Complex Assessments in Mathematics and Computer Science with STACK and VPL

T. Kutzner*[a], C. Steinert[b], M. Weigert[c], O. Wälder[b], K. Robel[c], A. Yan[c]
[abc]Brandenburg University of Technology Cottbus – Senftenberg, Germany
[a] Multimedia Center
[b] Institute for Mathematics
[c] Institute for Medical Technology
* Tobias.Kutzner@b-tu.de; phone: 03573 / 85 672; www.b-tu.de

ABSTRACT

Programming is not complicated and yet many students do not pass the first hurdles. Software is logical, follows clear rules and mathematical laws. Any healthy person with basic literacy, numeracy and writing skills can learn it. Writing a book, painting pictures or other creative challenges are much harder. And yet many fail at comparable competences. Teaching programming skills by listening and reading is not enough. Students need the process of trial and error in order to acquire the skills needed and should be able to learn by them. However, students should be able to work independently on a computer. The problem that the beginners are overwhelmed and the advanced are bored. This is the case especially in computer science. Despite the use of a comfortable Learning Management System (LMS), many tasks have been submitted to paper and PDF at our university so far. Since the available question-types could not cover all possible complex tasks, it was difficult to inspire the lecturers for a complete digitization of their tasks. To be able to automatically evaluate even tasks in mathematics, computer science and related scientific disciplines two modules have been successfully evaluated at our university. This paper presents the use of these Moodle modules STACK and VPL in selected courses in mathematics and computer science and how to randomize tasks. In these modules complex tasks are prepared in such a way that they are still solvable for the students and automatically evaluated by the system, thus could considerably facilitate the work of the lecturers and academic staff. On concrete examples possible solutions are shown.

Keywords: LMS assessment, automatic formative assessment, education, mathematics, computer science

1. INTRODUCTION

In recent years, different digital learning methods in the field of higher education have become established. This includes at least providing the teaching materials in script form as a PDF or the presentation of the lectures in PowerPoint format in an online course. Other formats such as educational videos and electronic tests are experiencing ever higher degrees of penetration at universities [1]. In addition to the established formats, new learning materials are constantly being developed (for example, virtual or augmented realities), or existing ones will be further developed and improved. These include the Moodle modules STACK and VPL. These tasks have expanded the existing online math and computer science tests with new options. After a brief presentation of the potential use of the test and the grades of randomization, which are also relevant for these types of tests, the two modules in this essay are described in their use at our university

2. TYPES OF ONLINE TESTS

The term "digital exams (or test or assessment)" is broad. It can refer to electronic exams (e-exams) and online exams as well as electronic assessments (e-assessments) in general. In principle, the form of input of the results of the test object is the same for all digital tests. This is not, as usual in paper form, but with a digital device it is needed in order for the tests formats to no differ on varying application scenarios. Otherwise, the digital test formats differ depending on the application scenario. For this reason, the terminology is to be delimited below [2]. A fundamental differentiation possibility with (electronic) forms of examination lies in the respective time of the use of the formats in the learning process. Here a distinction is made into formative, summative and diagnostic assessments. This work focuses on these three forms in the context of digital input.

During the diagnostic tests, an actual state of knowledge of the candidates at universities is often recorded prior to their studies or at the beginning of a study module. These should show existing competences and deficits to both students and teachers. For example, teachers can explain faster or skip known course contents faster.
Some diagnostic assessments can also predict the expected learning outcome. Possible scenarios are [3]:

- Aptitude Test
- Entrance Test

Formative assessments are study-accompanying tests. These should give the students and lecturers feedback on the current learning success within a series of courses (several content-mediating attendance phases with home learning phases). This form of examination reflects to what extent aspects have already been understood and where uncertainties still exist. Teachers can thus address the deficits in the presence phase or adapt their teaching tempo. Possible scenarios are [3]:

- Lecture Preparation
- Follow-up and e-exercises
- Intermediate tests during self-learning
- Audience response in the lecture hall (also possible in only one lesson)

