Assessment System for IT Configuration Tasks

Tamás Kádek[1], Piroska Biró[1, 2]

[1]University of Debrecen, Faculty of Informatics, Kassai út 26, 4028 Debrecen, kadek.tamas@inf.unideb.hu; biro.piroska@inf.unideb.hu

[2]Sapientia Hungarian University of Transylvania, Faculty of Economics, Socio-Human Sciences and Engineering, Miercurea-Ciuc, Romania, biropiroska@uni.sapientia.ro

Abstract: During the COVID-19 pandemic, which also deeply impacted the higher education system, several creative solutions were found to address the challenges faced. Some of these solutions have stood the test of time and are still with us in some form today. At the University of Debrecen, for distance learning of the IT Security subject and especially for the management of exams, we have developed an exam system to automate the assessment of configuration tasks on virtual machines. The basic idea proved to be useful even after the pandemic had passed, and the system, which was quickly put together three years ago, is still in use. Given our longer-term commitment to using the solutions we have put in place, the time has come to rework the software, identify its weaknesses, and fix them in a new version. However, before starting this work, it seems worthwhile to analyse the data series accumulated during three years of use. The purpose of this is twofold: firstly, to carry out the usual pedagogical analyses so that the output can be fed back into the educational system, and secondly, to formulate the new requirements for the new system to examine the pedagogical analyses. In this article, we will describe how the previous system worked, present a brief analysis using the data collected over three years, and gather our requirements for the new system we are planning.

Keywords: assessment system; IT security education; operation system configuration exercises

1 Introduction

The subject of IT Security is the practical raising of questions related to basic IT security [1]. The course consists of two labs per week, where an important objective is to correctly configure computers with security in mind. This includes the management of users, their privileges, login methods, authentication of files, and encryption.

The forced introduction of distance learning in the spring 2020 semester has also substantially changed the design of such labs, which were previously done in a computer lab, with students trying to do it in a home environment.

Since we were also using virtual machines in the labs for configuration tasks, there was the opportunity for students to work with the same virtual machine in their home environment. The advantage of doing the tasks on the student's own machine is that it does not impose any additional burden on the faculty IT infrastructure. At the same time, the organisation of examinations [2], which could previously be carried out in the laboratories under the supervision of a tutor, was a challenge, but this was no longer possible.

The University of Debrecen, Faculty of Informatics has been experimenting for decades with various automatic and remote task evaluation systems, such as ProgCont, which was developed for the automatic evaluation of programming tasks [3], [4]. For the IT Security subject, using the experience gained in this field, we have developed a system for the end of the Spring 2020 semester, during the exams, which can automatically check different configuration tasks of virtual machines.

The rapid implementation of the proprietary system (initially completed in 2 weeks) was made possible by the use of the existing infrastructure (servers) of the ProgCont system [5], even though we know that similar systems already existed [6], [7], [8], learning and implementing them seemed to be a much more difficult task.

We have since put the COVID-19 pandemic behind us, but the system has certainly shown benefits that have justified the use of the Exam system ever since:

- make the evaluation of tasks automatic and objective
- students receive immediate feedback and have the opportunity to make corrections
- they can practice on their own virtual machines at any time

However, the time is right to use the experience of the software, which has been suddenly put together out of necessity during the semester, to rethink how to make the possibility of automated assessment an effective tool for teaching the subject.

2 The Exam System

The first challenge is to test students at a distance. This requires a sufficient number of tasks so that everyone is likely to be given different ones; they cannot work together, and they are informed immediately of the assessment of these tasks so that they can make corrections. We have had experience with automatic assessment and examination systems because of the ProgCont system developed by our faculty. The ProgCont system evaluates programming tasks [9], [10], [11]. To do this, students have to upload the source of the application that implements the

programming task into the system, and in return, they get back an evaluation by examining the running result of the compiled program (e.g. whether it matches the expected output for a given input). Within a given time limit, a student can upload source code for evaluation as many times as he/she wishes.

The software, which was first developed for ACM-style programming competitions, was soon used to help run assessments.

