



Possibilities of OpalExam to improve exam assessment and viewing

M. Fiedler, M. Kästner

Professorship for Numerical and Experimental Solid Mechanics, Institute for Solid Mechanics, Faculty of Mechanical Engineering, TU Dresden

Abstract

Durch die Umstellung von Präsenz- auf Onlineklausuren stehen Lehrende vor neuen Herausforderungen. Dabei geht es nicht alleine nur um die technische Herausforderung, sondern auch um den eigenen Anspruch, Klausuren online so zu modellieren, dass Betrugsversuche entweder verhindert werden können oder dem Betrügenden keine Vor- sondern Nachteile bieten. Lehrenden stehen dabei während der Klausurmodellierung in OpalExam verschiedene Möglichkeiten zur Verfügung: Definition von Zufallswerten durch Verwendung von Variablen in der Aufgabenstellung, Erstellung mehrerer Varianten einer Aufgabe, die den Studierenden zufällig zugewiesen werden oder auch Änderung der Aufgabenreihenfolge für die Studierenden und Definition eines linearen Durchlaufs durch die Klausur. Die einzelnen Optionen können dabei natürlich auch kombiniert werden.

Wird ein nichtlinearer Verlauf (Studierende können zwischen den Aufgaben beliebig hin- und herspringen) durch die Klausur angestrebt, so werden die Klausuren durch die Wahl der ersten beiden Optionen komplexer. Für Studierende wird dadurch zwar die Zusammenarbeit erschwert, die Bewertung der Antworten des Studierenden durch den Prüfenden allerdings auch. Innerhalb dieses Aufsatzes werden deshalb Möglichkeiten aufgezeigt, wie der Bewertungsaufwand gesenkt und dabei gleichzeitig die Klausureinsicht erleichtert werden kann.

The switch from offline to online exams presents teachers with new challenges. Not all challenges are technically, a challenge is also the own demand to model online exams so that cheating attempts can either be prevented or do not offer advantages but disadvantages to the cheater. Teachers have various options at their disposal when modelling exams in OpalExam: Definition of random values by using variables in the assignment, creation of several variants of an assignment that are randomly assigned to the students or also change of the assignment order for the students and definition of a linear run through the exam. Of course the individual options can also be combined.

If a non-linear progression (students can jump back and forth between tasks at will) through the exam is the goal, the exams become more complex by choosing the first two options. While this makes it more difficult for students to collaborate, it also makes it more difficult for the examiner to evaluate the student's answers. In this paper, it will be shown how the assessment effort can be reduced while at the same time making it easier for the students to review the exam results.

*Corresponding author: melanie.fiedler@tu-dresden.de

This article was originally submitted in German.

1. introduction

Switching from face-to-face to online examinations presents teachers with new problems and challenges. In addition to the restructuring of assignments and technical problems and challenges that arise, see e.g. [1, 2], attempted cheating is a serious problem, cf. [3, 4, 5]. In order to prevent cheating through collaboration without online proctoring, assignments can be individually tailored to students, but this can significantly increase the effort required for assignment creation, assessment and viewing. In this regard, Onyx in OpalExam includes several features that can simplify exam marking and review. Section 2 gives an overview of current possibilities to avoid cheating during exam creation in OpalExam. Section 3 discusses the advantages and disadvantages of the resulting more complex modelling. The documentation of the achieved points per answer for tasks with multiple task fields is not yet available as a setting in Opal. Nevertheless, this can be realised is described in section 4. Awarding partial points to answers that do not correspond to the correct solution, e.g. because a term was forgotten in the calculation, is only possible in Opal when using question type "formula comparison". A documentation of the necessary programming is given in section 5. The consideration of consequential errors in longer calculation tasks is also possible with question type "formula comparison", described in section 6. Finally, a summary of the possibilities described here follows in section 7.

2. Ways to avoid fraud attempts

When using the test function in Opal or OpalExam, assignments and tests can be modelled in Onyx and made available to students online. If examinations are conducted online, collaboration between students cannot be ruled out. This could take the form of several students theoretically sitting next to each other with their laptops or even being connected via video chat. Control by the teacher is difficult or even impossible, especially in the latter scenario.

The only option is to model exams so that the result of collaboration between students leads

to large time losses and thus becomes less attractive to students.

There are several ways to do this:

- The use of random values in the task by defining variables,
- the creation of several task variants,
- changes in the order of tasks and setting a linear progression through the exam.

