Article:
Walker, P, Gwynllyw, DR and Henderson, KL (2015) Diagnosing Student Errors in e-Assessment Questions. Teaching Mathematics and its Applications, 34 (3). 160 - 170. ISSN 0268-3679

https://doi.org/10.1093/teamat/hrv010

Diagnosing Student Errors in e-Assessment Questions

Philip Walker†, D. Rhys Gwynllyw‡ and Karen L. Henderson*‡

†School of Mathematics, University of Leeds, Leeds, LS2 9JT, United Kingdom
‡Department of Engineering Design and Mathematics, University of the West of England, Bristol, BS16 1QY, United Kingdom

*Email: Karen.Henderson@uwe.ac.uk

In this article we demonstrate how the re-marker and reporter facility of the DEWIS e-Assessment system facilitates the capture and reporting of student errors. The process of diagnosing student errors is illustrated in two case studies. The first involves an e-Assessment given to level 1 computing students at the University of the West of England on the general topic of indices and logarithms and the second involves an e-Assessment taken by level 2 mathematics students at Leeds University on the topic of Sturm-Liouville problems. These e-Assessments involve numeric and algebraic inputs, two common type of inputs used for mathematical e-Assessment questions, and the difference in approaches needed for each type of input is discussed. The key advantages to being able to efficiently capture and report student errors are threefold. Firstly, this information may be used to improve the questions by providing enhanced, tailored feedback which will benefit future students. Secondly, through the use of the re-marking facility, current students, who have already tried the e-Assessment, may access this new improved feedback by viewing details of their previous attempts. Thirdly, by looking at the results for a particular cohort, the academic is able to see which areas of the syllabus need more emphasis in lectures.

1 Introduction

Using e-Assessment for formative and summative means has become standard practice in many University mathematics departments (Sangwin, 2013). This is due in part to academics having access to open-source algorithmic e-Assessment systems, such as *STACK* (Sangwin, 2004), *Numbas* (Foster, Perfect and Youd, 2012), *DEWIS* (Gwynllyw and Henderson, 2009) and *Math e.g.* (Greenhow and Kamavi, 2012) and also due to the many advantages that e-Assessment affords, such as providing students with instant feedback in a time-efficient manner. A fuller review of the benefits of e-Assessment can be found in Bull and McKenna (2003).

The e-Assessment systems listed above have the capacity to give a fully worked through solution to the question asked. Greenhow and Gill (2008) found that students learn from e-Assessment feedback, using it to perfect their technical knowledge and there is evidence that students find the availability of practice tests to be one of the most useful study resources which supports their learning (McCabe, 2009). However, one of the potential barriers to the uptake of such systems by lecturers is the perceived lack of individualised feedback (Broughton, Robinson and Hernandez-Martinez, 2013).

A mal-rule, or common student error (CSE) is a consistent but incorrect rule used by a student (Glendinning, 2008). Understanding *why* a students is making a mistake as opposed to simply identifying their mistake was the motivation for the research of Seely Brown and Burton (1978) on creating diagnostic models for procedural bugs in basic mathematical skills. Payne and Squibb

(1990) examined paper-based in-class tests given to children at three different secondary schools in an attempt to classify the algebra mal-rules made in solving linear equations with a single unknown. They reported that the process of finding and classifying CSEs was time-consuming and concluded that the frequency of mal-rules is extremely skewed. Gill and Greenhow (2007) examined several years' worth of paper-based exam scripts in order to discover mal-rules used on mechanics questions and attempted to characterise them with metadata. They created e-Assessment questions covering this material. For the multiple-choice questions (MCQs) the mal-rules found were used to create distractors and tailored feedback was provided if a particular distractor is chosen by the student. Jordan (2007) analysed student answers to interactive online assessment questions taken by science students in order to gain insight on their mathematical misconceptions. This information was used to improve the questions for subsequent years giving targeted feedback in response to commonly incorrect responses.

A key advantage to capturing and reporting CSEs within an e-Assessment is that students may receive tailored, personalised feedback, which will enable them to see where they are going wrong and to repair their understanding for future attempts. In this way the e-Assessment is able to simulate the human marker. Also by examining which particular mal-rules have been triggered by a cohort of students, in an easy to read format, the academic is in a position to tailor future classes to address any misconceptions that have arisen. For e-Assessment systems which have the capacity to store all data from students' assessment attempts, there is a wealth of post-assessment information available which may be analysed. The focus of this paper is to illustrate how the re-marker and reporter facility of the DEWIS e-Assessment system facilitates the capture and reporting of CSEs. In particular we focus on two e-Assessments run at different institutions involving numeric and algebraic inputs. These are two common type of inputs used for mathematical e-Assessment questions, and the difference in approaches needed for analysing mal-rules in each type of input is described.

