Sophie Kröger, Paul Schwarz
Using STACK with Moodle in the Physics Laboratory to review the results of student experiments and their evaluation
In: Contributions to the 3rd International STACK Conference 2020.
TTK University of Applied Sciences: Tallinn, Estoni. DOI: 10.5281/zenodo.4415620

Using STACK with Moodle in the Physics Laboratory to review the results of student experiments and their evaluation

Sophie Kröger[1] and Paul Schwarz[2]

Abstract: In the physics lab at the HTW we use STACK with Moodle for the review of the results of laboratory experiments. The entire evaluation of the experimental results (including the expression of uncertainties) is checked by means of STACK queries. Very comprehensive queries are summarized in amalgamated Moodle tasks. In particular, the verification of the number of significant digits in combination with an automated, differentiated feedback for students represents a big challenge in this context.

Keywords: STACK, Moodle, Physics Laboratory, Error calculation

1 Introduction

STACK with moodle offers possibilities for computer aided assessment packages. The standard application can be found in the field of mathematics, where STACK asks for mathematical expressions and evaluates these expressions using computer algebra systems. But there are also good uses for STACK in other disciplines, especially in natural sciences and technology.

We are part of the physics team in the department of electrical engineering at the HTW Berlin (University of Applied Science Berlin). The STACK extension for the learning platform moodle was implemented at our university about three years ago. We immediately noticed, that this tool has a lot of potential to save teaching staff time when checking the work of the students in our lab. Shortly afterwards, together with two student assistants, we began to program the evaluation of the laboratory exercises using STACK queries.

We, the physics team, use STACK with moodle also for other applications such as i.e. a self-learning physics course for exam preparation. But in the present paper we limit ourselves to presenting the application in the laboratory. We are not aware of similar approaches by other experts in this area.

[1] Hochschule für Technik und Wirtschaft, Fachbereich 1, Wilhelminenhofstraße 75A, D-12459 Berlin, sophie.kroeger@htw-berlin.de, https://orcid.org/0000-0003-4991-9176

[2] Hochschule für Technik und Wirtschaft, Fachbereich 1, Wilhelminenhofstraße 75A, D-12459 Berlin, paul.schwarz@htw-berlin.de

2 Scope of Application, Special Requirements

The HTW Berlin has about 14000 students and offers about 70 courses in the fields of business, engineering, computer science, design and culture. About a quarter of all students are going through our physics laboratory, around 800 students per year or 400 per semester, respectively. In the physics laboratory we work with groups with a maximum of 24 students. That results in about 18 parallel courses per semester. The lessons are carried out by full-time professors and part-time lecturers. These explanations serve to show that the programming effort is worthwhile here. Before the implementation of STACK, the students' results have been checked 'by hand', depending on the capacity sometimes only on a random basis. This was a very time-consuming task for the teachers.

In our physics laboratory, great importance is attached to error analysis. This includes error estimation when recording the measurement data and error propagation when evaluating the measurement data. In particular, students should learn how experimental values should be given with the correct number of significant digits.

Compared to the type of STACK tasks that are often used in mathematics or physics, there were some special requirements for our applications:

- The verification of the results is not based on specified values or on random numbers, but on measurement data that the students measured beforehand.
- The specification of the required number of significant digits depends on the inputs made previously by the students and may be different.
- Comprehensive evaluations cannot be divided into several STACK queries because the entered data cannot be transferred from one query to a following query.

The first point is self-explanatory, since it is the evaluation of measurement data that the students measured themselves.

