# An Overview of Computer Assisted Learning Systems in Computer Science

**Madalina Pana[1] and Alin Zamfiroiu[2]**
[1)] *Bucharest University of Economic Studies, Bucharest, Romania*
[2)] *Bucharest University of Economic Studies, Bucharest, Romania and National Institute for Research and Development in Informatics – ICI, Bucharest, Romania*
E-mail: madalina.pana@csie.ase.ro; E-mail: alin.zamfiroiu@csie.ase.ro

**Please cite this paper as:**
Pana, M. and Zamfiroiu, A.,2022. An Overview of Computer Assisted Learning Systems in Computer Science. In: R. Pamfilie, V. Dinu, C. Vasiliu, D. Pleșea, L. Tăchiciu eds. 2022. *8th BASIQ International Conference on New Trends in Sustainable Business and Consumption.* Graz, Austria, 25-27 May 2022. Bucharest: ASE, pp.656-663.
**DOI: 10.24818/BASIQ/2022/08/087**

### Abstract

Having the pandemic restrictions applied almost worldwide (including Romania), the educational system was necessary to be changed, so all activities have moved online. The main objective of this paper is to highlight the importance of eLearning tools and propose two methods that can assist students during their educational process. Both methods are inspired by the Computer Assisted Learning System (CALs) literature, and they are based on the student's behaviour analysis and providing adaptability to it. The first component is an example of a Computer Adaptive Test System (CATs) which was built using the Item Response Theory (IRT) methodology and its scope is to help students to find what is their understanding level for a subject and to answer questions which are corresponding to their current learning progress. Furthermore, a lot of research suggests that testing your knowledge right before you start learning can improve your performance during the exam itself, more so than if you used the same time to read the subject. The second component that we propose is a users' recognition system which has the objective to identify if a student is cheating during an exam from the way he/she is typing. The key for this type of system is to define an algorithm that compares the keystroke dynamics. Overall, this study contributes to the scientific literature by revealing mechanisms for adapting to user behaviour and the positive effects that occur using this type of application.

### Keywords

eLearning, CALs, CAT, assisted, adaptive, keystroke dynamics.

**DOI: 10.24818/BASIQ/2022/08/087**

### Introduction

During the last decade, the interest in various CALs (Computer Assisted Learning System) has increased and now, going throw the pandemic restrictions, schools are more using online platforms which are integrated with learning resources tools for audio, text, video as well as chat, forum discussions, email, and assessments tools (Mtebe and Raisamo, 2014). Thus, the eLearning method is more complex than traditional one, since it is not restricted by time and space, however one does not exclude the other. In fact, both methods could lead to a positive and better result. The student profile is a portrait analysis based on big data and labelling. The scope is to collect, process, and analyse the data generated by the students' behaviours. According to the theory of behavioural psychology, the use of the student's profile to analyse the data on student's behaviour can reflect the student's characteristics and psychodynamics (Liang, et al., 2017).

This research proposes two ways of building a computer assisted learning system. The first one is an adaptive assessment based on the students' responses and feedback. The second one is an anti-cheating solution because it is concentrating on validating the user's identity during the exam by comparing it with his profile during the seminar activities. The structure of this paperwork is systematized into five sections. The second section is a literature review dedicated to the proposed two methods of Computer Assisted Learning. The third section establishes the methodological framework. The fourth section presents the proposed solutions and highlights results and discussions. The fourth section draws the conclusions.

## 1. Review of scientific literature

A computer assisted learning system (CALs) can be viewed as an adaptive learning system, because one of the requirements is to change its actions and provide learning content and methods for every student corresponding to their level and behaviour. These types of eLearning platforms should focus on the personalized and adaptive learning style of learners rather than just content delivery (Rani, Nayak and Vyas, 2015). Thus, we can say that user modelling system is the kernel of an adaptive learning system (Nguyen, 2014).

Adaptive testing algorithms, also known as CATs (Computer Adaptive Tests) or CAAs (Computer Adaptive Assessments), have a relatively recent practical implementation, but it seems that the number of uses is increasing significantly in the United States compared to the rest of the world. This type of assignment is different from one student to another, because it is adjusted based on their abilities and knowledge (Molins-Ruano, et al., 2014). Pluralsight has become a well-known company through its testing algorithms, which has improved a lot over time. These algorithms are based on the CAT principle, more precise the test changes in a dynamic way depending on the answers that the tester offers (Figure no. 1).