A summative exam measures achievements or skills following the learning process and thus the final learning success at the end of the semester. This indicates the degree of consistency of learning objectives with the achieved learning outcomes. Examples of this type of test are the following [3]:

- Electronic exams
- Scan-examinations (written exam on a standardized questionnaire, which is scanned after submission and compares with a stored solution schema and assigns appropriate evaluations)
- Video distance tests

Probably the most important legally binding principle in summative examinations is, in addition to the inclusion in the examination regulations in the higher education sector, that of equal opportunity [4].In principle, it can be deduced from the Equal Treatment Act that all examinees should be given the same chances of success. Test specimens should therefore be granted no advantages or disadvantages. Changed test conditions due to disadvantage compensation only compensate for disadvantages in this case and cannot be regarded as an advantage.
From this it can be deduced that comparable examination modalities are to be applied for comparable candidates. E-exams are to be completed on digital input devices. These represent an aid. In terms of equality of treatment, all equipment under test must have the same characteristics available. The use of own devices (e.g. laptops) is therefore not readily possible.

3. RANDOMIZE TASKS

The random compilation of digital test questions makes it possible to make tests more sustainable and more secure against simple copying tasks [1] [5]. However, the way of randomizing tasks can now be designed differently. This section briefly presents common methods of randomization and their grade for digital tests that have been identified by the author.
Grade 1 and therefore the easiest to be implemented is when the order of questions within a test is put together randomly. For this only several tasks have to be present in a test. However, it should be noted here that tasks that build up on each other are also consistent (coherent), since otherwise some tasks cannot be solved, or only indirectly. This is the case for questions that require a certain sequence and make use of calculations/values obtained in previous questions.

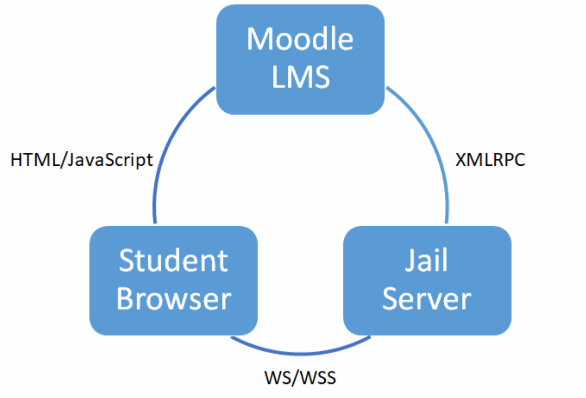
Randomization of grade 2 occurs when at least one task of a test draws on a pool of different questions. Basically, this is a procedure similar to the randomization of grade 1 (tasks must be kept consistent here, too), but not all questions are presented to the test participant here. The participants can process so different tests. Insofar as a test with such tasks is to be used as a final test, it should be noted that the task difficulties must be similar to ensure equal opportunities for each student. An example task that relies on such a pool may be the calculation of extremities of functions. One task of the pool can have a maxima and another, a minimum as a result, the third alternative would then be a task with a function that has neither maximum nor minimum (deposited responses and feedback are then designed depending on the case). As an example, here's the math app developed at the BTU whose tasks are divided into tests and pools [6]. Depending on how well the tasks were solved, the student gets tasks from the pool or gets to the next test.
Randomization grade 3 can now only occur for tasks in which numerical values or variables (for example, tasks from mathematics or programming tasks from computer science) are present. Here, one or more values of the task can come from a previously defined number or letter range and then randomly assigned to the task. The randomization here can be freely chosen, which opens up a variety of task opportunities. Here, too, the principle of equal opportunities in final examinations must be given. When creating the task, care should also be taken to

 ISBN 978-84-09-02374-5

ensure that the randomness of the variables excludes impossible or incorrect solutions. (e.g., disregarded division by 0)
Should individual tasks lead to incorrect tasks in summative tests, they should not be taken into account in the evaluation. The rating scale should therefore be adjusted. The same is true for ambiguous answer-choice tasks.
A test can contain tasks with all three degrees of randomization just mentioned.