If only the task order of an exam is changed, students cannot work on the task together via video chat, but they are still able to exchange results with each other if they all have the same numerical values, questions or pictures. A simple screenshot can be exchanged very easily. In addition, a big difference to presence is that students can no longer switch back and forth between individual tasks at will by setting the linear progression. If a non-linear progression is set instead, so that students can switch between tasks at will, the effect of changing the sequence of tasks is almost cosmetic.

An improvement can be achieved by using random values in the task. For this, the given variables, e.g. length a , are assigned a random number. This is done by introducing variables that are automatically assigned when the task is called up by the students and integrated into the task text, see Fig. 1. These variables are referred to in the following as pre-submission variables, since their values are already fixed for the students before the task is submitted and can no longer be changed.

For these pre-submission variables, e.g. number intervals with step widths to be defined can be set in the "Variables" tab of the task, Fig. 2.

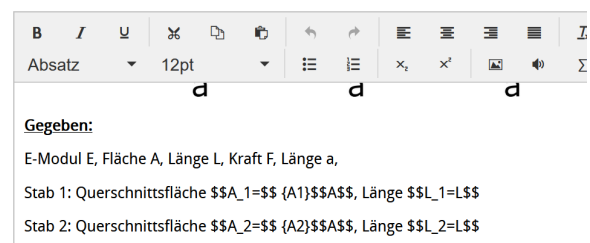


Fig. 1: Inclusion of pre-submission variables in the task text

The students get automatically assigned numerical values that differ from those of their fellow students when the task is called up. In

this way, at least the copying of numerical values can be avoided.

Fig. 2: Definition of an interval of random values for a pre-submission variable

What is still possible, of course, is sharing the solution path among several students. To prevent this, different versions of a task can be developed, see Fig. 3. This option is comparable to the output of A and B exam versions in offline exams, if the necessary space between two students could not be maintained in the past.

Due to minor changes in the assignment, a 1:1 adoption of the fellow students' solution is no longer easily possible. Instead, orientation on another students solving for a different exam version can lead to loss of time and points if the student has to find the differences or overlooks them.

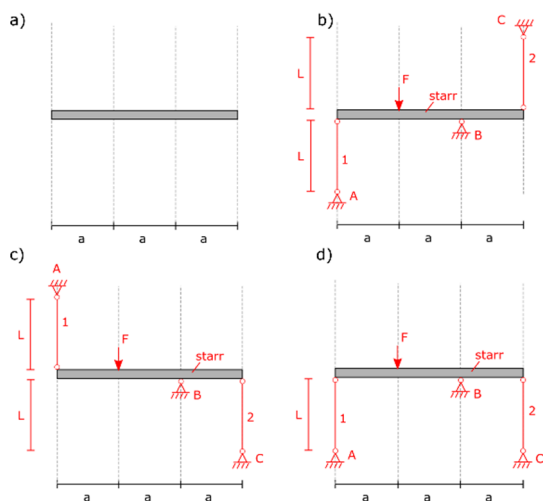


Fig. 3: Representation of different variants for an assignment: a) Examination template for the students, b)-d) Variants of the assignment.

3. Advantages and disadvantages of complex modelling

A combination of random values and task variants thus makes copying and collaboration less attractive for students. However, for the modeller of the exam, this also increases the modelling effort and thus the complexity significantly. In addition, there may be time differences in the assessment of the exam when the automatic evaluation is checked. In contrast to an attendance exam, correctors are no longer able to recognise the error in the solution at first glance based on the result alone, since each student receives their own numerical values and have different task variants.

Modellers therefore have two options: Either to model the exam tasks and their result queries so closely that a re-assessment or awarding of partial points by correctors is not necessary. However, with a close-meshed result query, there is quickly the danger of already having to specify a solution path, which lowers the degree of difficulty of the exam.

Or to have the examination assessment in mind already during the modelling of the examination and create the tasks so that an allocation of partial points, a consideration of consequential errors and a documentation of achieved points per answer and not only per question is automatically possible. The second variant leads to a faster, because clearer post-assessment and fewer questions from students during the subsequent examination.

4. Display of achieved points per answer for tasks with several answer fields

Within the exam modelling, tasks can be created where students have to give more than one answer, e.g. "numerical input interaction" with several gaps or a "choice interaction" with several choices. However, the Opal evaluation only shows the total score of the task and the information right/wrong for each answer. The information on how many points were scored per answer is missing.