2 Methodology

The e-Assessments were run using DEWIS, a fully algorithmic open-source e-Assessment system, which was designed and developed at UWE. It was primarily designed for numerate e-assessments and is currently used in the fields of Business, Computer Science, Nursing, Engineering and Mathematics (Gwynllyw and Henderson, 2009; Gwynllyw and Henderson, 2012). The DEWIS system is *data-lossless*, that is, all data relating to every assessment attempt is recorded on the server. The DEWIS system, via its reporter feature, facilitates a detailed analysis of every e-Assessment run. The analysis of an e-Assessment is much more than the simplistic approach of analysing student's marks. For example, the analysis includes the use of performance indicators (PIs) to identify the triggering of mal-rules. The analysis also includes a search mechanism to identify previously unanticipated student errors. Such 'new' mal-rules can be fed back into the e-Assessment's marking and feedback schemes for detection and reporting. Not only will future students benefit from this updated feedback but it will also benefit current students; using the data-lossless feature, the updated feedback can easily be applied retrospectively to past assessments.

In this paper we aim to illustrate in detail the process of identifying CSEs in two separate e-Assessments and to demonstrate how the re-marker and reporter facility of DEWIS facilitates this process. The first case study concerns an e-Assessment given to a cohort of level 1 computing students at the University of the West of England (UWE) on the general topic of indices and logarithms. The second case study concerns an e-Assessment requiring students to put a differential equation into Sturm-Liouville form given to level 2 mathematics students at Leeds University.

The process of searching for CSEs in a traditional paper-based assignment, where every student sits the same paper and submits their workings to each question, although time-consuming is relatively straightforward. Typically, in the process of marking, similar wrong responses, submitted by students can be spotted. For e-Assessment questions the task is potentially more difficult, because firstly no intermediate workings are submitted and secondly each student will be attempting a different but equivalent version of the question, due to the use of random parameters. Some student errors can be anticipated in advance of running the e-Assessment and coded into the question from the start. In order to spot a new candidate CSE, it is necessary to examine wrong answers submitted by a student and where possible to determine the mal-rule that may have been used to achieve this wrong answer. By coding this candidate CSE into the question and retrospectively re-marking all the submissions it is possible to see how many students triggered the same mistake. Having identified a new mal-rule, the feedback to the question was amended to provide detailed, tailored feedback in this situation. This process may be repeated until all the incorrect responses are exhausted.

The educational benefits to being able to efficiently capture and report CSEs to students and staff are threefold

- this information may be used to improve the questions by providing enhanced, tailored feedback which will benefit future students taking the e-Assessment;
- through the use of the re-marking facility, current students, who have already tried the e-Assessment, may access this new improved feedback by viewing details of their previous attempts;
- by looking at the results for a particular cohort, the academic is able to see which areas of the syllabus need more emphasis in lectures.

3 Case Study 1

In this section we shall illustrate how DEWIS facilitates the diagnosing of CSEs for an e-Assessment given to a cohort of level 1 computing students at UWE. The e-Assessment content was on the general topic of indices and logarithms. The material covered in this e-Assessment was not formally taught in the award but was part of a directed reading assignment. The purpose of the e-Assessment was two-fold. For a period of two weeks the students were given access to the e-Assessment in formative mode as part of the learning process. After this period, students were allowed two attempts in summative mode. Full feedback was provided at the end of each e-Assessment attempt for both delivery modes.

The e-Assessment in question contained eight questions and we will concentrate on the analysis of data for one of the questions asked in this assessment. In formative mode, students were allowed up to a maximum of five attempts. In all, there were 329 submissions from 110 distinct students and 81 responses for that question were incorrect. In the following discussion, we shall include snap-shots of displays provided by the reporter facility on DEWIS. Note that the student identities in these displays have been anonymised.

Figure 1 shows the Reporter output regarding the marks awarded per question for the e-Assessment. Each one of the marks is actually a hyperlink. On clicking the link the academic can view the actual instance of the question that was asked, together with the result of the marking and feedback process for that particular question.