4. VPL

VPL stands for Virtual Programming Lab and is a development of the University of Las Palmas de Gran Canaria (ULPGC). VPL is composed of three elements: a Moodle's module, a browser-based code editor and a jail component [7]. A basic description of VPL's operations is as follows; when each student submits her/his program, the Moodle server packs the submitted program in XMLRPC message, ships it to the jail server where it is executed, and the captured output is sent back to the Moodle (see Figure 1). With the help of VPL, programming tasks can be carried out directly in the learning platform. The upload of programming tasks, the compilation with subsequent error output, feedback on the tasks and evaluation of the tasks is possible.



WS (WebSocet)
WSS (WebSocet Secure)
XMLRPC (Extensible Markup Language Remote Procedure Call)

Figure 1. VPL configuration relations

Following the existing agreement for scientific exchange between BTU and ULPGC since 2014, the developer introduced us to the Moodle module VPL during one of the meetings. The research (see Table 1) also resulted in a predominantly positive evaluation of the module.

Table 1 Strengths and weaknesses present on automatic evaluation systems such as iVProg[1] and VPL [8] [9] [10]

Positives	Negatives
Presentation of results through the test cases lets you know if your code is working correctly [8]	Dependence on compilation and of the results of the test cases making the student does not look for errors alone [8]
Easy to view the syntax errors [8]	Missed print messages /instructions to users [8]
Immediate feedback to the student [8]	Show that the automatic evaluation has flaws [8]
Saving time for solving exercises [8]	Inhibits the practice of how to test the algorithm [8]

It gives an idea of your note through the percentage of correct answers of the exercises [8]	Students perceived the built-in editor of VPL is not preferable [9]
Easy to correct [8]	
Three students from 16 declared "I do not see negatives aspects" [8]	
VPL can be access free [9]	
VPL can compile and run many programming languages [9].	
82.5% from 63 students asked in [10] “I wish I had been able to use the VPL to solve more exercises”.	

In summary, it can be stated that with VPL it is possible for students to learn and work with the same development environment. Local programming environments are not necessary, which is especially beneficial for less program-intensive degree programs such as electrical engineering and mechanical engineering. Programming tasks can be solved immediately in the learning platform and so every student has the same conditions. As the research has shown, the positive aspects of using automatic evaluation systems such as VPL in teaching are overwhelming. Therefore, we have decided to introduce the Moodle VPL module into the teaching at BTU and to test it in a first step (pilot phase) in a selected course of electrical engineering and computer science.

A first pilot phase with students testing courses with tasks for electrical engineering and procedural programming will start at winter semester 2018/2019 at BTU. Previously, the tasks were solved in the computer lab or on their own development environment at home, and students submitted the results as a PDF, source code and sometimes even on paper. The module VPL now enables the complete solution of the tasks in the learning platform and the evaluation by the lecturer as a formative assessment. Uploading PDFs, source code files and submitting paper tasks is no longer necessary. Previously, lecturers in Moodle often had problems editing the PDF files and provide the files with comments, and the submitted source code files had to be carefully reviewed and in case of doubt run again on the lecturer’s computer to find, and correct errors. The evaluation was done by hand. The whole procedure entails a high expenditure of time on many given tasks. Now the effort is shifted once in the creation of the tasks, if the task is completed with appropriate description to independently solve the tasks and the test cases are reduced by the complete automation of the tasks of the time spent on the lecturer considerably, what our lecturers are convinced that the module could lead to a considerable time saving in the correction of the tasks.
The tasks have been adapted so that they can be solved completely without the help of the lecturer in the learning platform, as little as possible source code is given, the students could compile and evaluate the tasks to be pointed out at the time of creating the task errors and be able to correct them. We hope this will lead to a greater learning effect, since everyone strives to solve the tasks as correct as possible. The possibility of errors due to typos, uploading the file, is also reduced even restricted to zero, since the by the student edited source code is submitted directly to the corresponding task in the online course.

Below is an example task in Moodle solved with the VPL module. We can see the built-in editor, the source code with syntax highlighting and on the right side the past tests. So the student has the opportunity to see if his program has been solved correctly or still has errors. Furthermore, there is always the possibility to execute the program at any time and thus to recognize and to improve errors during compilation. (see Figure 2)

[1] *Interactive Visual Programming in the Internet* is a system targeting the teaching-learning of algorithms through the World Wide Web.