Its advantages:

- the evaluation of solutions is almost instantaneous; students have the possibility to correct them,
- the evaluation is objective: the decision to accept a solution is based solely on the results of the pre-prepared test cases
- online: the same environment that hosts the student during the assessment is also available for home practice
- developed in-house so it is fully adapted to our needs.

While the goal is to evaluate the source code of a program in the case of the ProgCont system, the goal is to check the configuration of a virtual machine in the case of the IT Security subject. In the case of virtual machines (also to save faculty resources), we have tried to keep the virtual machines at the student's premises, running on the students' own machines. As a result, to evaluate the solution of the tasks (which means testing the configuration of the virtual machine), only the client program running on the virtual machine and controlling it needed to be created. This is advantageous for several reasons. On the one hand, students can use virtual machines created on their own system for training purposes; virtual machines are easy to manage, their state can be saved and restored, students can restore a broken configuration with a single click, and virtual machines do not place any burden on the faculty infrastructure. Of course, if someone didn't have the right equipment to run a virtual machine (which was rare in 2020), we could create a limited number of faculty servers.

The procedure for solving the exam questions was as follows:

1. The student logs in to an online interface and requests an exam task. The set of tasks is automatically compiled from a set of specific tasks randomly selected according to the rules of the task book prepared by the instructors to avoid students taking the same test at the same time and receiving the same tasks.
2. Each set of tasks created for the student has its own unique ID, which the student must use to prepare their virtual machine. The preparation consists of running a single command that, based on the exam ID, performs the necessary preparatory steps to solve the tasks, e.g., creating the necessary files to perform the tasks.

3. The student will complete the configuration steps required to solve each task in any order.
4. The student can request a check of the configuration at any time during the exam by running a check command on the virtual machine and entering their own exam ID. The check command checks the configurations required in each task, sends the result to the exam portal, and displays the result in aggregated form in the command output.
5. The student can check which tasks have been successfully solved on the exam portal. They can retake an unlimited number of failed problems during the examination.
6. At the end of the exam, the current score displayed on the examination platform is the result of the exam.

The advantage is that the student can no longer lose points during the exam; if he executes an instruction on the virtual machine that breaks previously accepted configurations, he still keeps the previously awarded points. So if he accidentally breaks the machine during the exam, he only has to reset it to its initial state (even before the exam) and only has to deal with the tasks that have not yet been solved.

Using the experience gained in the development of the ProgCont system and drawing on the foundations of the existing infrastructure, it seemed easier to build a system for this task than to adapt existing systems. The design of our new examination system was very similar due to the nature of the task, and the infrastructure (web and database servers) required to implement it was also very similar. The exploration of similar software (e.g. Cisco Networking Academy) was left for later, but it is the basis for many ideas for further development [12], [13].

3 Methodology

Our system was first used in the examination process, so this is the area where we have the most student data. Later, the question sets for the examinations were also made available during the practice sessions, but of course, the circumstances of the assignment were different, so in this study, we concentrated only on the data collected in the examinations, which involved the evaluation of a total of 1329 tests.

We grouped the tasks into groups, as there were many different tasks with the same configuration problem in the system, in order to make the set of tasks generated for each student as unique as possible. Often, there was only a single file name to distinguish the tasks. However, this could be done to ensure that two different sets of tasks were not interchangeable at the debriefing. On the other hand, the correct configuration done by one student should not give the correct solution to another student's tasks.

In our study, we collected the groups of tasks that were used in more than one examination, so that for the English language course, we identified 11 groups of tasks to be tested.

3.1 Methods of the Selection

In the 3 years since the introduction of the system, we have organised a total of 14 exams in English. In total, 11 of these examinations included several sets of tasks, but not all of them included all of the tasks.

Our aim was to select the groups of tasks in such a way that we could test as many participants as possible in examinations containing all of them.