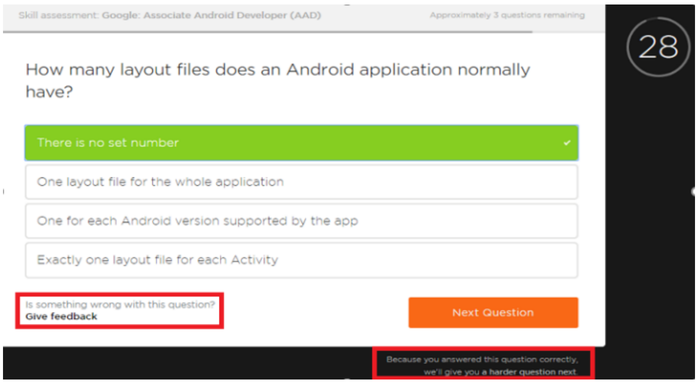


**Figure no. 1. Pluralsight test**
*Source: Pluralsight, 2022.*

From another point of view, not only could learning or testing be adaptive, but plagiarism testing methods could also be adapted based on user's typing behaviour. For example, Microsoft offers the opportunity for any student to take the exam at home, but only under certain conditions. First, the candidate is obliged to install on his/her computer a Microsoft licensed system through which he/she can take the exam. This system blocks access to any other applications and documents on the computer, leaving only the exam available. At the same time, the system creates a video conference with a supervisor and the candidate is obliged to keep the camera and microphone on during the entire session. Thus, a supervisor is responsible for verifying each candidate does not use alternative sources to cheat. This type of approach used by Microsoft cannot be implemented also in the education system due to the unequal ratio between the number of students and the number of teachers.

A part of high education institutions is paying attention to remote proctoring technologies, but they are also considering using an application which validates the keystroke dynamics of the user. For example, TypingDNA is a Romanian start-up, and their concept is about how to identify a person based on their typing signature. Their algorithm works with a minimum of 4 characters to over 100 characters and it uses two methods. One method can be used for different texts (any text pattern) and the other method has the condition that texts need to be identical, even if there could exist some typos that the users will correct. Also, the longer the text is, the better the algorithm works, by avoiding getting a high false acceptance/rejection rate.

## 2. Research methodology

Our research objective is to perform an analysis of literature in the field of computer-assisted education. We chose to study two main branches: adaptive testing and typing behaviour analysis during exams. Our motivation was that there is no complex product on the market that encompasses multiple functionalities to support computer-assisted education. Thus, between 2018 and 2019, our focus was to implement a solution for adaptive testing, which could be used by students for accelerated learning. This type of testing is not intended to create a ranking of student grades, but it is used by students to observe their level of learning by testing. During 2020-2021 our goal was to implement an anti-cheating solution for online evaluation. The scope was to observe how a student is typing during the exam and to compare this behaviour to the previous three seminars.

## 3. Results and discussion

### 3.1. Proposed solution for the adaptive online assignments

We took the scenario where the test contains only 20 questions, and where these questions are grouped according to their difficulty: low, medium, or high. The requirements are that the professor must create at least one subject that students can access and within that subject, the professor must introduce at least 10 questions of medium difficulty, at least 17 questions of low difficulty and at least 17 questions of high difficulty. Even though the difficulty of the question is set by the professor, the system is also recording how the students are answering every question and it is classifying again the questions using the number of correct and wrong answers. The adaptive assessment algorithm can be defined by the following schemes (Figure no. 2, Figure no. 3):

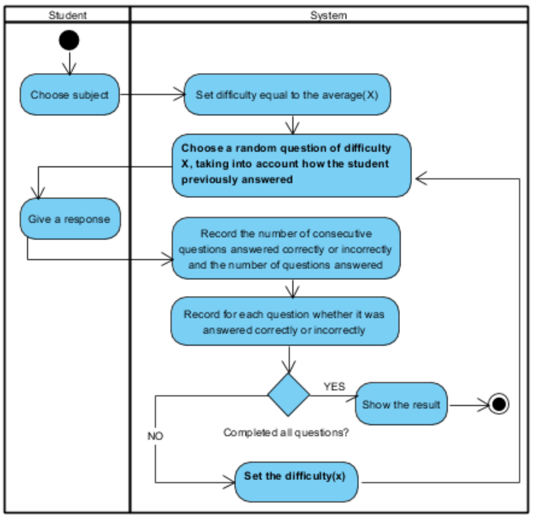


**Figure no. 2. Activity diagram for performing the test**
*Source: Diagram implemented based on the algorithm created*

The principle, on which this algorithm is based on, is recursion, as it can be seen in the diagram above. The first step that a student must take is to choose the subject from which he/she wants to be tested. Once chosen, the system will look for a medium difficulty question and set it as the starting difficulty. Subsequently, if the student answers the questions, the system will execute the function until the student has completed all the questions in the test (e.g., 20). The next step is to change the difficulty of choosing the next user-appropriate question. If the student answered three consecutive questions correctly in the current difficulty, then the difficulty will be increased, otherwise, if the student answers three consecutive questions incorrectly, then the difficulty will be reduced. If the student does not meet the above conditions, the difficulty is kept at the same level. The last step in each iteration is to search for the next question. In this case, the difficulty for the student will be considered as follows:

a) If the student has three consecutive correct answers, the next question from a higher difficulty will be chosen at random, but it will be a question that more than 70% of the students knew how to answer correctly. If there is no question with a percentage above 70% in that category, one with a percentage between 30% and 70% will be randomly searched. Otherwise, a question will be chosen at random.

b) If the student has three consecutive wrong answers, a question with a lower difficulty will be chosen at random, at the same time considering that it is a question to which more than 70% of the students knew how to answer correctly. As in the previous case, if no questions are found, the next 30% -70% difficulty range will be searched. If it is still not found then, a question will be searched randomly from the new difficulty category. The same algorithm from point a) applies.

c) If there are two consecutive wrong answers, then the difficulty will be kept, a question will be looked for which over 70% of the people knew how to answer. If none are found in that range, it will be searched in the range of 30% -70%. If it is found, then that question will be returned, and if not, a question of that difficulty will be chosen at random. The algorithm from point a) will be applied.

d) If the student has two consecutive correct answers, the difficulty will be kept, but the algorithm will try to find a more complicated question, by the simple fact that it will look for a question to which more than 70% of the people answered incorrectly. If there are no questions that meet these requirements, then one with a 30% -70% error rate will be searched. If it is found, it will be returned, otherwise a random question will be chosen with that degree of difficulty.

e) If the student has the consecutive wrong and correct answers equal to zero, then it will proceed as in the previous case from point c).

f) If the student has only one correct answer or only one wrong answer, then the algorithm will randomly extract the next question from the same difficulty, but with the percentage of correct answers in the range of 30% -70%. If no question is found that satisfies this condition, one is chosen by the random function.

If the student has passed all the questions, then the results during the test will be recorded and displayed. A very important aspect of the adaptive assessment algorithm is how the level is determined. We decided after some analyses performed on other specialized tests as IRT (Item Response Theory), that this algorithm distinguishes four levels of knowledge, namely: beginner, intermediate, upper-intermediate and advanced (Abdullah, 2003). In this case, to be able to estimate for each student where he/she is situated in terms of knowledge, the Levels values for that student are extracted and interpreted after an analysis carried out following some possible case examples. For example, if a student answers all the questions correctly, he/she deserves the title of "advanced", having at the end more correct than wrong answers for the questions of medium and high difficulty. Conversely, if a student makes a mistake in all the questions, he will have more wrong than correct answers to all the questions of medium and low difficulty, and his level can be classified as "beginner".

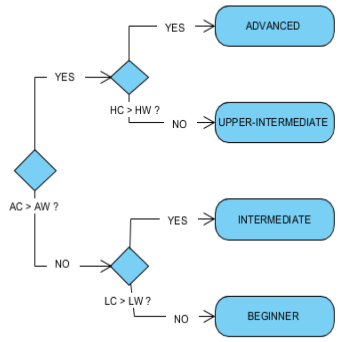


**Figure no. 3. Student level determination diagram**

*Notes: AC = number of correct answers the student gave to each medium difficulty question; AW = number of wrong answers the student gave to each medium difficulty question; HC = number of correct answers the student gave to each question of high difficulty; HW = number of wrong answers the student gave to each high difficulty question; LC = number of correct answers the student gave to each low difficulty question; LW = number of wrong answers the student gave to each low difficulty question.*

*Source: Diagram implemented based on the algorithm created*

In the end, an indicative score is calculated from all the questions that the student answered correctly, considering that a low difficulty question has 5 points, a medium difficulty question has 10 points, and a high difficulty question has 15 points.

The result of such an application can be seen in communication between students and teachers. The experience that the user (student) has during the evaluation is a better one by the simple fact that he/she receives only questions relevant to his/her level of knowledge. At the same time, the adaptive tests are more advantageous than the traditional tests, because the student's answers will be considered during the evaluation, the result being a much more concrete and precise score.