 ISBN 978-84-09-02374-5

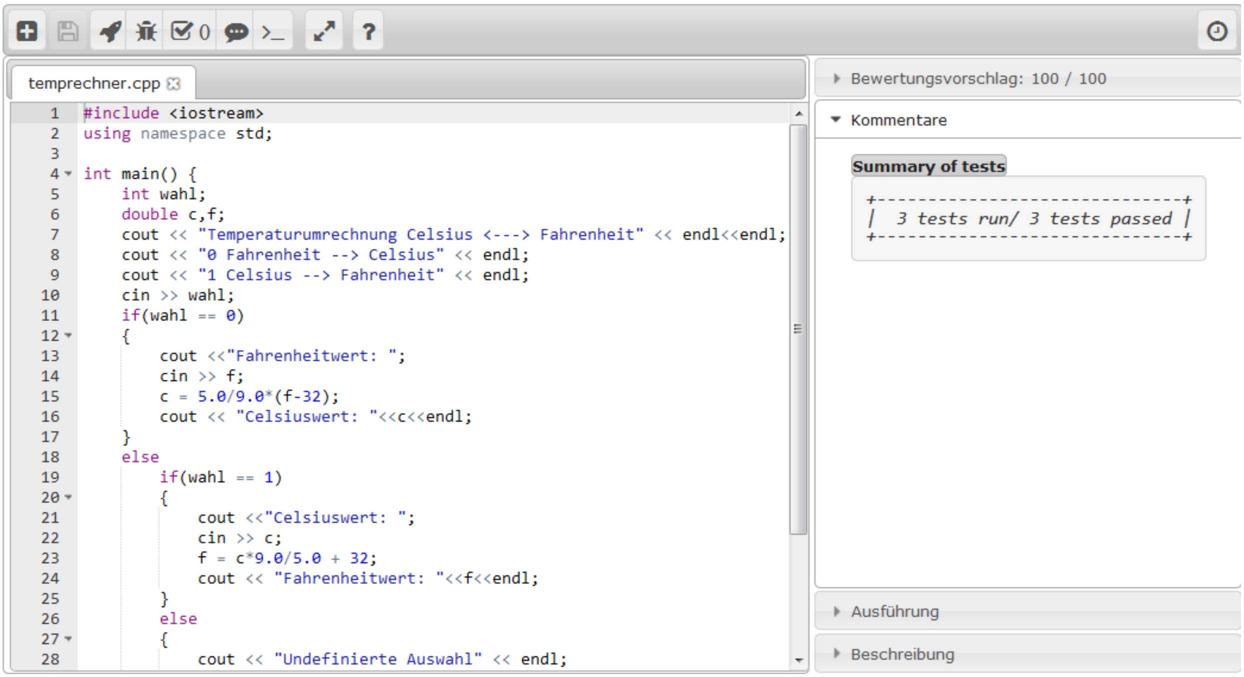


Figure 2. Source code files (Solution of the lecturer)

To have the opportunity to define specific tests to give the student the possibility to test his program on the one hand and the test cases can then also be used to automatically evaluate the task and give a mark to the student on the other hand. (see Figure 3)

Figure 3. Test cases

5. STACK

STACK stands for "System for Teaching and Assessment using Computer algebra Kernel" and is a CAS system in the field of mathematics. Different mathematical expressions such as variables, functions, equations, sets, lists and matrices can be used and entered. Through various answer verification these answers can be checked. A big difference to other question types is that you can not only check whether the input matches the model solution or not. It can also be checked, for example, whether an entered function is specified in a factorized form or that the solution matches the naming of the variables or constants. Through these possibilities, the teacher has scope in the evaluation and can therefore also derive the inputs from the learners. This is made possible by a computer algebra system (CAS) maxima whereupon STACK accesses back (see Figure 4) [11].

