AUTOMATED ASSESSMENT OF ADVANCED OFFICE SKILLS

Zlatko J. Kovačić and John Steven Green
The Open Polytechnic of New Zealand
Private Bag 31914, Lower Hutt, New Zealand
{Zlatko.Kovacic, John.Green}@openpolytechnic.ac.nz

ABSTRACT

This paper describes the design of an assessment and an automatic web-based grading system developed to assess students' spreadsheet and database skills in the *Information Systems* course at The Open Polytechnic of New Zealand. This system is fully integrated into a learning management platform (Moodle).

Our practical experience with the automatic grading system exceeded all our initial expectations. The system significantly decreased turnaround time for assignments providing either faster or instant feedback to students. At the same time the system reduced lecturers' workload freeing them up from administrative tasks and the time-consuming tasks of checking individual aspects of the spreadsheet and database applications submitted by students. A comparison of the marks allocated by markers and the automatic grading system indicates little difference in mark distributions in the last six semesters. The automatic grading system allowed for much finer probing of individual aspects of the spreadsheet and database applications with no additional work by either student or lecturer.

Keywords: Assessment, Automatic grading, Distance education, Spreadsheet skills, Database skills.

INTRODUCTION

Increased student numbers have led many tertiary institutions to consider alternative forms of assessment, such as automatic assessment systems. Further development of these systems was also triggered by a shift in the lecturer's role. Advancements in technology have also helped to enable the development of automatic assessment.

The broadest definition of e-assessment adopted in this paper is given in Whitelock & Brasher (2007). They make a distinction between computer-based assessment and computer-assisted assessment. The former refers to assessments delivered and marked by computer, while the latter relies only in part on computers. In this paper we have used the terms 'computer-based assessment' and 'automated grading systems' interchangeably.

Automatic grading systems have been in focus in computer education research for almost five decades (see Douce, Livingstone & Orwell (2005) for an overview of three generations of automatic assessment systems). However, there were only a few attempts to build a grading system for automatic assessment of Office skills (Hill, 2003, 2004; Koike, Akama, Chiba, Ishikawa & Miura, 2005; Koike, Akama, Morita & Mura, 2006; and Zhenming, Liang, & Guohua, 2003). The major expectation was that the introduction of automatic grading systems would significantly decrease the turnaround time for assessments while reducing the lecturers' workload freeing them from administrative tasks and from time-consuming routine tasks of checking every aspect of the software application students were asked to develop. However, building an automatic assessment system is not a quick or cheap option as Koike, Akama, Morita & Mura (2006) emphasised.

The goal of our research was to build an automatic grading system to assess advanced Office skills. We focussed on the conversion of the practical parts of the *Information Systems* course assignments (building spreadsheet and database applications) into computer marked assignments using Moodle quizzes.

The aims of this conversion were to design and develop tasks for the assessment of spreadsheet and database skills in distance mode without need to open and check the actual software applications and to automate the marking of practical software applications.

The conversion of assignments into the Moodle quiz format was undertaken with the expectation that it would be beneficial both for students and for lecturers. Students would get immediate feedback and their work would be marked objectively and swiftly. This would significantly reduce the lecturers' workload allowing them to allocate time to more creative activities and student support.

The rest of this paper is organized as follows: Section 2 presents related work on platforms for the automatic assessment of student programming and Office skills. Section 3 discusses our approach to the design of an assessment for assessing Office skills (namely spreadsheet and database skills). In Section 4, we give an overview of our automatic grader system and its components, briefly stating issues of implementation. Section 5 outlines conclusions and future work.

LITERATURE REVIEW

Drivers for use a computer-based assessment

Tshibalo (2007) discusses the role that computer-assisted assessment might play in distance learning institutions, listing the rationale for its use, potential benefits and limitations as well as its impact on student learning and strategies to develop effective online assessments. He identifies the following reasons for the use of online assessments: workload is increasing in higher education and in the long-run it may provide time/cost-effective student assessment. It might also help to manage the large volume of marking and assessment-related administration. Swithenby (2006) also lists several drivers for the increasing use of computer-based assessment: economic drivers (demand for portable qualifications and cost-effective means of testing) and pedagogical drivers (the most important being rapid feedback in form of both marks and comments with the potential for immediate shaping of future learning).