Students may view all their previous assessment attempts, with the resulting view being similar to that shown in the pop-up box in Figure 1. One significant disadvantage of displaying the results in the form shown in Figure 1 is that it is not possible for the academic, without clicking on each question link, to view why a student, or a student cohort, has scored specific marks. For example by viewing all the data corresponding to Figure 1, we would only see that a significant

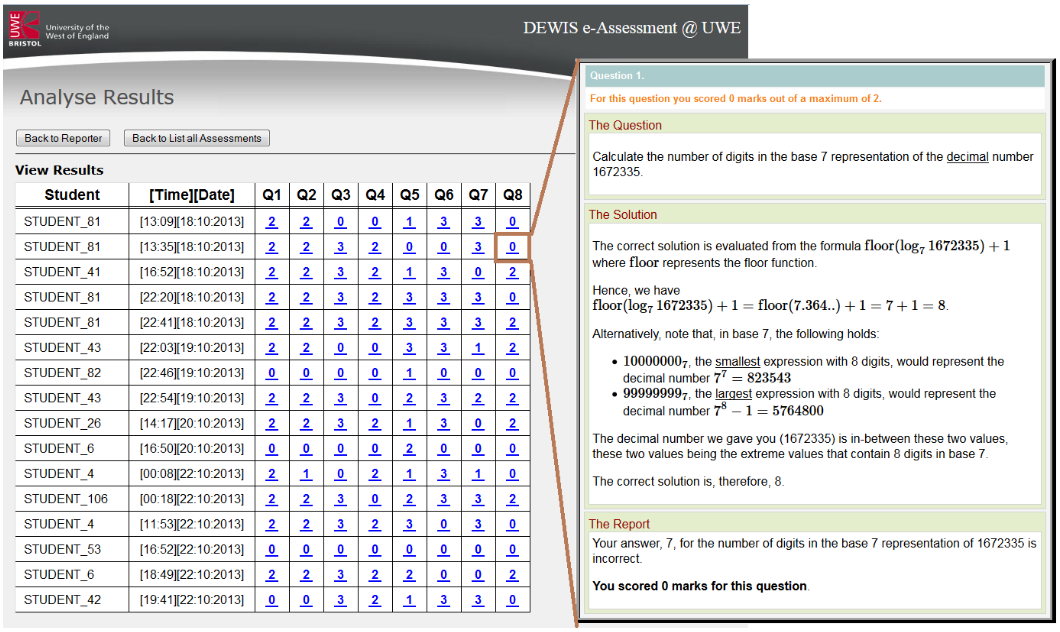


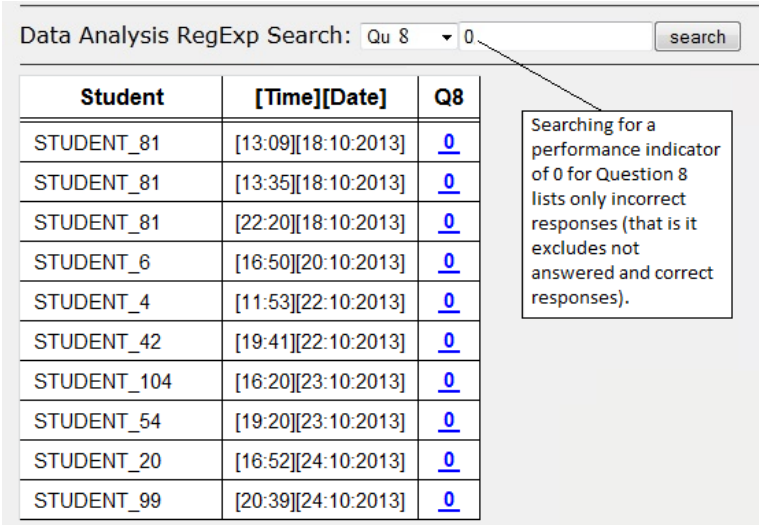
Figure 1: Data from the first sixteen e-Assessment submissions, showing the marks awarded per question. Each mark is a hyperlink to the actual instance of the question asked together with the marking and feedback information, as displayed on the right.

proportion of students obtained zero marks for Question 8. However, from this we cannot see whether the students obtained zero by not answering the question or by answering the question incorrectly. This further analysis could be performed by clicking on each 'zero' link but this would be cumbersome.