### 3.2. Proposed solution for online evaluation

The main objective of this system is to analyse the student’s typing behaviour in order to authenticate him/her only under certain conditions. This system describes a product that can be used during class hours as an IDE (Integrated development environment), where programs can be written in a programming language (e.g., JavaScript). Considering everything presented so far, before using this application, it is necessary to meet the following requirements:

- The professor needs to create a session which can be a seminar, or an exam and the students can access it using a password that is generated by the system.

- Each student needs to participate in a minimum of three seminars in order to be eligible to take the exam.

- The professor needs to close each session at the end of each class hour. Otherwise, the students can continue to write the code, which could affect the results.

If all these requirements have been met, then the student can take the exam, and at the end he will receive an analysis based on the way he/she typed, compared to the way he/she used to type during the seminars (Douhou and Magnus, 2009). In order to analyse the identity of a user, it is mandatory to compare two types of profiles: “training profile” which is created during the seminar and “test profile” which is created during the exam, one that needs to be validated.

The first algorithm is being used during seminars and exams and has the role of collecting the typing behavioural data and saving it to the database. The indicators that are measured are the following:

- timestamp for each keypress;

- how the student is choosing to use capital letters (Caps Lock or Shift);

- how the user chooses to delete the text (Delete or Backspace);

- how often the student is using the following commands: cut (Control and X), copy (Control and C); paste (Control and V), run the code (Control and B).

These indicators are measured during each session and saved to the database in real time. At the end of each session, the algorithm of the student profiling starts, and this result too is saved in the database. In the same manner, during the exam, the collecting data process is running again and at the end of the exam the profile is created. This profile is compared to the last three profiles in order to validate the student. If the profiles are mismatched, then the student is not identified, and he receives 0 points. In case the authentication is successful, then the student’s exam can be evaluated by the professor (Zamfiroiu and Ciurea, 2017).

a) Algorithm for creating the student’s profile

The professor needs to close the session a the end of each seminar or exam and then the system starts to analyse the collected data and creates the profile. Starting with the first characteristic, the distance between each keypress is calculated and it is taking into consideration if it is less than or equals to 6 seconds, because anything that takes more than 6 seconds will be considered as a pause. This characteristic is called TS for a data set that contains only distances less than or equal to 6 seconds, and the average will be calculated using the following formula:

$$TS = \frac{\sum_{i=0}^{n} x_i}{n}, x_i \leq 6 \text{ seconds (distances)}, n = \text{number of distances} \quad (1)$$

For example, hypothetically, we have two students with the following data sets for typing distances: *{2 seconds, 2 seconds, 6 seconds, 2 seconds*} and *{3 seconds, 3 seconds, 3seconds, 3seconds}.* As can be seen in the figure no. 4, the average on the first data set is equals to 3, but it is not entirely representative because in the same manner we have the second student that has all the distances equals to 3 seconds, so in the end, their average time is the same:

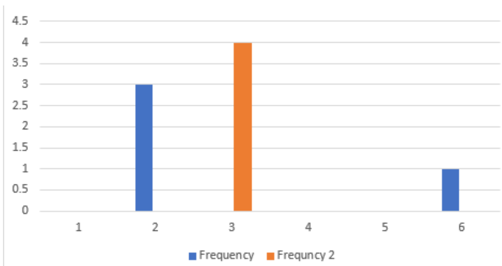


**Figure no. 4. Distribution graphic example**
*Source: Graphic implemented based on the example taken*

In this situation, the system applies the histogram principle, and the distances are divided according to the frequency of occurrence. The algorithm uses three intervals: (0,2], (2,4], (4,6] to distribute the distances. At the end, when the system is calculating the average, it is taking into consideration only the distances from the interval that have the highest frequency.

The second characteristic is called DB and it shows in percentages how many times the student used Delete, comparing to the Backspace during the entire session. The third characteristic is called BD and it uses the same mechanism, only vice versa. In the same manner are calculated the next characteristics called CS (how many times the student uses caps instead of Shift) and SC (vice-versa). For the last characteristics (CX, CC, CV, CB) regarding how many times the student used Control+X, Control+C, Control+V, Control+B, the system divides the frequency of occurrence by the time spent for typing. Having all these characteristics calculated, a profile will be created containing a list of the values resulted during a specific session. To summarize, hypothetically, in a Nine-dimensional space, a profile has as coordinates the following characteristics: {TS, DB, BD, CS, SC, CX, CC, CV, CB}.