Whitelock & Brasher (2006) listed the following drivers for the adoption of computer-based assessment: perceived increases in student retention, enhanced quality of feedback, flexibility for distance learning, strategies to cope with large student numbers, objectivity in marking and the more effective use of a learning management system.

However, there are also some potential weaknesses and barriers to further increases in the use of computer-based assessments. As Swithenby (2006) pointed out there is still some “cultural antipathy to computer-based assessment” which is viewed as impersonal and suitable only for assessment of tasks at low cognitive level and requiring closed responses. Some academics feel that computer-based assessment cannot test high order skills such as synthesis and analysis (Musham, 2004). We feel that the acceptance and effectiveness of multiple-choice tests depends heavily on their design. With properly designed computer-based assessments even the higher ranked objectives of Bloom’s taxonomy (such as application and analysis) can be assessed.

As the development of computer-based assessment requires academic staff time and a high initial investment Whitelock & Brasher (2006) see the need for academic staff development time as one of the major barriers for greater use of computer-based assessment.

Automated assessment of Office skills

Waldman & Ulema (2008) briefly described three different methods for implementing a custom grading program for Excel. In this paper we partially adopted the first option which requires writing a grading program in Excel using Microsoft's Visual Basic for Applications (VBA). The second option requires writing a program that interfaces with a running copy of Excel (Hill's MEAGER is an example of such program). Finally, the third approach adopted by Waldman & Ulema (2008) "*is to write a program that simply reads the student's Excel data and extracts the data for the items that need to be graded*". (p. 77)

As Kline & Janicki (2003) pointed out the physical handling of the submission of assignments becomes difficult and slows the marking process. Each student receives a file that contains an embedded identifier. Once students return the Excel file, the marker runs the automated grader which compares students work with the grading templates which contain elements of Excel file to be graded, acceptable correct answers, mark allocation and the feedback comments for incorrect answers. Only two correct answers are permitted to be built in each spreadsheet cell. This feature decreases the benefits of automated grading when more than two variations in question are expected. In such cases the involvement of human markers is required. The following items are included in the automatic grading system for spreadsheet applications: name and existence of worksheet, value of a particular cell, formula, range name, use of range names, checking the validity of formulas, graphs, etc.

Koike, Akama, Chiba, Ishikawa & Miura (2005) built an automatic grading system for the assessment of intermediate Office skills. They wrote programs in VB.NET that checked, i.e. automatically marked MS Word and Excel files. A student may download the program and run it on their files for formative assessments.

Zhenming, Liang & Guohua (2003) offered an overall solution to the examination system for practical basic computing skills though they didn't provide sufficient details about their solution for us to comment further.

The most frequently cited automated graders for the assessment Office skills are systems developed by Hill (2003, 2004). Hill developed two automatic grading systems for Excel (Microsoft Excel Automated Grader - MEAGER) and Access (Microsoft Access DataBase Automated Grading System - MADBAGS).

MEAGER is a generic grading program that compares the Excel workbook submitted by student and the lecturer-supplied solution workbook and allocates marks for each submitted file. MEAGER also includes reports and tools for plagiarism detection. MEAGER is a Microsoft Access application. It extracts attributes from student's and lecturer's workbook and stores them in separate tables for each attribute. MEAGER then compares each attribute in the student's and lecturer's tables, identifies differences and records them in an errors table. The lecturer can use this errors table to identify which particular spreadsheet skill the student is lacking. MEAGER marks and embeds a grade report in the student's workbook. Then the updated student workbook can be sent back to student.