This task is facilitated in DEWIS by viewing performance indicators (PIs) as opposed to the mark scored. Each question is allocated at least one PI, which is either an integer or a string of integers, that indicates the performance of a student in a particular question. For all questions the standard PI contains at least a simple indication of whether the student's answer was correct, incorrect or not answered. Additional PIs can easily be created by the question author to supply more information about the performance of the student's answer. For example additional PIs can be used to indicate whether particular mal-rules were triggered in the marking process.

For the specific question being analysed here, which requires an integer answer, the standard PI takes one of the following values: 1 (correct), 0 (incorrect), -1 (not answered). The DEWIS reporter supports a *regular expression* search mechanism which allows the academic to display student attempts that satisfy a particular PI criteria. Figure 2 illustrates the resulting screen output for the PIs where we have included the search criteria of only including the assessments for which the performance flag for Question 8 has a value of zero. This corresponds to listing only the assessments for which an incorrect answer was supplied for Question 8, thus ignoring the correct and not answered responses. The displaying of PIs together with the search/filter facility facilitates the process of analysing why a student has answered a question incorrectly.

The next step in the process is to view some of these incorrect answers and to attempt to understand why that particular single question attempt was incorrect. Once we have identified a candidate reason for a student error we include such a check for this error in the question code. Typically a new PI is introduced for the question which takes the value of 1 if this error is triggered and 0 if not. All the e-Assessment submissions are re-marked automatically and, by viewing the value of this new PI, we can easily observe which student attempts triggered this new mal-rule.



Student	[Time][Date]	Q8
STUDENT_81	[13:09][18:10:2013]	0
STUDENT_81	[13:35][18:10:2013]	0
STUDENT_81	[22:20][18:10:2013]	0
STUDENT_6	[16:50][20:10:2013]	0
STUDENT_4	[11:53][22:10:2013]	0
STUDENT_42	[19:41][22:10:2013]	0
STUDENT_104	[16:20][23:10:2013]	0
STUDENT_54	[19:20][23:10:2013]	0
STUDENT_20	[16:52][24:10:2013]	0
STUDENT_99	[20:39][24:10:2013]	0

Figure 2: Display of the performance indicator data for Question 8, filtered to show incorrect responses only *(not all data shown)*.

For the 2013/14 academic year, Question 8 was presented to the student without any mal-rule detection. From the *Question* part of Figure 1, we can see that the question asks the student for the number of digits in the base b representation of a decimal number n, where in this instance the base is seven and the decimal number is 1672335. The value of n is chosen randomly to be a number containing between four and ten decimal digits. The value of base b is chosen to be between 3 and 9 but excluding 8 (the octal base). One intention of this question was for students to be able to evaluate the answer efficiently; a valid exercise for computing students.

Even in the case of $b = 3$, the answer to the question is not a big integer. However, it was initially surprising to note that some students were entering answers with a large number of digits, and thus misunderstanding what the question was asking for. This led us to suspect that the students were not reading and/or understanding the question correctly and that they may be entering the base b representation of n. It would have been inefficient for us to manually trawl through all the incorrect answers checking for this proposed mal-rule. One powerful feature of DEWIS is that we can alter the question code and mark retrospectively. In order to detect whether any students performed this incorrect base conversion, an additional PI was programmed into Question 8 which took the values shown in Table 1.

1	The student entered the base b representation of n.
0	Else.

Table 1: Values and explanations of the second performance indicator for Question 8.

3.1 Outcomes from the analysis

A re-mark was performed including this alteration and the results are displayed in Figure 3. Now, Question 8 has two PIs associated with it. The first value is the original PI value (1: correct, 0: incorrect, -1: not answered) and the second is as described in Table 1. In Figure 3 we have set the search settings so that only the attempts that trigger a second PI value of 1 are displayed. We see, from this data that, out of 329 submissions, only six students calculated the number n to base b and a snapshot of the suggested enhanced feedback provided in this case is shown on the right-hand side of Figure 3. It is interesting to note that the same student made this mistake on three occasions and it is hoped that this would not occur in future years due to students having access to the enhanced feedback immediately after submission.

