/

Approach to Development of Assignments with Graphical Solutions for MOOC Course

Denis A. Korolev, Anastasiya V. Gorokhova-Alekseeva, Anna A. Vitshas, Aleksandra O. Pleshkova
Department of Computer Engineering
National Research University Higher School of Economics
Moscow, Russia
dkorolev@hse.ru

***Abstract* -- Assessment tools in MOOCs vary from simple tests to customizable scripts with adaptive feedback, but commonly they check student's submission for correct answer. In real life we often meet cases where many correct answers are possible and the best solution requires good feeling of the context of implementation. This paper describes an approach experimentally implemented for online course of video technologies where engineering tasks are often based on real-life experience, not just specifications and rules. To transfer this experience a multi-layer model of assessment was proposed and partly implemented in experimental tasks. This example shows how assessment tool turns out a training platform where technically correct solution is just a start of work and variable feedback leads students to understanding of engineering approach, given by the author of the tasks.**

Keywords – MOOC; assessment; training; variable feedback; video technologies; automated submission processing

I. Introduction

MOOCs are widely used both by higher education institutions and by individuals willing to get new knowledge in selected areas. Course development requires to create a set of tasks to make the learning interactive and deep, but often platforms and courses offer just tests of fill-in questions and, sometimes, peer review assignments. There are dedicated platforms such as Codeacademy, providing interactive tools to teach specific types of courses such as programming.

There are platforms such as Stepik.org, that offer a rich set of instruments for course creation, and are open for custom processing of students' responses. Custom response checking code allows to analyze student's documents or text responses and this feature allows to create new types of assignments that are not provided by the platform. In this paper, we will discuss approaches to the creation of a natively unsupported type of assignment for training rather than evaluating knowledge, with consultative responses to students. Despite the courses on dedicated platforms such as programming, suggested approach is applicable for different topics that require graphical solutions (diagrams, schemas) instead of text answers.

II. Approach

Here we will take example assignments for Video technologies course where students have a library of devices (Figure 1) and a variety of types of cables to construct setups for given purposes with given priority criteria such as price, size, usability, etc. -- the same task will be solved different way when having different criteria in mind.

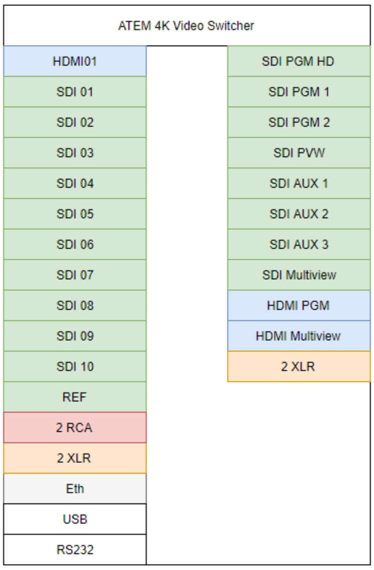

Fig. 1. Representation of a video switcher in library of elements

Automatic assessment assumes an assignment has a correct answer and student's response is checked against this predefined answer. But there are tasks with no exact answer and subject to check is complex. We will determine six levels of abstraction where respectively the following tests are applied:

1. Drawing format. This is a basic test on format compliance. We suggest students use Draw.io online editor and provide element libraries for this editor.

2. Shapes and connections on the drawing: shapes should be taken from the given library, connectors should lead from outputs to inputs of the shapes.

3. Physical interface check. Connection lines represent cables, inputs and outputs represent connectors such as HDMI, USB, Ethernet, etc. Types of cables (coaxial, UTP, etc.) can be color-coded and the connectors defined with a text caption on the connection line, e.g.

978-1-5090-3680-6/16/$31.00 ©2016 IEEE

"BNC-RCA" or "XLRm-XLRf" when the male-female type of connector need to be defined.

4. Logical interface check. If a BNC connector is plugged into a BNC socket, it does not guarantee that a composite output was not connected to an SDI input or any other type of input with BNC socket. This level is responsible for signal compatibility validation.

5. The logical consistency of the diagram. Devices in the library have inputs and outputs of the same physical and logical interface yet having different purposes. E.g. video switcher can have a program, auxiliary and multi-view outputs of the same video format, but normally multiview output is not sent to video recorder and at least program output should be connected to a consumer -- recorder, capturing or streaming device.

6. Priority criteria compliance. Is the given solution (possibly, technically correct) optimal for the given purpose (the most portable, the cheapest, etc.).

Besides the tests for correctness, there can be formed a list of devices and cables used.

The most effective way of using this analysis of users' responses is to give tasks for training rather than validating knowledge. This is why the feedback is very important and multi-layer check releases teacher's' attention from low-level errors such as incorrect drawing or improper cable or connectors usage to forming the idea of the task visualized in diagrams.

On the other side, users' formal low-level errors are described once and then inherited to all tasks that use the same library or, partly, the same approach: another library of elements will probably have similar drawing rules, though connectors may have different meaning (e.g. not SDI or Ethernet cable, but RTMP or RTSP stream when passing from explanations of signal switching to stream processing).

III. Technology

Stepik provides two ways to customize response processing:

1. Analysis of the response with Python code on the platform for each assignment. Adaptive feedback depending on user's response is possible.
2. "Teacher review" where the "teacher" is not really a human, but a program accessing students' responses and returning results back via its REST API [1] using JSON. In this case, there are no specific requirements to processing software or languages used.

The first way is natively supported and has no integrity risks, and lightweight scripts that do not require much of processing job will be smoothly executed on the Platform immediately after student's submission. Another drawback of running code inline is the restricted access to external communication such as linking common data for all tasks in the course, but this can be solved in cooperation with platform developers team.

The second way fits best for 'heavy' processing or responses with large uploads such as video clips or high-resolution photos. From previous experience on Stepik, we know that simple conversion of 8 megapixels RAW to JPEG may tape up to 10 minutes because the servers are shared between many courses and users on them. This delay gives bad user experience and such (and more consumptive) tasks require external processing power and, sometimes, much disk space to operate video and graphical data. For MOOCs redirection to a single dedicated server may become a narrow place, but only if student expects a real-time response to their submission. Once submitted to 'teacher's review', same as peer review, the work is not expected to be reviewed immediately. On the server's side, all submissions are queued and feedback is sent after processing. This approach allows implementing processor-consumptive algorithms, that could never give response in real-time [2]. Also, this allows to run tasks, that themselves require noticeable time to complete and cannot be hurried up.

IV. Conclusions

Normally MOOC and LMS systems provide more or less comprehensive tools for assessment learners' knowledge and these tools have one correct answer. In real life, there can be a variety of different correct solutions and the assessment program should give a review of the proposed solution rather than just give a conclusion -- correct or not.

Splitting the assessment to layers allows to keep common low-level tests apart from unique high-level tests and pay more attention to programming the idea of task.

V. Future work

Currently, few sample tasks were programmed [3], but the logic of the task and the feedback are placed in the code. The aim of the further work is to create a simple formal language of description for the tasks (upper levels description) and for the library elements (for lower levels). There should not be required programmer's skills to create new task if using the same library.

Acknowledgements

This experimental work is being done in contact with Stepik.org team and inspired by their approach to online teaching.

References

[1] Stepik REST API https://d.pr/11YiXC (Google document).

[2] Mi, F. (2015). Machine learning models for some learning analytics issues in massive open online courses. // Machine learning models for some learning analytics issues in massive open online courses,2015. https://lbdiscover.ust.hk/bib/991002043169703412

[3] Example of a task with graphical solution: https://stepik.org/lesson/45655.