MEAGER grades various worksheet attributes: text labels, numbers, formulas, fonts, cell alignments, number formats, merged ranges, worksheet names, chart types, location and source data, etc. MEAGER does not grade conditional formatting, embedded objects or drawing objects or controls. MEAGER marks in greater detail than human graders, can detect errors a human grader cannot and can reduce the time required to grade assignments. However, there are also some weaknesses: the lecturer still has to download and upload the student workbook and also run MEAGER; the grader is in some cases too rigid to accept an alternative, but correct answer if it does not match the specimen answer exactly.

Marking formulas in Excel could be a challenging task as Hill (2004) illustrated, because in some cases the same result could be achieved in different ways, i.e. using different formulas/functions. MEAGER approached this challenge by applying an interpreter for the symbolic manipulation of mathematical expressions. An assignment should be written very strictly without even a small room for interpretation. Otherwise MEAGER would not be able to differentiate between two equivalent formulas though they answer the same question correctly. In this case a lecturer's intervention would be required. We have adopted a different approach to this problem by explaining the tasks and requirements to students in detail. So, when the function or formula is required to be entered in the online quiz students are fully aware which particular function they should use. However, our automatic grading system also accepts an equivalent formula or an alternative solution which is hard-coded in the quiz.

MADBAGS, i.e. Access Grader works similarly to MEAGER. It embeds an errors table in the student database. MADBAGS compares a correct version of an Access database with a student version and records the errors in an Access table. It grades the following database attributes: table data, table structure, field attributes, indexes, primary keys, relations, and queries. MADBAGS does not grade data access pages, macros or modules. MADBAGS can detect whether a form or report is absent, but it does not grade the form or report in detail.

E-ASSESSMENT: DESIGN ISSUES

In the Open Polytechnic of New Zealand we use the Moodle learning management system to both deliver online courses and to support students using print based materials. Moodle is an open source learning management platform that enables the creation and management of an active learning environment. Moodle is integrated with our student management system, enabling electronic submissions for essay-like assignments (both as an attachment and as a question in quizzes), online marking and in the case of Moodle quizzes, for the marks to be transferred directly to each student's record. The provision of this link from Moodle to the student management system was additional motivation for us to develop online assignments that could be automatically marked.

Course and assignments descriptions

The *Information Systems* course is a Level 5 course for the Information Systems and Technology major at the Open Polytechnic. The 71150 course has an average of 110 students per trimester with the multicultural student body consisting of two-thirds women and one third men mostly in the 30-50 age range. The majority of students are employees or self employed. Students study the course online via Moodle from all parts of the globe. Students need to pass a pre-entry test before enrolling on this course to ensure they have basic skills in Excel and Access.

Information Systems has two assignments:

Project 1: Task 1: Build the "Tiki" sales tracking system
Task 2: Business information system research

Project 2: Task 1: Build the “Tiki” database system
Task 2: Presentation
Task 3: SDLC and telecommunications technologies

Practical tasks in both assignments are developed around the case study that describes a small, local New Zealand based business.

Before conversion of these assignments into online quizzes, students were asked to submit the actual applications they were asked to build adhering to detailed client requirements, create a PowerPoint presentation explaining the benefits of the system they created and written reports. We specifically designed these practical tasks to cover higher ranked objectives of Bloom’s taxonomy (e.g. “application” and “analysis”). Therefore, as well as building spreadsheet and database applications students were also asked to use these applications to answer questions related to different scenarios relevant to the case study.

The design of assessment tasks

All the reviewed automatic grading systems built to assess Office skills require students to submit the actual application. To further reduce the time required to handle students’ assignments we designed an assignment that would ensure that the Office software was used as along with specific features, but without asking students to submit the actual application. That was at the same time the most challenging part of the conversion process. Since the table in our case study is small in size, students might try to find solutions manually. Therefore assignment tasks had to be designed in such a way as to prevent students from finding the solution manually. We will use the spreadsheet application and pivot table they were asked to create to illustrate our approach. After they have created a pivot table, we asked them to change the field settings in the pivot table to summarise the field by StdDev (i.e. standard deviation). While most of them would be able to manually create pivot table even those who completed statistical course won’t be able easily to calculate manually standard deviation and change the display of data in the pivot table unless they use Excel.