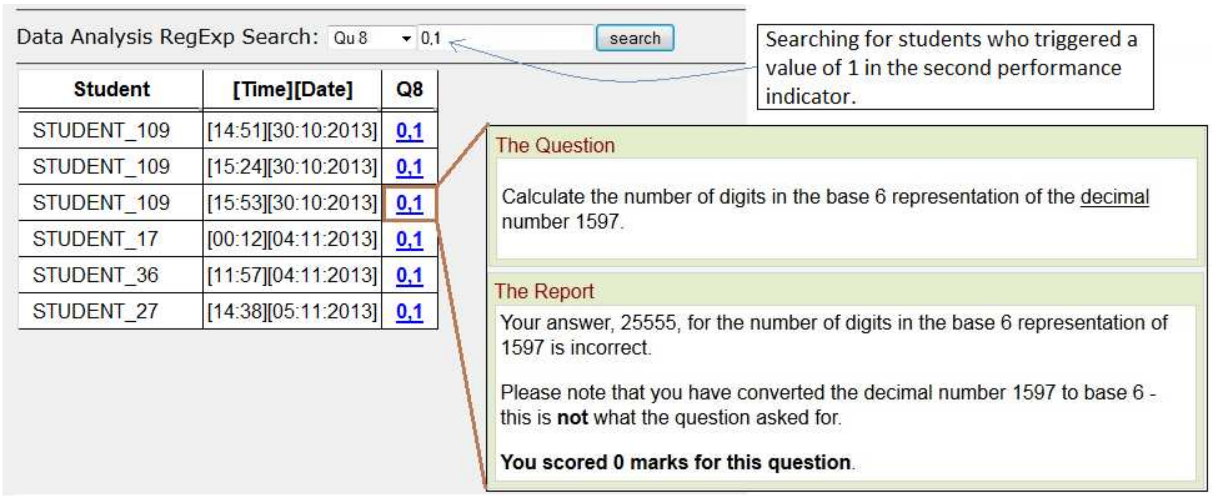


Figure 3: Display of the performance indicator data for question 8, filtered to show the second PI taking a value of 1. The right-hand side shows the proposed enhanced feedback supplied in this case, note that the worked solution has been omitted for brevity.

Further investigation revealed that two students had evaluated the base 10 representation of nb. For the remaining 25 attempts that entered an excessive number of digits, it was not possible to determine exactly what mistake the student had made. Some may have attempted to evaluate the decimal n to base b but simply failed in their attempt. A complete list of the mal-rules detected for this question is illustrated in Table 2. It was not possible to explain the mistake in 13 of the 81 incorrect attempts.

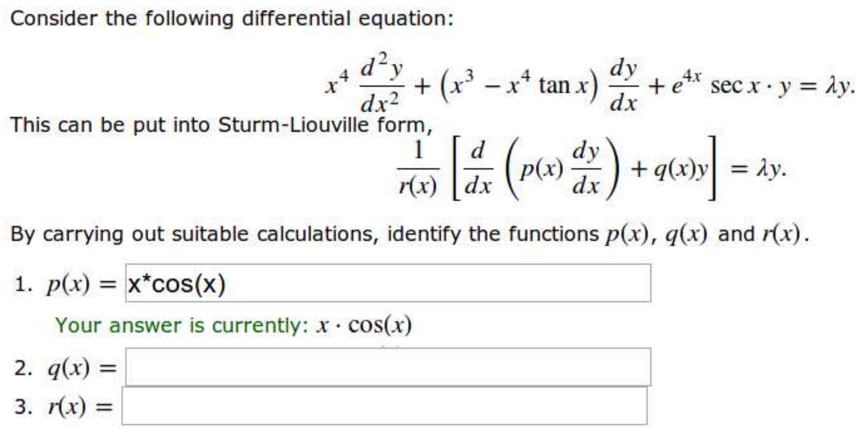
Including mal-rule detection has very little additional computational overhead, hence, for future uses of this question, we will search for the mal-rules listed in Table 2 with any CSE detection being reflected in enhanced feedback provided to the student.

1	evaluating n to base b	6
2	evaluating nb in decimal	2
3	entering an excessively large number	25
4	entering $\texttt{floor}(\log_b n)$	14
5	entering $\texttt{floor}(\log_b n)$-1	7
6	entering $\texttt{ceil}(\log_b n)+1$	14

Table 2: Mal-rule analysis for Question 8: performance indicator flags, descriptions and counts.