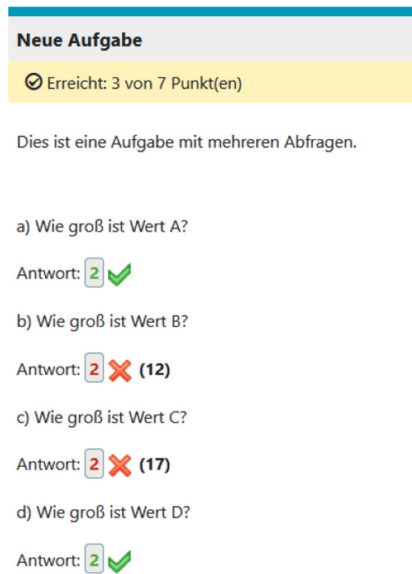


Fig. 4: Display of the evaluation of a task with several gaps. Only the total score is displayed at the top, [6].

If all answers have the same level of difficulty and each gap is assigned the same number of points, this is not a problem in the evaluation. If, on the other hand, the individual answers are to be weighted differently, there is no indication of the points achieved for each answer. This leads to unnecessary difficulties when correcting the automatic evaluation and viewing the exam, see Fig. 4. Therefore it makes sense to display the number of points achieved after each answer.

Post-submission variables can be used to display points achieved per answer, cf. [6]. These are integrated into the task text in the same way as pre-submission variables, but only assigned values after the task has been submitted. Post-submission variables do not appear for students during the exam and only become visible when the exam results are viewed. In addition to displaying the points achieved per answer, post-submission variables can also be used to display explicit information for the scorers or for exam review.

The post-submission variable is defined in the "Variables" tab below the pre-submission variable. Post-submission variables can also be assigned text, fixed values or variable values. Unlike pre-submission variables, post-submission variables can also access the programme-in-

ternal variables SCORE, MAXSCORE, MINSCORE, LEARNERRESPONSE, etc.

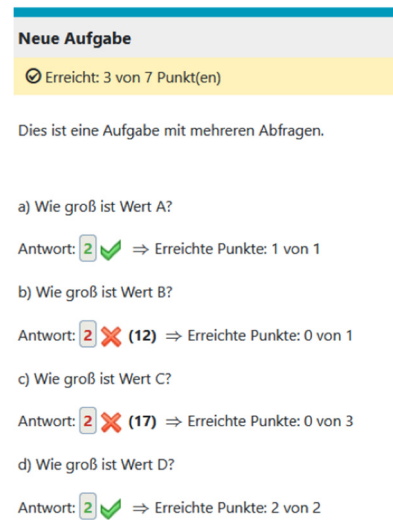


Fig. 5: Extension of the task by post-submission variables to display the achieved points per gap, [6].

If, for example, the post-submission variable "{1Score}" is defined and set equal to the internal programme variable "SCORE_GAP_1", this variable contains the score achieved by the student after answering gap 1. If this variable is included in the task text, the score achieved by the student appears in the task text after submission, see Fig. 5.

In [6] a step-by-step guide on how to use the post-submission variable to display achieved points per gap is documented.

5. Award of partial points

An extension of the scoring for correct and incorrect answers by partial points can also be realised during the examination modelling. The basis for this is question type "formula comparison", cf. [7]. The maximum number of points for the gap is still defined via the setting option "Points" of the gap. The correct answer is also still defined via the "Solution" field.

In order to award partial points, the gap type "Formula" must be used in expert mode, which enables the possibility of programming an if-else condition. The evaluation criterion must be changed from "True/False" to "Points", see Fig. 6.

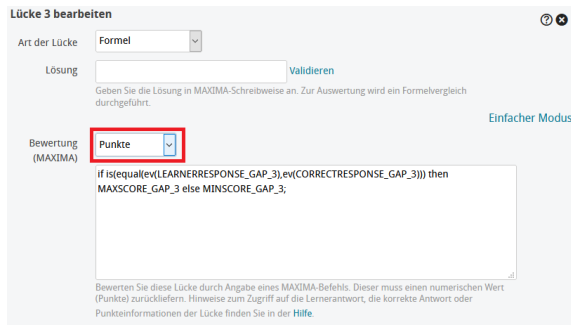


Fig. 6: Captions in Open Sans, 9pt, italic, left-aligned. 4pt spacing before caption text, [7].