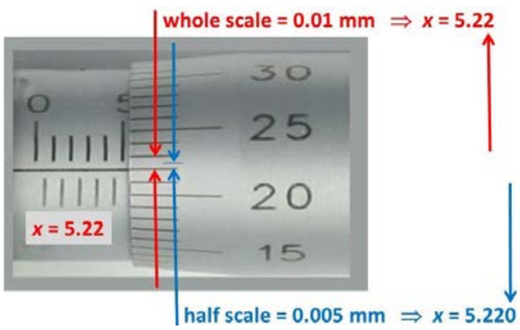

Fig. 1: Scale of an micrometer

As a short example to explain the second point in, Fig. 1 a simple measurement with the micrometer is presented. As in all of our experiments, here it is the individual decision of the student how great the reading accuracy is. It could be a half or a whole scale, maybe even two. The measurement error would be 0.01 mm for a whole scale division and 0.005 mm for a half scale division (see Fig. 1). For a reading accuracy of 0.01 mm, x should be given as 5.22 mm; for a reading accuracy of 0.005 mm, x should be given as 5.220 mm. Depending on which value is selected as reading accuracy, the measured value must be given up to the second or third decimal place. The corresponding STACK query must check this depending on the student's decision.

Additionally, in our Lab the following rule should be applied to values, which are calculated from primary measurement data [Kr19]:

- An error bar should be given with only one significant digit unless this digit is a '1' or a '2'. If the first digit is a '1' or a '2', two significant digits should be given.
- The value belonging to the error bar should be given up to the same decimal place as the (last) significant digit of the error bar.

These conditions also make the required number of significant digits dependent on the inputs made previously by the students.

3 Implementation and Examples

We have implemented STACK queries in our learning platform moodle for all of our standard experiments that are performed by all students. After carrying out the experiment, the students enter their measured values in the moodle STACK query together with the corresponding reading accuracies. If appropriate, systematic errors of the measuring devices are also queried. Now the students evaluate their measured values and calculate intermediate results and final results including error propagation. All these results are submitted to the STACK query also. By clicking the 'Check' button, the students may verify if their results are correct. It is allowed for the Students to check the results as often as they want before the deadline for submission of the test. This should enable the students to test themselves whether their results are correct. Differentiated feedback from the STACK query should help them to identify autonomously the source of mistakes and to arrive at the right result. Of course, this does not replace personal care by the teaching staff, but it simplifies the work.

When implementing the STACK for our task, we went through different iterations. In the first step we used complicated self-programmed approaches for checking the number of significant figures. In the next step we got to know the verification function NumSigFigs [Github1]. That made the job a little easier. But checking whether the rules for the correct number of significant numbers of error bars and corresponding values (as given above) are applied correctly remained 'manual programming work'.

Since this is one of the key points for our application, we show a small section of the code for this query here:

```
x : ans-errorbar;
n : 0;
a : if (x >= 10) then (for i :0 thru 20 do (if (x>=10)
then (x : x/10 ,n : n+1 ))) else if (x<1) then (for i:0
thru 20 do ( if (x<1) then ( x : x*10 , n : n-1 )));
b : if ( floor(x) = 1  or  floor(x+0.05) =2 ) then (z :
2) else (z : 1);
```

Here, `ans-errorbar` is the value of the error bar. The decimal point is shifted so far that there is only a one-digit number before the decimal point. This one-digit number is then checked to see whether it is a '1' or '2'. The resulting numbers `z` and `n` are the number of significant digits required for the error bar and the corresponding power of ten of the first significant digit of the error bar, respectively. This core is used in the feedback variables in the feedback trees. The value `ans-errorbar` is then tested with the function NumSigFigs and the test-option [z,z].

We initially worked with a sequence of nodes in the feedback trees in order to query all conditions and to provide individual feedback, which means not simply 'wrong' but differentiated feedback as for example 'too many digits', 'too few digits', 'incorrectly rounded', 'completely wrongly calculated', 'wrong, probably units that do not match the value'.