Though we tried to design tasks that students would not be able complete without using a particular software feature or in a particular way, there is always a possibility that students might workaround an assignment task by calculating the result manually. In such cases a human may need to review the quiz. For this reason we asked students

to include some screenshots from their application in their essay questions to ensure that the tasks were completed as required.

E-ASSESSMENT: IMPLEMENTATION ISSUES

As we said before, we partially adopted the first implementation option described by Waldman & Ulema (2008). We were using user-defined functions to check whether the attributes in a student workbook are according to the requirements or whether a particular function/feature has been used. Since we are not asking students to submit the actual application they have to collect information from their workbook about any discrepancies between tasks requirements and their solution. When the attributes of their workbook are checked the outcome is stored in a separate **Answers** worksheet. The results of the checking procedure are "scrambled" using an Excel function with individual students getting different numbers to prevent comparisons. Students are asked to enter these numbers from the **Answers** worksheet into the online quiz. The online quiz recognises the correct answer and allocates marks accordingly. Similarly Russell & Cumming (2005) built in the so-called "hidden database check" to ensure students did not complete the SQL query manually.

Every specific advanced spreadsheet skill such as conditional formatting, goal seek, and pivot table were assessed in different parts of the worksheet or even in a separate worksheet. Utilizing the same group of cells for assessing multiple Excel skills might complicate automatic grading as explained by Waldman & Ulema (2008).

Description of the automatic grader

Our assessment system has two parts: (1) a Moodle quiz based mostly on Cloze type of questions where students get an immediate feedback on their submission and (2) a set of essay-type of questions marked by a human marker. The Cloze question combines multi choice, numerical and short answer types of questions into a single question.

The first part of the quiz is marked automatically; no intervention from lecturers is required. Students are not required to submit the actual software applications. Once they build the software application they need to answer a few questions by inserting the requested information into the quiz. The information and answers they are inserting are generated by the application.

Preparing spreadsheet for use

For spreadsheet application students need to download a few Word documents containing the data for this task and an Excel file which contains four worksheets (the last one is the **Answers** sheet). The student then edits the file as required by the assignment. The initial Excel file helps us in the clarification of the assignment and in controlling the format of worksheets. This simplifies the marking process and keeps the students focus on the advanced Excel features rather than dealing with basic formatting issues.

Detailed instructions are provided on how to prepare the initial workbook for use. The preparation of the workbook ensures that all the user-defined functions required for checking the student workbook attributes are working correctly. To ensure students won't have any technical problem with the actual quiz we introduced an example quiz as part of an incremental assessment strategy. Students are given a password after scoring 100% in the example quiz to access the actual quiz. The example quiz did not contribute towards the student's grade and students had the opportunity to take it multiple times. Similarly Russell & Cumming (2005) split the assessment into four assessments. Only when student completed 75% of a tutorial group was the related assessment made available.

The **Answers** sheet is password protected. It contains answers to the following questions:

- How many worksheets does the Excel workbook contains?
- Are the worksheets named and ordered as required?
- Are the column labels replaced with descriptive names?
- Is the currency format used where appropriate?
- Is the requested information inserted in the left/right footers?
- Is the page setup as required?
- Has the AutoFilter been used?

These are all requirements built in the assignment, but student are not aware that e.g. *Answer (1)* is related to the number of sheets in the workbook, etc. They know about all these requirements but the numbers that the system is inserting into the **Answers** worksheet have no meaning to them.

User-defined functions

The user-defined functions check whether the attributes in a student workbook are according to the requirements. The outcomes of these user-defined functions are then used in the **Answers** worksheet to generate random numbers. There are two possible ranges of these random numbers. Numbers from the first range indicate the correct answer, while the numbers from the second range indicate an incorrect answer.