The examined task groups were the following:

- Task T1: Configure a public key authentication for a given private key.
- Task T2: Create a file with a given size.
- Task T3: Set file permissions using ACL entry.
- Task T4: Sign a given file using a given private key.
- Task T5: Decryption using DES-EDE.
- Task T6: Decryption using AES.
- Task T7: Create a new file system on a new partition.
- Task T8: Mount an encrypted file system.
- Task T9: Verify the authenticity of the signature of a given file.
- Task T10: Create a new user with a locked password.
- Task T11: Find files owned by a given user.

Table 1
The most participants, choosing groups of different numbers of tasks

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	∑	Students
								1			1	226
		1	1								2	204
		1	1			1		1			4	188
1		**1**	**1**			**1**		**1**			**5**	**167**
		1	1			1		1	1	1	6	164
1		1	1			1		1	1	1	7	143
	1	1	1	1		1		1	1	1	8	123
1		1	1		1	1	1	1	1	1	9	114

Our possibilities are shown in Table 1. If we had selected a single task group, T9, which would have been the group with the largest number of students (226), as shown in the last column, we could have examined 9 task groups, but the number of task groups was so small that only 114 students were examined, so the selection of 5 task groups seemed to be the most appropriate mean, which was able to examine the results of 167 students. Based on Table 1, from the above-mentioned groups, we have selected T1, T3, T4, T7, and T9. These 5 tasks were simultaneously involved in 14 exams in 4 different semesters. These and the number of participants are listed in Table 2.

Table 2
Distribution of selected task groups by year

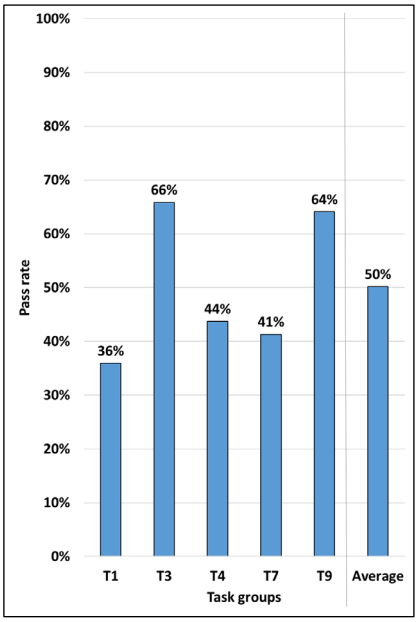
	2020				**2021**					**2022**						**2023**						
	E1	**E2**	**E3**	**E4**	**E1**	**E2**	**E3**	**E4**	**E5**	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**
	16	16	9	5	1	13	13	10	10	19	20	11	4	9	6	12	12	15	16	9	10	6
T1	0	1	1		1	1				1	1			1		1	1	1	1	1	1	1
T2	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0					
T3	0	1	1	0	1	1			0	1	1			1	0	1	1	1	1	1	1	1
T4	0	1	1	0	1	1			0	1	1			1	0	1	1	1	1	1	1	1
T5	0	0	0	0	0	0			0	0	0			0	0	0	0					
T6	0	0	0	0		0			0		0				0	0	0	0	0	0	0	0
T7		1	1	0	1	1			0	1	1			1	0	1	1	1	1	1	1	1
T8		0	0			0	0	0			0	0	0			0	0	0	0	0	0	0
T9		1	1	0	1	1	0	0	0	1	1	0	0	1	0	1	1	1	1	1	1	1
T10		0		0	0	0			0	0	0			0	0	0	0		0	0	0	0
T11		0		0	0	0			0	0	0			0	0	0	0		0	0	0	0

Each column of the table represented a single examination. The first row shows the number of students who took the exam. The exams are typically completed by students during their laboratory time, and the capacity of the room (maximum 24 students) therefore limits the number of students who can take part in a single exam. Normally, students work in a controlled environment in the computer lab, but the lab schedule has been maintained even during the pandemic-affected semester. The number of participants in the exams related to the same paper in 2020 is not out of line. In the table, we have marked with 0 or 1 the tasks that were part of the accountability, with 1 highlighting the selected tasks.