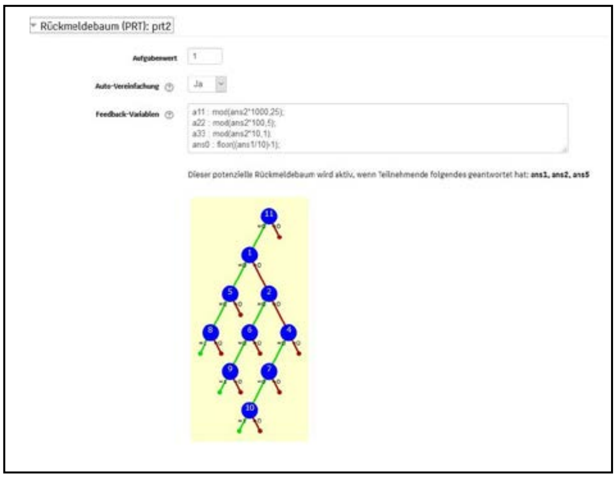

Fig. 2: STACK feedback tree with more ten nodes

We have come to feedback trees with more than ten nodes (as shown in figure Fig. 2) for more than 20 inputs. Each node is followed by the corresponding entry. We noticed that the processing of moodle with STACK became very slow due to the high number of nodes.

Finally, we learned that the query of conditions can be queried in the field 'feedback variables' instead of in nodes, which makes the process much faster. In the field 'feedback variables', if-then-else queries are used to set a *feedback-answer*-parameter and a *response-check*-parameter depending on the student’s answer. The *response-check*-parameter is then queried with a single node and the text of the *feedback-answer*-parameter is given as feedback output.

As an example for the STACK application in our physics lab in Fig. 3 the STACK query is shown for the experiment for determination of gravitational acceleration with the help of a pendulum. In this experiment the students measure the period of oscillation as a function of the pendulum length for 20 different values of the pendulum lengths. As mentioned above, our STACK queries for the lab have to query all measurement data and the entire evaluation of these data in one single task.

On the left side of Fig. 3 a screenshots of the entry form is shown. In the top the experimental data consisting of 20 pairs of measured values for the pendulum length and the time are queried. Below the query of the evaluation data follows as well as of the input of data for the other parts of the experiment together with the corresponding evaluation data.

We have now arrived at the fourth revision cycle for our moodle STACK queries in the physics lab. Every time the result gets a little more sophisticated. This time, consequential errors are caught and commented on. For the verification in some feedback trees, intermediate results entered by the students are used for the calculations. This avoids that a mistake just at the beginning leads to the fact that all subsequent entries are rated as wrong. So far, it was not always clear whether the query was based on the start data or was calculated from intermediate results. If everything is correct, that's not a problem. But experience shows that it is confusing when an interim result is wrong, and later entries are rated as correct. Even parts of the teaching staff, which are not involved in the STACK programming process, are confused. Therefore, both options will be queried in the future (based on the start data or based on intermediate results, respectively) and corresponding differentiated feedback will be given.

4 Summary and Outlook

We have programmed STACK queries for all of the frequently used experiments in our physics student lab. The rules for determining the required number of significant digits for error bars have been implemented. We had and still have a lot of programming effort. When querying so many different options, it is difficult to test everything. Therefore, our students have to serve as testers.