For each of these questions listed in previous subsection a user-defined function was created. For example, the function SheetsCount() counts how many worksheets are in the Excel workbook (**Figure 1**).

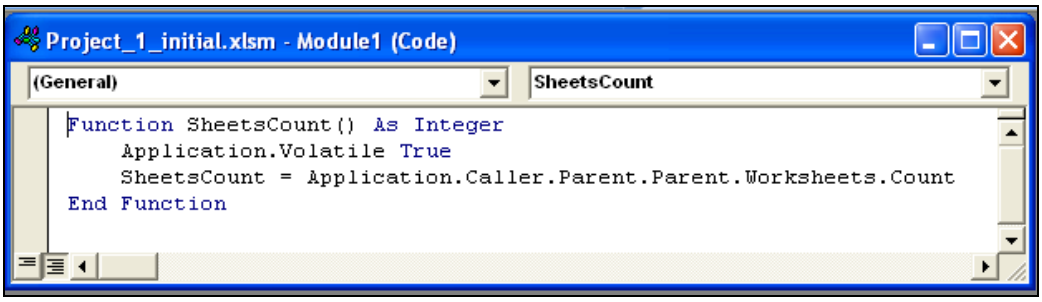

Figure 1: User-defined function – SheetsCount()

For further protection we additionally scrambled the output from these functions by using the RANDBETWEEN() Excel function. This function assigns a random number from a specified interval to the correct answer. Any number from the specified interval will be accepted as a correct answer.

How the quiz works

In the case of a few key Excel functions, such as IF and VLOOKUP functions, each argument of the function was requested to be entered separately. Separated arguments allow us to allocate marks for a single argument if it is correct, reducing the chance of students making typos on entering a formula or function and at the same time helping students not to skip some of the arguments.

The **Figure 2** illustrates one of the questions in the online quiz.

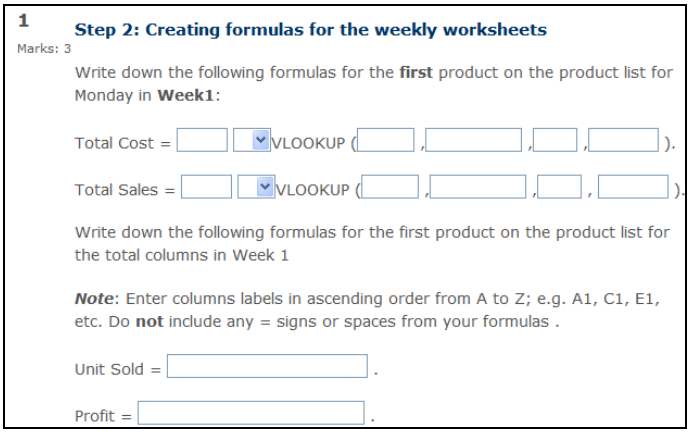

Figure 2: Quiz question

Once they submit the quiz, the system will mark their submission automatically (**Figure 3**). The actual answers were removed from the screenshot.