4 Results

Figure 1 shows the results for each group of tasks, with T1 being the most difficult and T3 the easiest. It is not surprising that the students were more interested in the T1, T4 and T7 tasks, as these are the ones selected for which the solution of a single

command is not sufficient. In contrast, T3 requires only the setfacl command to be properly parameterised, and the name of the required command is immediately apparent from the task description, while T9 requires the execution of several commands, but they differ only in the name of the public key file used. The T1 task is really difficult because the required input data (the public key) is not available in the correct format, but it has to be converted using separate instructions. It often happens that the student forgets the need for the conversion, but pasting the original file into the correct configuration file gives incorrect results. In the case of T4, the conversion of the key can also be a challenge, although in this case, it is easier to detect the error caused by the omitted conversion (the error message of the programs to be used is more informative).



Students	T1	T3	T4	T7	T9
1	0	0	1	0	0
1	0	0	0	1	0
1	1	1	1	0	0
1	0	0	0	1	1
1	1	0	1	1	1
1	1	1	0	0	0
1	0	1	1	1	0
1	1	0	1	0	0
1	0	0	1	1	0
2	1	1	0	0	1
2	0	0	1	0	1
4	0	1	1	0	0
4	0	1	1	1	1
4	0	1	0	1	0
5	1	1	1	0	1
6	1	1	1	1	0
6	1	1	0	1	1
7	0	1	0	1	1
9	0	1	1	0	1
10	0	1	0	0	0
13	0	1	0	0	1
20	0	0	0	0	1
29	0	0	0	0	0
37	1	1	1	1	1
167	**60**	**110**	**73**	**69**	**107**
	36%	66%	44%	41%	64%

Figure 1
Results for task groups

Table 3 also shows similar results by year, with no significant difference between grades, and T3 remains the easiest. For the T9 task, we see some improvement, highlighted in Figure 2, in results year on year.

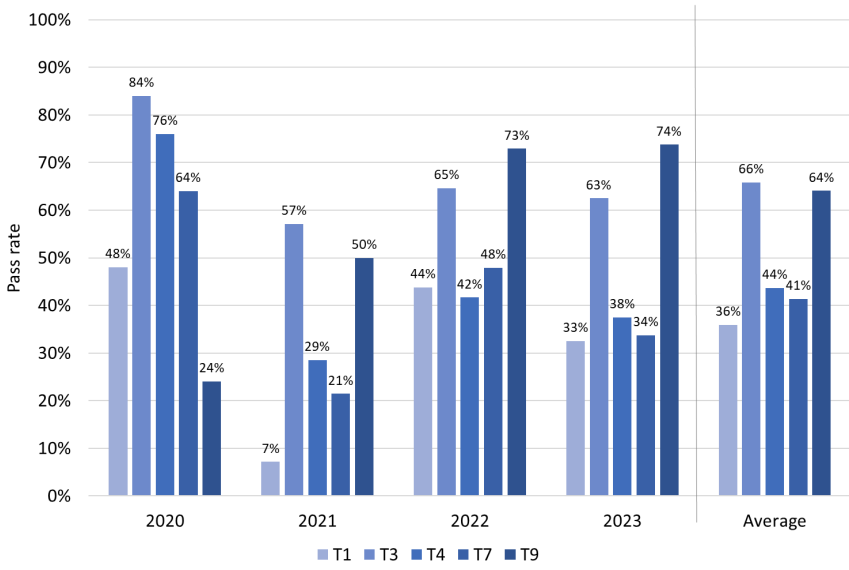


Figure 2
Comparison of success rates

Table 3
Success rate by year for each group of tasks

year	2020	2021	2022	2023	Average
T1	48%	7%	44%	33%	**36%**
T3	84%	57%	65%	63%	**66%**
T4	76%	29%	42%	38%	**44%**
T7	64%	21%	48%	34%	**41%**
T9	24%	50%	73%	74%	**64%**
Average	**59%**	**33%**	**54%**	**48%**	

The correlations between the task groups are shown in Table 4. The first part shows (Success) the percentage of successful completers of the task groups in the columns who also successfully completed the task group in the row. Considering that T3 is the easiest and T1 the hardest, it is not surprising that 97% of those who completed T1 also solved T3 well.