4 Case Study 2

In this section, we shall illustrate how the detection algorithm was applied on an e-Assessment which was run with second-year mathematics students at Leeds University. The syllabus included Sturm-Liouville operators and the question that we are going to consider here required students to find three functions, denoted by p, q and r, from a given differential equation, and to input these functions in algebraic form. The question was constructed by choosing parameters randomly for


Consider the following differential equation:

$$x^4 \frac{d^2y}{dx^2} + \left(x^3 - x^4 \tan x\right) \frac{dy}{dx} + e^{4x} \sec x \cdot y = \lambda y.$$

This can be put into Sturm-Liouville form,

$$\frac{1}{r(x)} \left[\frac{d}{dx} \left(p(x) \frac{dy}{dx} \right) + q(x)y \right] = \lambda y.$$

By carrying out suitable calculations, identify the functions $p(x)$, $q(x)$ and $r(x)$.

1. $p(x) =$ x*cos(x)

   Your answer is currently: $x \cdot \cos(x)$
2. $q(x) =$
3. $r(x) =$

Figure 4: A realisation of the Sturm-Liouville question asked, with the answer for $p(x)$ filled in by the user, for illustration purposes.

p, q and r, and using the functions so generated to construct a differential equation of the form $a(x)\frac{d^2y}{dx^2} + b(x)\frac{dy}{dx} + c(x)y = \lambda y$. A realisation of the question is shown in Figure 4.

The question was run as a live assessment during the module, and then re-opened as an opportunity for practice during the revision period. In the 2013/14 academic year, there were 563 student attempts, of which 197 attempts had input which was marked as algebraically valid, that is the expression entered was a well-formed function, but incorrect.

Algebraic student function input is one of the more complicated forms of input an e-assessment system is required to handle, and provides some of the greatest scope for student errors to arise. It is therefore beneficial to provide students with feedback which relates directly to the answers they have given which identifies potential sources of error. The DEWIS e-assessment system is able to mark a student inputted algebraic entry explicitly against a test function in order to determine whether the two match. The most obvious use of this facility is to mark the student answer against the correct answer; however, it can also be used to mark against potential mal-rules.

For the case of the question requiring one algebraic input, the marking algorithm that incorporates CSE detection will consist of three inputs: the student's answer, the correct answer and a lookup-table containing a list of mal-rules. Both the correct answer and the lookup-table are dependent on the question parameters which construct the question. Without CSE detection there is only one performance indicator (PI) associated with the marking process, in which case the student input is simply compared to the correct answer. In this case the PI can take any one of four integer values as shown in Table 3.

1	when the two functions match
0	when the two functions do not match
-1	if the question is not answered
-2	when the student answer is not a well-formed function

Table 3: Standard PI values for algebraic inputs

With CSE detection, there will be a PI corresponding to all the mal-rule entries in the lookup-table in addition to the PI corresponding to the correct answer. The marking process first compares the student answer with the correct answer and the PI associated with the correct answer is populated accordingly. If this first PI value is zero (the student's answer is a valid function but is an incorrect answer), then the student's answer is marked against all the entries in the mal-rule lookup-table, resulting in a string of PIs which can be easily viewed in the DEWIS Reporter.

In the question being analysed here, there are three inputs, and an error could potentially affect anywhere between one and all three inputs. As a consequence, the data structures used in the actual marking and CSE detection code were more complicated than described above. In particular, the lookup table containing mal-rule answers contained a key and an array with its effect on each of the three inputs in turn.

Prior to running the algorithm on the data, four potential student errors were identified by thinking about the structure of the question, and another four were identified by inspecting a few student attempts. Not all student attempts could be explained, as some of them were valid *chatter* (such as "`0,0,0`" or "`x,x,x`") supplied, presumably, in order to receive feedback which took the form of a fully worked solution to the question.

After running the algorithm, some of the remaining unexplained answers were inspected in order to glean new mal-rules; these mal-rules were then coded into the system and the algorithm re-run. This process continued iteratively for some 16 runs of the algorithm. By this point, after various amalgamations of equivalent errors, the lookup table contained 20 separate potential sources of error divided into 17 categories. Table 4 gives a complete list of the mal-rules considered, together with a description of each and a count of the number of occurrences.