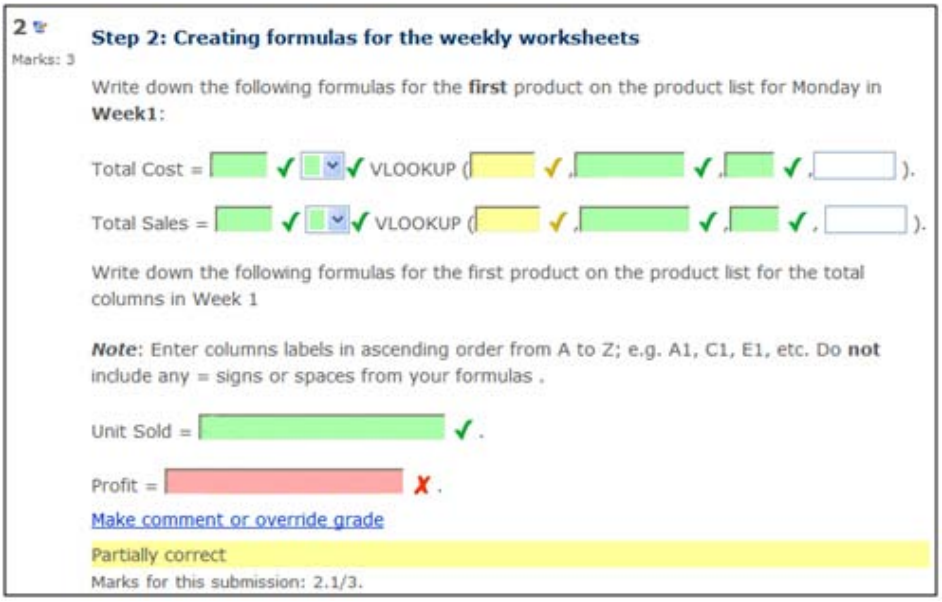

Figure 3: Quiz answers

The Cloze question allows allocating a fraction of mark for a partially correct answer and also accommodates the multiple ways in which each question can be answered and scored correctly. Each question within Cloze question could have a different weight depending on whether the question is more important than the others.

After the submission date scores are automatically transferred to the student management system (SMS).

Results

The online quizzes were implemented for the first time in Semester 2, 2009. From student feedback received from the course survey we believe that most of them loved the fact that they received immediate feedback and the marks they have scored:

> "*Loved real time assessment*"

> "*The online assignments – with the results coming back instantly were awesome*"

A few students said that they disliked the fact that the automatic grading system required attention to such details as spelling, spacing and absolute correct answer in their input:

> "*The assessments are far too oriented towards achieving the absolute correct answer and totally ignore the method used by student. Assessment marks were based on putting the absolute correct answer within the absolute correct cell within a spreadsheet with no consideration for the method used.*"

We would argue that accuracy in spreadsheets is a fundamental requirement. No one would accept results that are almost correct or range names that were "close enough".

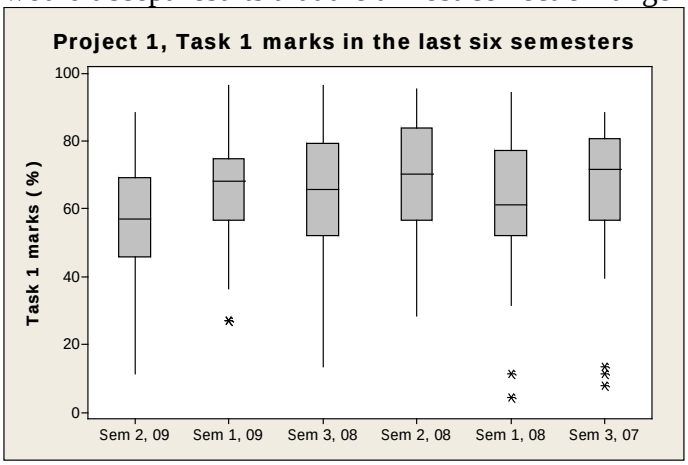

Figure 4: Comparison between automatic and manual marking

The distributions of marks for spreadsheet task are presented in **Figure 4**. The first box-plot on the left shows the distribution of marks allocated by the automatic grader. The rest of box-plots show the distribution of marks in each semester allocated by human marker.

CONCLUSIONS

Using the user-defined functions we've been able to check the attributes of software applications and see whether the requested feature or function was used appropriately. Lecturers were not required to open the actual application because students verified their application by entering scrambled user-defined functions outputs in the online quiz. Alternative correct answers and partially correct answers were coded in the quizzes allowing for flexibility in acceptance of different solutions.

We found the following advantages of our automated grader system to students and staff.