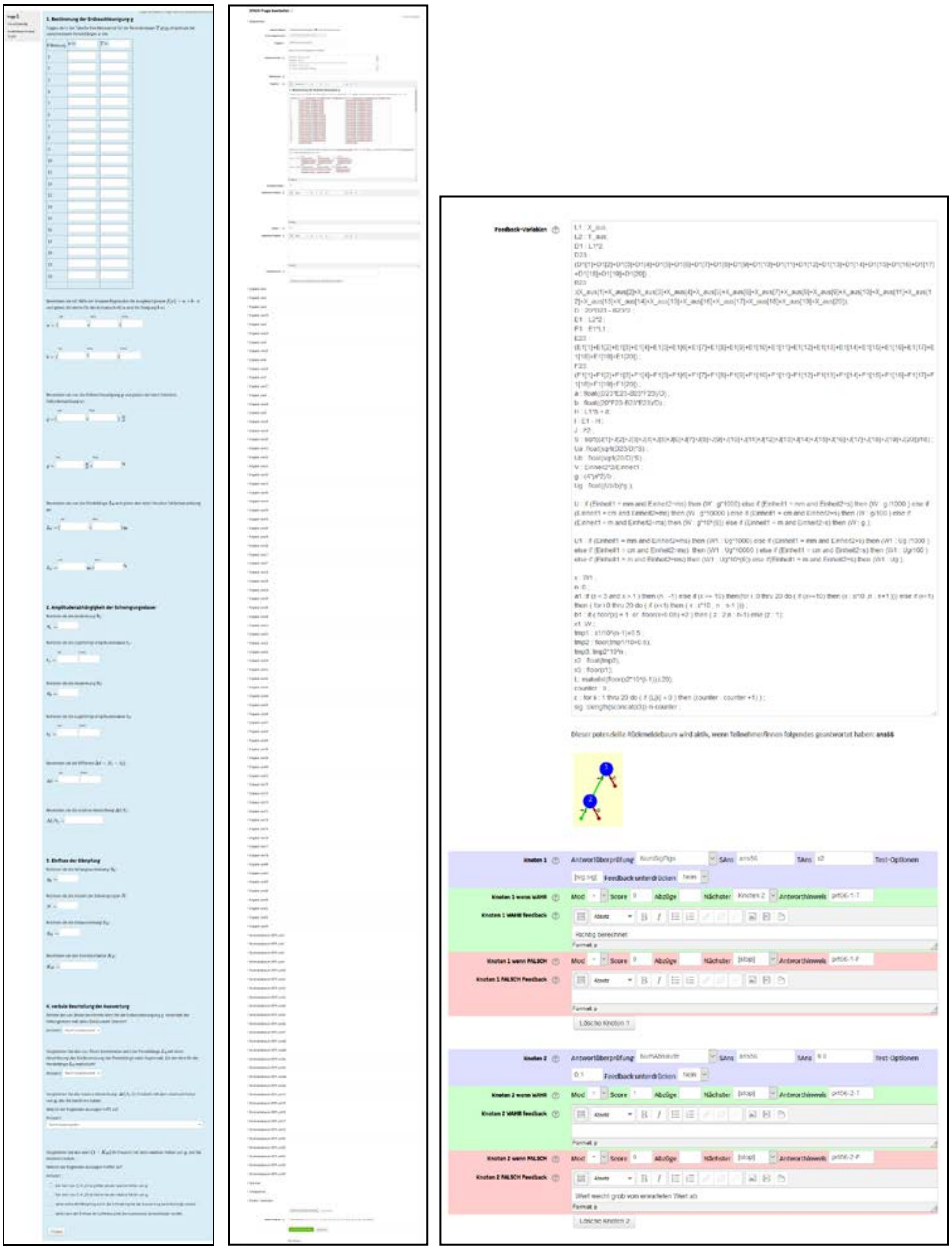

Fig. 3: Example of a lab STACK query: experiment for determination of gravitational acceleration left: screenshots of the entry form, centre: screenshots of the STACK question editor, right: Zoom into the STACK question editor with open feedback tree

Although not everything always runs smoothly, the STACK queries have been used in the physics lab successfully for several semesters.

The review of the student’s results, which has been very time-consuming before, is now automated. Students receive timely feedback, which helps them to process the tasks autonomously.

For the next steps there are still things that can be optimized:

- So far, we have queried the units with self-programmed queries and did not use the ready-to-use STACK unit tool. We will test the STACK unit tool and see how it works for us.
- Up to now, 'Check' is allowed for the students as often as they want before the deadline for submission of results. In some cases, students are simply testing all values until the correct feedback is received. This is not what we want. In the next iteration, we will implement deduction of points when clicking the 'check' button.

Acknowledgement: Great thank to the STACK developers for creating the prerequisite to automate the review of the results of laboratory experiments!

Bibliography

[Kr19] Kröger, S. et.al.: Skript zur Fehlerrechnung [online] https://moodle.htw-berlin.de/pluginfile.php/478691/mod_resource/content/6/V_Fehlerrechnung_0519.pdf [01.06.2020]

[Github] https://github.com/maths/moodle-qtype_stack/blob/master/doc/en/Authoring/Answer_tests_numerical.md [01.06.2020]