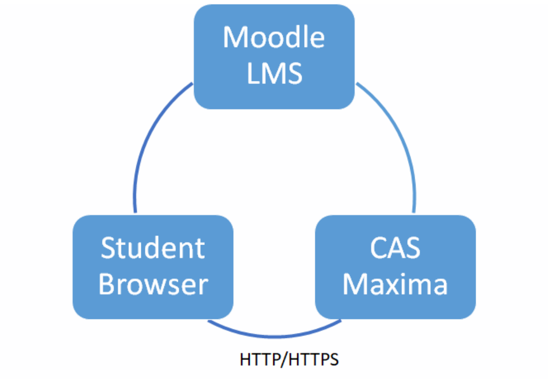


Figure 4. STACK configuration relations

Maxima allows you to use up to 900 math commands in LMS Moodle for task creation. Unfortunately, not all STACK commands work in every Moodle or STACK version. For this reason, a test scenario for the 900 commands was implemented at the BTU. At the same time, the commands have been documented. The documentation includes calling the command in Moodle and the expected impact. The documentation was created as part of various student projects at the BTU. Due to the complexity of STACK also a community founded from teachers at the BTU, which is regularly exchanging to the module. The meetings take place at least once per semester. On these will be exchanged for the innovations in STACK and individual developments. In case of problems, the teachers will also help each other. For example, STACK training courses are also offered at the BTU.

So far only a few STACK questions have been implemented at the BTU. This is because now even more questions were created with a similar Moodle module "Formulas" (over 400 questions) (Steinert, Wälder, Bothe, 2017). These questions have already been tested with students and are used as formative assessments in the math lectures at the BTU (in Senftenberg). STACK offers some advantages compared to Formulas due to the CAS system. This allows students to see their input mathematical compiled appropriately (see Figure 5).

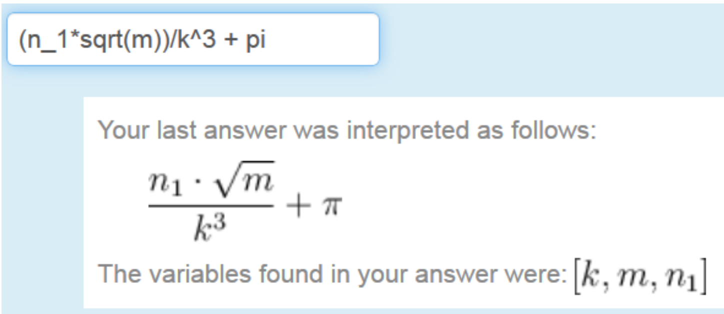


Figure 5. Input options in STACK and automatic interpretation with visualization for students

Furthermore, the STACK commands can be used to establish some new types of tasks, such as graphical plots (Figure 6). Therefore, some selected existing tasks will be adapted and new ones developed.

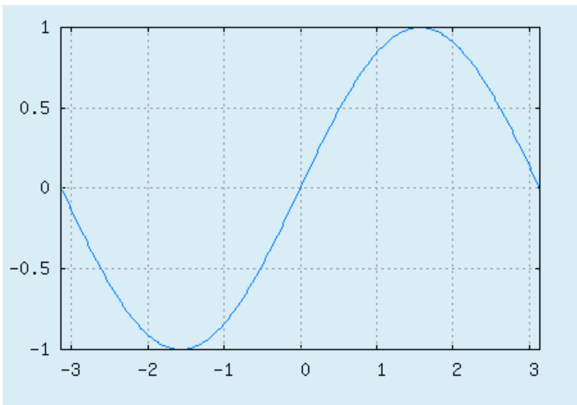


Figure 6. Plot presentations in STACK for student feedback or assignments

 ISBN 978-84-09-02374-5

6. CONCLUSION

Creating good and sustainable digital testing tasks is complex. The two described software solutions STACK and VPL are suitable for formative testing. With this work we intend to study the impacts of VPL on the process of teach- and learn programming. We have the expectation that the process will lead us to an improvement in programming learning methodologies. For diagnostic and summative testing, special cases have to be developed and tested. In any case, the creation of such tasks is associated with a considerable additional effort for the preparation of the materials by the teacher. So far in the STACK-tasks there is no real compensation in the work with the students. The students have begun to formulate an increasing number of messages with queries or notes on the tasks. However, dynamic tasks make for the students optional formats like tutorials and office hours more likely to be visited, resulting in increased dialogue with learners. An uncommented learning in the field of mathematics, which focuses only on querying the last answer (querying partial solutions here would be too strong a solution) also does not make sense, if the exam is a written exam, in which participants must comply with formalities [12]. Here, however, the increased interest in the optional formats can be used to talk about the solutions.