The second part shows the same for the failed solutions (Failure). It can be read that 96% of the students who did not even pass the easiest T3 did not even pass the hardest T1.

Table 4
Correlations between the completion of task groups

Success (%)						Failure (%)						Similarity (%)				
T1	**T3**	**T4**	**T7**	**T9**		**T1**	**T3**	**T4**	**T7**	**T9**		**T1**	**T3**	**T4**	**T7**	**T9**
	97	85	83	85	**T1**		51	79	82	48	**T1**		68	81	83	61
53		61	59	75	**T3**	96		89	93	58	**T3**	68		71	71	69
70	92		68	79	**T4**	90	54		80	48	**T4**	81	71		75	62
72	94	72		81	**T7**	90	54	77		48	**T7**	83	71	75		62
48	78	54	52		**T9**	85	55	75	78		**T9**	61	69	62	62	

The strongest correlation can be seen in the third part (Similarity), where 83% of students performed the same on T1 and T7. The biggest difference is between the T1 and T9 tasks, with only 61% of students scoring the same on these.

5 System Requirements

Over the past few semesters, we have primarily used three different software products to automate programming assignments. In addition to the version of our system currently under review, we have used the locally developed ProgCont system and the well-deservedly popular Moodle for programming assignments. Based on our experience with these systems, reflecting on their strengths and weaknesses, we consider the following requirements to be the most important.

Task-related features [4]:

1. Reusable tasks
2. Multilanguage support
3. Multiple OS support
4. Task verification
5. Solution evaluation
6. Immediate feedback

Management features [4]:

1. Different instructors roles
2. Student group management
3. Exam and practice sessions, lab assessments
4. Advanced statistics

5.1 Task-related Features

- **Reusable Tasks:** The Moodle system offers a great opportunity to edit and collect reusable tasks in the form of a question bank. The questions compiled in this way can appear in multiple tests, just as our original system was designed to reuse task groups.
- **Multilingual support:** All the subjects in which we have implemented our system are part of both our Hungarian and English language courses. Due to the nature of the tasks, the solution procedure is the same regardless of the language of the training, i.e., the same commands are used to perform the necessary configuration steps; only the task description varies. This gives the next level of reuse of the tasks in different language sets. The introduction of multilingual task descriptions thus avoids the duplication of scripts for automated task evaluation. This contributes to both easier maintenance of the question bank and easier production of item-based statistics. This also means greater usability, whereas in the Moodle system, each question belongs to a single course, it would be worthwhile to link the tasks of courses in different languages in this way.
- **Multiple OS support:** The software environment has also changed over the years, including the operating system of the students-managed virtual machines. After an operating system upgrade, it is necessary to check that the tasks can still be performed in the same way. Sometimes, a new tool may become available, or an old utility may become obsolete or unavailable, which may also affect the verification of the students' readings.
- **Task verification:** To automatically check whether the solution we have in mind is still feasible on a new system, we only need to record the solution we have in mind. If, when running the solution on a new system, the verification of the execution of the task finds an error, it is an indication to us that we need to rework it for the new system. In this way, the maintenance of tasks can be partially automated.
- **Solution evaluation:** The positive feature of the old system, namely its ability to perform automatic evaluation, even under exam conditions, must be retained [14]. The ProgCont system has repeatedly demonstrated the advantages of automatic evaluation, primarily because it enables a clearly objective evaluation without the need for teacher intervention. At the same time, care must be taken to ensure that the feedback is useful to the student [15], [16], [17], for example in the case of an incorrect solution, it should help to identify the causes of the error, as we have done in several experiments in the ProgCont system, for example, by introducing the annotation method [18], [19].
- **Immediate feedback:** We must also retain the feature of our previous system, whereby immediate feedback allows students to make corrections to incorrect answers. [20].

