at least one question in each choice pool. For marking purposes each MCQ is given a number, which together with the correct answer sequence for the MCQ is added to the *marking key*. The batch of individualized MCQs may be stored in whatever format is required (e.g. LaTeX file, dvi file, postscript file) or printed immediately. The automatic generation of 450 individualized MCQs from 25 questions (not necessarily evenly) distributed over five choice pools takes about 10 minutes.

Once the students have written the MCQs, the student number, answer sequence and MCQ number are captured manually or using a scanner. Just like any other marker, the automated testing system requires a *mark scheme* (including also incorrect answer penalities) for each question. There is some flexibility here since each question in an MCQ need not be given the same mark and/or penalty. Excluding the choice of mark scheme, marking is entirely automated and quick: 450 MCQs are marked in at most 5 minutes. On completion of the marking two reports are produced: the results (including student numbers, MCQ numbers, correct answer sequences, student answer sequences, scores) and an analysis for each variation of each question asked in the batch of MCQs.

There is a facility for marking MCQs individually and updating the report accordingly; this is particularly useful for recording the results of late-writers. Queries may easily be addressed off-line using the report and on-line by calling up the details of a given MCQ.

Automated assessment

The case study described here involved continuous assessment in 2000 of Science and Business Science students enrolled for a first year mathematics course at the University of Cape Town. The continuous assessment included ten fortnightly MCQs that were generated and marked by the automated testing system. The actual tests were paper based.

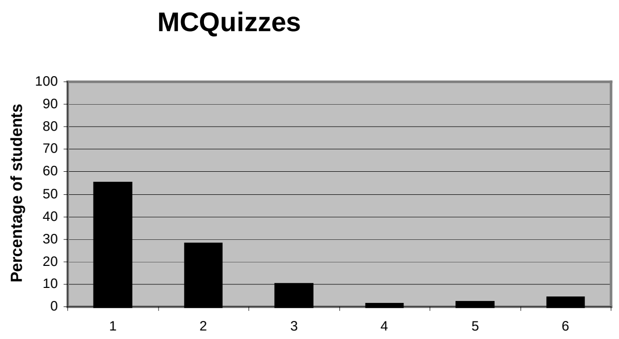
A week before an MCQ, a 'take-home' sample problem sheet of 25-30 problems (together with the correct answers) was given to the students for practice. Help was available but the actual solutions were not distributed. The choice pools used for generating the batch of MCQs consisted of questions from the sample problem sheet. The selection was made in a way that (a variation of) each question would appear in some MCQ in the batch, and the MCQs would be of comparable standard while differing as far as possible from each other in the choice of questions. There was a quick turnover time with results being available within 3-5 hours of writing an MCQ. If a student missed an MCQ for any reason, a make-up MCQ was arranged. Having generated individualized MCQs this required no extra work on the part of the instructor.

The usefulness of the automated testing system for generating free-response questions was realised when used to set tests for assessing students’ efficiency with a graphics calculus program. Each test had an MCQ component and a free response component, and all the questions were generated by the automated testing system. With the individualized tests it was possible to prevent copying of answers, something which would otherwise be difficult to enforce in computer laboratories, and it was possible to run more than one session.

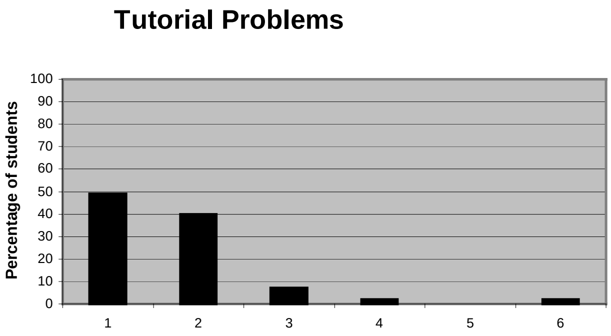
Feedback

All this may sound promising but what is the opinion of the students of the course? Most of the feedback was positive including comments such as *brilliant*; *definitely worthwhile*; *kept me on my feet*; *examples were good*; *gave me more practice with more difficult types of problems; understood the material far better and learnt the work faster*. Of course there were also a few unfavourable comments such as *I dreaded them*; *the negative marking was unfair*.

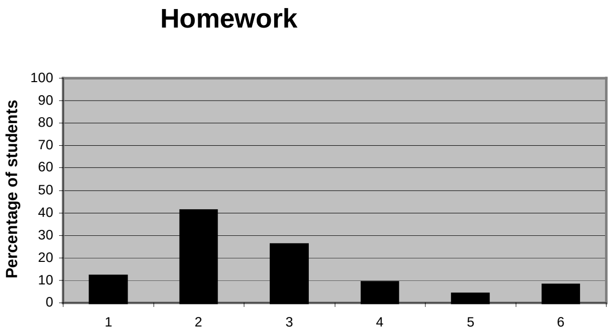
Students were asked to compare the usefulness of the various forms of preparation for tests used through the year, namely tutorial problems, homework exercises and MCQs. The results, summarised in the graphs below, show that the MCQs helped them to master the work more effectively than conventional tutorials which in turn were more useful than homework exercises. This evaluation is based on 192 respondents. For each graph, option 6 denotes no response.



1. Helped greatly to master work	55%
2. Were quite helpful	28%
3. Were adequate	10%
4. Were not helpful	1%
5. Were irrelevant	2%



1. Helped greatly to master work	49%
2. Were quite helpful	40%
3. Were adequate	7%
4. Were not helpful	2%
5. Were irrelevant	0%



1. Helped greatly to master work	12%
2. Were quite helpful	41%
3. Were adequate	26%
4. Were not helpful	9%
5. Were irrelevant	4%

In informal interviews with students most commented that they felt inspired to work through *all* the problems on the multiple choice sample problem sheets. Although the students' enthusiasm may be linked to the ability to score high marks on the MCQs, provided they worked through the sample problems, it can be argued that the fortnightly MCQs gave the students a good selection of problems to master and encouraged them to work continuously.

Conclusion

What we have done is to suggest an approach for automated testing and give a case study. The positive feedback from the class of students used in the case study can be summarised as saying that the approach is effective and an innovative way for students to master course material.

Most of the work to date has involved building an extensive database of correct questions for a first-year calculus course. Work is underway on retooling the automated testing system into a more user-friendly system with secure web-based facilities. There are prospects of using this system for outreach to other students (e.g. both at tertiary and secondary education levels).

Acknowledgements

I am grateful to Dr Arnold Jansen for the opportunity to collaborate with him on this project, and then take it over before his untimely death in 1999.

References

Jansen, A. (1995-1998). *User guide for the LaTeX based test generating system*.

Laurillard, D. (1993) *Rethinking University Teaching: A Framework for the Effective Use of Educational Technology*. London: Routledge.