7. OUTLOOK

In the tests with STACK and VPL selected tasks will be implemented in the future within the STACK and VPL community at the BTU. In the future, it will also be discussed how far it is possible to participate in cross-university questionnaires [13]. For VPL we examine the possibility of testing by debugging tasks and integrating UNIT tests.

8. ACKNOWLEDGEMENTS

The authors would like to thank the Federal Ministry of Education and Research (BMBF) for their support of the project (01PL17029) under the direction of Prof. Dr.-Ing. M. Koziol. In addition, the authors want to thank BTU's data center (URZ) for the provision of CAS Maxima and VPL server as well as the operation of the LMS Moodle.

REFERENCES

[1] Steinert, C., Kutzner T., and Wälder O., Higher Education in Mathematics with Interactive Media, InnoEducaTIC, 157-163 (2017).
[2] Vogt, M., and Schneider, S., [E-Klausuren an Hochschulen. Koordinationsstelle Multimedia.], Hochschulrechenzentrum. Justus-Liebig-Universität Gießen, (2009).
[3] Schmees, M., and Horn, J.; [E-Assessments an Hochschulen: Ein Überblick: Szenarien. Praxis-Klausur-Recht], Wachsmann Verlag, (2014).
[4] Niehues, N., Fischer, E., and Jeremias, C., [Prüfungsrecht 6., neubearbeitete Auflage], C.H.Beck, (2014).
[5] Gottwald, M., Steinert, C., and Brandt, A., "Studienvorbereitung mit randomisierten elektronischen Testaufgaben," U. Kortenkamp, and A. Kuzle, Beiträge zum Mathematikunterricht 2017, Münster: WTM-Verlag, 1193-1196 (2017).
[6] Kutzner, T., Steinert, C., and Wälder, O., " Bilinguale mobile Anwendung (App) zur Auffrischung der Mathematikkenntnisse in der Studieneingangsphase," GML², 303-305 (2016).
[7] Rodríguez-del-Pino, J. C., Rubio-Royo, E. and Hernández-Figueroa Z., "A Virtual Programming Lab for Moodle with automatic assessment and anti-plagiarism features" International Conference on e-Learning, e-Business, Enterprise Information Systems, & e-Government, (2012).
[8] Brandão, L., Bosse, Y. and Gerosa, M., "Visual programming and automatic evaluation of exercises: an experience with a STEM course", IEEE, 978-1-5090-1790-4, (2016).
[9] Kaunang, S., Paturusi, S., Usagawa, T., Mangindaan, G., Sambul, A. and Sugiarso, B., "Student perceptions of virtual programming lab on e-learning class at University of Sam Ratulangi", International Conference on Information, Communication Technology and System (ICTS), (2016).
[10] Cardoso, M., Castro A. and Rocha Á, "Integration of Virtual Programming Lab in a process of teaching programming EduScrum based", 13th Iberian Conference on Information Systems and Technologies (CISTI). IEEE, (2018).
[11] Vasko, M., "Ein Online-System für Hausaufgaben zur Ingenieurmathematik-Chancen und Herausforderungen," 14. Workshop Mathematik, 34-41 (2017).
[12] Biggs, J., "Enhancing teaching through constructive alignment "Higher Education 32, 347-364 (1996).
[13] AK Mathe Digital, "Datenbank für Digitale Mathematikaufgaben", Slate 12 Juli 2018,< http://www.ruhr-uni-bochum.de/ak-mathe-digital/domain.html> (19 September 2018).