For students:

- Instant feedback on formative real time assessment (RTA) quizzes assists learning and motivation
- Instant feedback on summative RTA quizzes provides instant feedback on their success
- RTA quizzes provide structure for the less organised, less clear thinking students

For staff:

- Although the initial setup involves a substantial amount of thought and effort the result of this has great ongoing workload benefits from then on enabling the staff to concentrate on providing ad-hoc feedback to student questions. This is therefore also a student benefit.
- As staff workload is reduced staff are able to do research which underpins the degree course and remain current. This is therefore also a student benefit.
- Moodle quizzes enable lecturers to receive statistics on the validity of their questions.

One possible extension of this research to pursue in the future, as suggested by one conference reviewer, would be comparing the manual and automated marking of a randomly selected sample of assessments in a future offering of the course.

At the end what could be said about the future of automated assessment? We agree with the vision outlined in Whitelock & Brasher (2007) paper that

> "*traditional paper-based summative assessments will continue to migrate to computer delivery …. Increasingly, aspects of courses that lend themselves to objective question types, or that use assessments based on visualisations of concepts or procedures, will be completed online. These strategies are likely to be combined with short-answer questions, marked by computer and checked by humans, to probe the learner's ability to form links between areas of knowledge*". (p. 36)

ACKNOWLEDGEMENT

We would like to take this opportunity to thank two anonymous conference reviewers for their time and extremely valuable work which helped us to improve the final version of this paper. However, the authors should be held responsible for any remaining errors.

An extended version of the paper was presented at the INTED2010 International Technology, Education and Development Conference held in Valencia, Spain from 8th to 10th of March, 2010.

REFERENCES

Douce, C., Livingstone, D., & Orwell, J. (2005). Automated test-based assessment of programming: A review. *Journal of Education Resources in Computing, 5*(3).

Hill, T. G. (2003). MEAGER: Microsoft Excel automated grader. *Journal of Computing Sciences in Colleges, 18*(6), 151-164.

Hill, T. G. (2004). Excel grader and Access grader. *ACM SIGCSE Bulletin, 36*(2), 101-105.

Kline, D. M., & Janicki, T. N. (2003). Enhancing economic and finance learning through automated grading of spreadsheet exercises. *Journal of Economics and Financial Education, 2*(2), 23-29.

Koike, H., Akama, K., Chiba, M., Ishikawa, T., & Miura, K. (2005). Developing and e-learning system which enhances students' academic motivation. *The 33rd Annual ACM SIGUCCS Conference.*

Koike, H., Akama, K., Morita, H., & Mura, K. (2006). Using an automatic marking system for programming courses. *The 34th Annual ACM SIGUCCS Conference*.

Kovačić, Z., & Green, J. (2010). Automated grader of advanced office skills. In the *Proceedings of the INTED2010 International Technology, Education and Development Conference*, Valencia, Spain, 8-10 March.

Musham, G. (2004). *Using computer technology to support student assessment*. OET Assignment, University of London.

Russell, G., & Cumming, A. (2005). Automatic checking of SQL: Computerised grading. *International Journal of Learning, 12*(3), 127-134.

Swithenby, S. J. (2006). *E-assessment for open learning*. The Open University, UK.

Tshibalo, A. E. (2007). The potential impact of computer-aided assessment technology in higher education. *South African Journal of Higher Education, 21*(6), 684-693.

Waldman, M., & Ulema, M. (2008). Automated measurement and analysis of effectiveness of teaching selected Excel topics in an introductory IS class. *Journal of Computing Sciences in Colleges, 23*(5), 73-82.

Whitelock, D. & Brasher, A. (June, 2006). *Roadmap for e-assessment.* JISC.

Whitelock, D. & Brasher, A. (2007). Effective practice with e-assessment. An overview of technologies, policies and practice in further and higher education. JISC.

Zhenming, Y., Liang, Z., & Guohua, Z. (2003). A novel web-based online examination system for computer science education. *The 33rd ASEE/IEEE Frontiers in Education Conference*.