1a	Missing the denominator of $a(x)$ in $q(x)$	11
1b	Missing the denominator of $a(x)$ in $r(x)$	17
2	Reading off coefficients from the initial equation	3
3	Using $\exp(A+B) = \exp(A) + \exp(B)$ (error in p)	19
4	Using $\int^x \tan(x)dx = \log\sin(x)$ or $\int^x \cot(x)dx = \log\cos(x)$ (error in p)	0
5	Using $x^{-n} = -x^n$ for $n > 0$ (error in p)	5
6	Using $\int^x \tan x\, dx = \frac{1}{\sin x}$ and likewise for cot (error in p)	5
7	Using $\int^x a\tan(ax)\, dx = a\log\cos(ax)$	5
8	Thinking $r = (ap)^{-1}$ (error in r)	3
9	Thinking $r = \frac{a}{p}$ (error in r)	6
10	Thinking $r = ap$ (error in r)	10
11	Thinking $p = \exp(\frac{b}{a})$ (error in p)	1
12a	Out by a minus sign (error in p)	1
12b	Out by a minus sign (error in q)	1
12c	Out by a minus sign (error in r)	3
13	Using $\exp(\int^x x^{-1}\, dx) = x^{-1}$ (error in p)	9
14	Thinking $q = \frac{r}{c}$ (error in q)	3
15	Swapped q and r	2
16	Thinking $q = \frac{c}{r}$ (error in q)	6
17	Using $(x^n)/(x^m) = x^{n+m}$ (error in r)	6

Table 4: Mal-rule analysis for the Sturm-Liouville question: performance indicator flags, descriptions and counts. Individual flags counted across 87 student attempts.

One extra enhancement, realised early in the process, was to use the marking and CSE detection algorithm as a means of awarding continuation marking relatively simply. In this question, two of the functions can be calculated from simple, linear equations involving the other. Therefore, many of the calculation errors propagate in a highly predictable way. The mechanism for testing against carry-through errors is similar to the mechanism for testing against mal-rules for algebraic inputs described earlier, though with a look-up table of candidate functions created dynamically in response to the student's input. This allows for continuation marking to be implemented for future runs of this question, as well as providing another source of valuable feedback for students. This also simplifies greatly the algorithm for mal-rule detection since it separates identification of errors from propagation of errors.

The conclusion of the iterations detailed above was that of 197 incorrect attempts, 87 were explained by the various rules identified above, and a further 42 had some other, unidentified error which continued through the student answer.

The feedback to the student, which initially consisted of a worked solution, has been improved by the addition of a section which appears if the student's answers trigger one of the flags. For example, flag 14 raises the prompt, "You might have thought that $q(x) = \frac{r(x)}{c(x)}$. In fact, $q(x) = c(x)\, r(x)$." It was constructed with a separate lookup table containing prompts that relate to each mal-rule discovered. For each of the mal-rules which was triggered, the relevant line of feedback can be displayed to the student. The continuation flags can be used similarly.

4.1 Outcomes from the analysis

A proportion of students' attempts included more than one error or potential error: at least 24 student attempts raised more than one flag, although this is undoubtedly an underestimate of the number of combined errors since the algorithm can detect at most one error per input, so student inputs containing more than one error would be missed entirely. This is an area which requires mathematical study as well as technical development, since a calculation contains several steps, and interchanging steps or introducing errors at different stages in different orders could easily result in different final answers.

One of the key features to arise from the analysis of the student data was the occurrence of errors due to relatively basic mathematical mistakes, also reported by Jordan (2007). Errors in integration and common identities involving exponential indices were not infrequent: for example, the mal-rule $\exp(A+B) = \exp(A) + \exp(B)$ was identified in at least 19 student attempts, and at least 10 student attempts included some kind of failure to integrate $\tan(ax)$. The errors associated with flags 3-7 and 17 were all errors of basic mathematics: 39 attempts triggered at least one of these flags.

With the exception of flags 12a–c, the remainder of the mal-rules correspond to errors in the formal syllabus material, although in some cases (*e.g.* flag 2) one could not tell whether a student misunderstood the material or simply copied the question's functions as a form of advanced chatter. There were 52 attempts which raised at least one of these flags. Five attempts raised both a 'basic mathematics' and a 'syllabus material' flag.