The default is an if-else condition that checks whether the student answer (LEARNERRESPONSE) corresponds to the correct answer (CORRECTRESPONSE). If this is the case, the student receives the maximum score (MAXSCORE), otherwise the minimum score (MINSKORE), i.e. 0.

In order to be able to award partial points in addition to the maximum and minimum scores, the if-else condition must be extended by an additional condition. For this purpose, another if-else condition is modelled, which checks whether the student has, for example, a deviation of 1 from the correct answer. If this is the case, the student still receives 1/3 of the maximum score, see Figure 7.

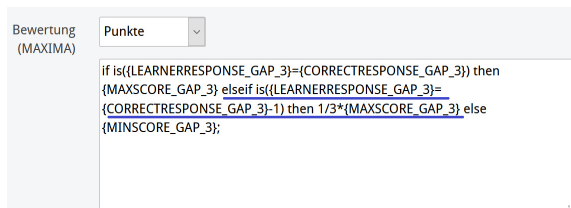


Fig. 7: Extension of the programming by another if-condition (marked blue)

If the full number of points for the task is no longer awarded to the student, Opal indicates this in the evaluation by awarding a yellow tick, see Figure 8.

Since 30.03.2021, the green and yellow ticks as well as the red crosses are also displayed in the assessment PDF when exporting via data archiving, [8]. The display can thus also be used for exam assessment and viewing.

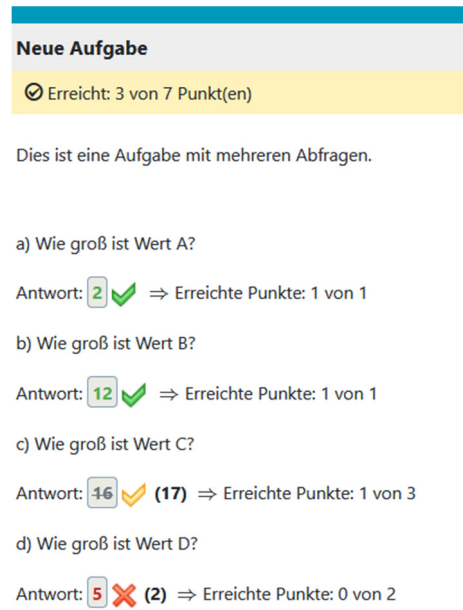


Fig. 8: Representation of the evaluation of correct, incorrect and partial points, [7].

6. Consideration of consequential errors

In the evaluation of presence examinations, the evaluation of consequential errors accounts for a large proportion of the examination evaluation time. In order to reduce this proportion, the automatic examination evaluation can be used. This makes sense especially for exams with a large number of participants. If it is to be possible to evaluate consequential errors within a task, there are various ways how this can be realised. A discussion of this can be found in [9].

For a non-linear examination modelling, the question type "formula comparison" is used here. Within the expert mode, the option "Points" is still selected.

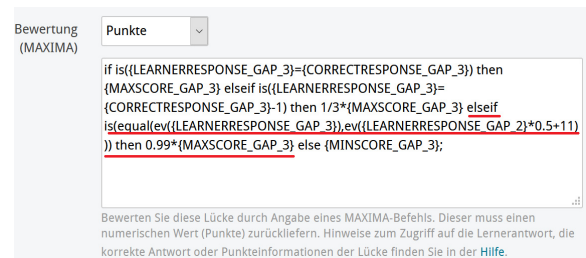


Fig. 9: Addition of another if-condition to the Maxima programming (underlined in red)

The extended if-else condition defined in section 5 is extended by another if-else condition to take into account a consequential error from gap 2. In the case of a wrong solution, this condition asks whether the wrong solution has arisen as a consequential error. If yes, points can also be given based on this, if no, the student receives 0 points.



Fig. 10: Automatic evaluation with consideration of consequential errors

In Fig. 10, 1% of the achievable score of the gap was deducted for a subsequent error. Due to the small deduction, the gap was evaluated as partial points, cf. section 4, and marked for the corrector. The size of the deduction can be freely chosen when modelling the task.