5.2 Management Features:

- **Different instructors roles:** From time to time, new teaching colleagues have been involved in teaching the subjects concerned and thus in using our software. This has created a user community that is not made up of developers but active users of the system. In order to make it as easy as possible for them to use the system, it is also necessary to separate the tasks and tools of the tutor and the administrator. Not all instructors need all the tools; if someone only wants to manage the tasks of his/her own students, he/she should be provided with a dedicated interface.
- **Student group management:** The management of separate groups of students is also a useful tool for monitoring student progress.
- **Exam and practice sessions, lab assessments:** The collection of exercises is, of course, not only useful for conducting tests, it is also important that students have the opportunity to practice under the same conditions, whether they are preparing independently at home or working in a computer lab.
- **Advanced statistics:** Measuring student performance in the three use cases mentioned above, and in a way that can be interpreted separately for each student group, provides effective assistance to instructors in tracking their students' progress [21], [22].

Conclusions

In this article, we summarised the principles and functions of our system designed for Assessment System for IT Configuration Tasks, how it helped to solve the challenges caused by the pandemic, and how it was later integrated into the teaching of IT Security and Operating Systems lab courses. The data collected since the start of development, while not specifically presented in a form conducive to pedagogical analysis, may be useful in highlighting future possibilities. The simple study carried out as an example also provides some help for educators. Although the system was vital at the outset, it is clearly now ripe for revision and rethinking, the foundation stone of which has been laid by the formulation of the requirements for our planned new system. We believe it is worth maintaining the enthusiasm we have invested in developing our own software and taking our system forward into the coming semesters by filling the gaps highlighted.

References

[1] Hu, J., Meinel, J., and Schmitt, M.: Tele-lab IT security: an architecture for interactive lessons for security education. In Proceedings of the 35th SIGCSE technical symposium on Computer Science Education (SIGCSE '04). Association for Computing Machinery, New York, NY, USA, 2004, pp. 412-416, https://doi.org/10.1145/971300.971440

[2] Lobb, R., and Harlow, J.: Coderunner: A tool for assessing computer programming skills. ACM Inroads 7.1 (2016): pp. 47-51, https://doi.org/10.1145/2810041

[3] Croft, D., and England, M.: Computing with CodeRunner at Coventry University: Automated summative assessment of Python and C++ code. Proceedings of the 4th Conference on Computing Education Practice. 2020, https://doi.org/10.1145/3372356.3372357

[4] Kádek, T. and Biró, P.: Next Generation of Assessment Evaluation System for IT Configuration Challenge, 2024 IEEE 3rd Conference on Information Technology and Data Science (CITDS) Proceedings, Debrecen, Hungary, 2024, pp. 102-106, https://doi.org/10.1109/CITDS62610.2024.10791350

[5] Biró, P. and Kádek, T.: Automatic Evaluation of Programming Tasks at the University of Debrecen. In INTED2020 Proceedings, 2020, pp. 3522-3527, https://doi.org/10.21125/inted.2020.0994

[6] Rodríguez-del-Pino, JC., Rubio Royo, E., Hernández Figueroa, Z.: A Virtual Programming Lab for Moodle with automatic assessment and anti-plagiarism features. (2012) Accessed on: January 21, 2025, from https://scispace.com/pdf/a-virtual-programming-lab-for-moodle-with-automatic-4k7t5cx2jm.pdf

[7] Pastor, V., Díaz G., and Castro, M.: State-of-the-art simulation systems for information security education, training and awareness, IEEE EDUCON 2010 Conference, Madrid, Spain, 2010, pp. 1907-1916, https://doi.org/10.1109/EDUCON.2010.5492435

[8] Kim, BH., Kim, KC., Hong, SE. et al.: Development of cyber information security education and training system, Multimedia Tools Applications 76, 2017, pp. 6051-6064, https://doi.org/10.1007/s11042-016-3495-y

[9] Biró, P., Kádek, T., and Pánovics, J.: Facilitating the use of the ProgCont system for novice programmers with the test case annotations, ICERI2021 Proceedings, 2021, pp. 4951-4956, https://doi.org/10.21125/iceri.2021.1133