Some of the mal-rules, though logically independent, were conceptually very strongly linked. An example of such an error is in the calculation of the functions q and r, having found p (flags 9 and 16). The correct formulae to use are $a = \frac{p}{r}$ and $c = \frac{q}{r}$; any student thinking one of $p = \frac{a}{r}$ or $q = \frac{c}{r}$ invariably thought the other as well. These errors were not logically equivalent, but could have a common cause in a confusion between the differential equation and its Sturm-Liouville form.

5 Discussion

We have shown a process of analysis of post-submission e-Assessment question and answer data that allows for the detection of previously unsuspected student errors. The process takes advantage of the fact that the e-Assessment system used is fully algorithmic and has lossless-data collection which allows for retrospective marking. The process is time efficient and allows for an evaluation of the e-Assessment resulting in improved feedback and thus improves the student experience of e-Assessment.

In the case studies considered we found several mal-rules which were triggered on only one or two occasions. It would be interesting to see whether these mal-rules are triggered in the future by monitoring the use of the e-Assessment tests over the coming years. This finding ties in with the work of Payne and Squibb (1990) who found that most mal-rules occur very infrequently.

The focus of this article was on the process of detecting and reporting CSEs, as opposed to providing a comprehensive study of mal-rules themselves. However, identifying and classifying mal-rules is an area which has received sporadic attention and more information can always be used. Building on existing taxonomies of errors (Haynes and Herman, 2014) would further facilitate this process. Considering mal-rule combinations is a rich area for future work, as it raises interesting mathematical questions, as well as being a clear technical challenge.

Philip Walker is currently a Teaching Fellow in the School of Mathematics, Leeds University, Leeds, United Kingdom.

D. Rhys Gwynllyw is currently a Senior Lecturer in the Department of Engineering Design and Mathematics at the University of the West of England, Bristol, United Kingdom.

Karen L Henderson is currently a Senior Lecturer in the Department of Engineering Design and Mathematics at the University of the West of England, Bristol, United Kingdom.

References

Broughton, S. J., Robinson, C. L. and Hernandez-Martinez, P. (2013). Lecturers perspectives on the use of a mathematics-based computer-aided assessment system, *Teaching Mathematics and its Applications* **32**(2): 88–94.

Bull, J. and McKenna, C. (2003). *Blueprint for Computer-Assisted Assessment*, Routledge, London.

Foster, B., Perfect, C. and Youd, A. (2012). A completely client-side approach to e-assessment and e-learning of mathematics and statistics, *International Journal of e-Assessment* **2**(2).

Gill, M. and Greenhow, M. (2007). Computer-aided assessment in mechanics: question design and test evaluation, *Teaching Mathematics and its Applications* **26**(3): 124–133.

Glendinning, P. (2008). View from the Pennines: Beyond mal-rules, *Mathematics Today* **44**: 200–202.

Greenhow, M. and Gill, M. (2008). How effective is feedback in computer-aided assessments?, *Learning, Media and Technology* **33**(3): 207–220.

Greenhow, M. and Kamavi, K. (2012). Maths e.g. - a web assessment application for STEM and beyond, *Proceedings of the HEA STEM Learning and Teaching Conference.*

Gwynllyw, R. and Henderson, K. (2009). DEWIS: a computer aided assessment system for mathematics and statistics, *CETL-MSOR 2008 Conference Proceedings*, pp. 38–44.

Gwynllyw, R. and Henderson, K. (2012). Intelligent marking in summative e-assessments, *Proceedings of the HEA STEM Learning and Teaching Conference.*

Haynes, L. and Herman, D. (2014). An investigation into common student errors in first year calculus and algebra. In preparation.

Jordan, S. (2007). The mathematical mis-conceptions of adult distance-learning science students, *CETL-MSOR 2006 Conference Proceedings*, pp. 87–92.

McCabe, M. (2009). The exponential growth of mathematics and technology at the University of Portsmouth, *Teaching Mathematics and its Applications* **28**(4): 222–227.

Payne, S. J. and Squibb, H. R. (1990). Algebra mal-rules and cognitive accounts of error, *Cognitive Science* **14**: 445–481.

Sangwin, C. (2004). Assessing mathematics automatically using computer algebra and the internet, *Teaching Mathematics and its Applications* **23**(1): 1–14.

Sangwin, C. (2013). *Computer Aided Assessment of Mathematics*, Oxford University Press.

Seely Brown, J. and Burton, R. J. (1978). Diagnostic models for procedural bugs in basic mathematical skills, *Cognitive Science* **2**: 155–192.