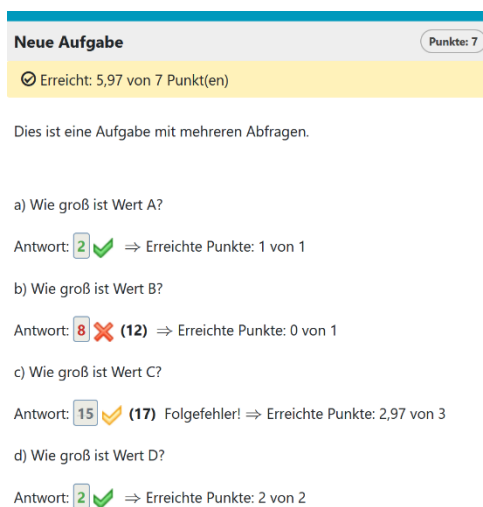


Fig. 11: Automatic evaluation with assessment and separate indication of subsequent errors

If it should become even clearer that a subsequent error was evaluated here, a post-submission variable {3FF} can be defined, which in the case of a subsequent error takes the text value "Subsequent error!" and otherwise the value "", Fig. 11.

To do this, an if-condition must be created for the post-submission variable, see Fig. 12. Afterwards, the post-submission variable can be included in the task text in the same way as the other post-submission variables.

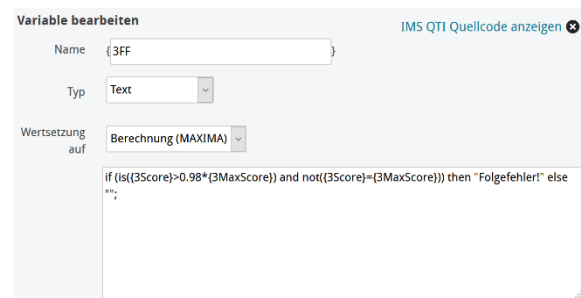


Fig. 12: Creation of the consequential error text variables for gap 3

7. Summary

The conversion from offline to online exams holds some hurdles. To minimise cheating attempts, transcription and collaboration must be made unattractive. This goes hand in hand with an increase in the complexity of task modelling. In order not to cause an increase in the assessment effort, post-submission variables can already be used during the task modelling or formal gaps can be programmed.

It should be mentioned that this increases the time needed for modelling and checking the tasks. However, if the use of digital exams is still planned in the future, e.g. in combination with an offline examination [9], the modelling effort can be worthwhile if similar tasks can be generated from the existing ones in the subsequent examination period through minor adjustments. Therefore, this can create a pool of tasks that specifically and permanently minimises the examination assessment time.

Literature

- [1] Schlecht, B.; Rosenlöcher, T.: Möglichkeiten und Grenzen der digitalen Lehre und Prüfung im Grundlagenfach Maschinenelemente; Book of Abstracts

- der 1. Lessons Learned Konferenz; 14.+15.10.2020 in Dresden; TU Dresden; 2020
- [2] Dohmen, Eike.; Lange, Adrian; Odenbach, Stefan: Online-Prüfung mit OPAL und ONYX – Wieviel besser sind 8 Monate?; Book of Abstracts der 2. Lessons Learned Konferenz; 18.+19.03.2021 in Dresden; TU Dresden; 2021
- [3] Schmidt, Th.: Morgen ist das heute von gestern, oder: Wie laufend alles anders kam; Book of Abstracts der 1. Lessons Learned Konferenz; 14.+15.10.2020 in Dresden; TU Dresden; 2020
- [4] Bilen, Eren; Matros, Alexander.: Online cheating amid Covid-19; Journal of Economic Behavior and Organization; Vol. 182; p. 196-211; 2021
- [5] Abdelrahim, Yousif; How Covid-19 quarantine influenced online exam cheating: A case of Bangladesh university students; Journal of Southwest Jiaotong University; Vol. 56; 2021
- [6] Fiedler, Melanie: Verwendung von Nach-Abgabe-Variablen zur Anzeige der Punktzahlen jeder Lücke in Lückentextaufgaben; <https://tud.link/2m1r>; Stand: 31.05.2021;
- [7] Fiedler, Melanie: Vergabe von Teilpunkten in Lückentextaufgaben; <https://tud.link/sbuw>; Stand: 31.05.2021
- [8] BPS Bildungsportal Sachsen GmbH: BPS Release Notes; <https://www.bps-system.de/help/display/OR/ONYX+Testsuite+9.9>, Stand: 30.05.2021
- [9] Fiedler, Melanie; Kästner, Markus: Rechnerische Klausuren in OpalExam unter Berücksichtigung von Folgefehlern modellieren – Vorteile, Nachteile und Zukunftsaussichten; Book of Abstracts der 2. Lessons Learned Konferenz; 18.+19.03.2021 in Dresden; TU Dresden; 2021