[10] Balázs, P., Biró, P., Kádek, T., Kósa, M., and Pánovics, J.: Mathability and exploring mathematical skills of programmers with exercises' annotations. In: Nikodem, Jan; Klempous, Ryszard (eds.) 12th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2021): Proceedings IEEE 2021, pp. 347-350

[11] Biró, P., and Kádek, T.: The Mathability of Computer Problem Solving with ProgCont. Acta Polytechnica Hungarica 19(1), 2022, pp. 77-91, https://doi.org/10.12700/APH.19.1.2022.19.6

[12] Hentea, M., Dhillon, H. S., and Dhillon, M.: Towards Changes in Information Security Education, Journal of Information Technology Education: Research. 5 (1), 2006, pp. 221-233, Informing Science Institute. https://doi.org/10.28945/2954

[13] Douce, C., Livingstone, D., and Orwell, J.: Automatic test-based assessment of programming: A review." Journal on Educational Resources in Computing (JERIC), 2005, 5(3), pp. 1-13, https://doi.org/10.1145/1163405.1163409

[14] Luck, M., and Joy, M.: A secure online submission system. Software: Practice and Experience, 1999, 29(8), pp. 721-740, https://doi.org/10.1002/(SICI)1097-024X(19990710)29:8<721::AID-SPE257>3.0.CO;2-0

[15] Watling, C. J., and Ginsburg, S.: Assessment, feedback and the alchemy of learning. Medical education, 2019, 53(1), pp. 76-85, https://doi.org/10.1111/medu.13645

[16] Hattie, J., and Timperley, H.: The power of feedback. Review of Educational Research, 2007, 77(1), pp. 81-112, https://doi.org/10.3102/003465430298487

[17] Nicol, D. J., and Macfarlane-Dick, D.: Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. Studies in Higher Education, 2006, 31(2), pp. 199-218, https://doi.org/10.1080/03075070600572090

[18] Biró, P., Kádek, T., Kósa, M., and Pánovics, J.: A New Method to Increase Feedback for Programming Tasks During Automatic Evaluation. Acta Polytechnica Hungarica 19(9), 2022, pp. 103-116, https://doi.org/10.12700/APH.19.9.2022.9.6

[19] Biró, P., and Kádek, T.: Task annotation capabilities of the ProgCont system in practice, In 16th International Conference on Economics and Business: Challenges in the Carpathian Basin: Global Challenges - Local Answers. Interdependencies or Globalisation? Cluj-Napoca, Editura Risoprint, ICEB 2023, pp. 318-327, Accessed on: January 21, 2025, from https://csik.sapientia.ro/content/2022-2023/felveteli_eredmenyek_2023/Challenges%20in%20the%20carpathian%20basin%2016th.pdf

[20] Wiggins, G.: Seven keys to effective feedback. On formative assessment: Readings from educational leadership (EL Essentials), 2016, pp. 24-35, Accessed on: January 21, 2025, from https://pdo.ascd.org/lmscourses/PD13OC005/media/FormativeAssessmentandCCSwithELALiteracyMod_3-Reading2.pdf

[21] Biró, P. and Kádek, T.: Comparing Student Results on a Programming Task During a Contest and a Practice Situation with the Help of the ProgCont System, 2024 IEEE Global Engineering Education Conference (EDUCON), Kos Island, Greece, 2024, pp. 1-6, https://doi.org/10.1109/EDUCON60312.2024.10578829

[22] Muñoz, J. L. R., Ojeda, F. M., Jurado, D. L. A., Peña, P. F. P., Carranza, C. P. M., Berríos, H. Q., Vasquez-Pauca, J. M.: Systematic review of adaptive learning technology for learning in higher education. Eurasian Journal of Educational Research, 2002, 98(98), pp. 221-233, Accessed on: January 21, 2025, from https://ejer.com.tr/manuscript/index.php/journal/article/download/707/99/1292