b) Algorithm for comparing profiles

The first step is to get the last three profiles created during the seminars (ps1, ps2, ps3) and current profile from the exam (pe). In general, to compare two profiles (p1, p2), the algorithm calculates the Euclidean Distance using the following formula:

$$dist(p1,p2) = \sqrt{\begin{matrix}(TS_1 - TS_2)^2 + (DB_1 - DB_2)^2 + (BD_1 - BD_2)^2 + (CS_1 - CS_2)^2 + (SC_1 - SC_2)^2 \\ +(CX_1 - CX_2)^2 + (CC_1 - CC_2)^2 + (CV_1 - CV_2)^2 + (CB_1 - CB_2)^2\end{matrix}} \quad (2)$$

Where: *{$TS_k$, $DB_k$, $BD_k$, $CS_k$, $SC_k$, $CX_k$, $CC_k$, $CV_k$, $CB_k$}, k=1,2* are the profile's characteristics.

Based on the Euclidean distance results, the algorithm sets a score that represents whether the user identity is validated or not. From the beginning, this score is set to 100% and the system decreases this value if some of the rules are not applied. These rules are used for the distances between seminar profiles: S = {dist(ps1, ps2); dist(ps1,ps3); dist(ps2,ps3) } and also for the distances between exam profile and each seminar profile: E = {dist(pe, ps1); dist(pe, ps2); dist(pe, ps3)}. For each set of values S and E the average is calculated (seminarAvg, examAvg), and the minimum and the maximum values are determined (seminarMax, seminarMin, examMax, examMin). Because the students do not type the same code during the exam as they did during the seminar, the algorithm should be more permissive and should not be sensitive to small changes. Hence, the rules that the algorithm is using are the following:

- If the examAvg is greater than seminarMax, then the algorithm cannot validate the identity of the student and decreases the score by 20%.

- If the examAvg is greater than seminarAvg the system can validate the identity, but with low accuracy. Here the algorithm is decreasing the score by 15%.

- If the examAvg is less than seminarAvg, it means that the algorithm can validate the student's identity.

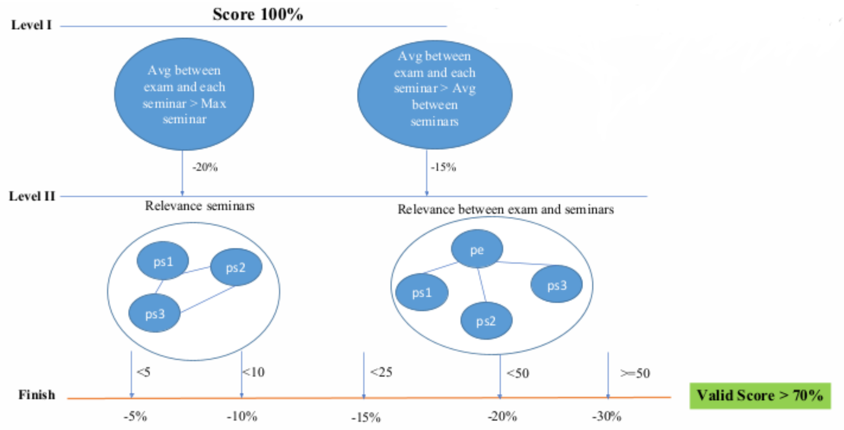


**Figure no. 5. Representation of the rules to validate identity**
*Source: Schema implemented based on the algorithm created*

Another important aspect is that these three rules may be applied to unrepresentative data sets, but then the result cannot be taken into consideration. It is necessary to evaluate the ratio between the maximum value and the minimum value in both data sets S and E to check the relevance of the seminars and the relevance of the exam compared to each seminar. The rules described above (Figure no. 5) were adjusted based on the results that we got so far:

- If $\frac{valMax}{valMin} \geq 0 \text{ and } \frac{valMax}{valMin} < 5$ , then the score is reduced by 5%;

- If $\frac{valMax}{valMin} \geq 5\ and\ \frac{valMax}{valMin} < 10$ , then the score is reduced by 10%;
- If $\frac{valMax}{valMin} \geq 10\ and\ \frac{valMax}{valMin} < 25$ , then the score is reduced by 15%;
- If $\frac{valMax}{valMin} \geq 25\ and\ \frac{valMax}{valMin} < 50$ , then the score is reduced by 20%;
- If $\frac{valMax}{valMin} \geq 50$ , then the score decreases by 30%;

Considering that the behaviour can change over time, the process always needs to recollect the data to rebuild a more accurate data set. Thus, if the score at the end of the exam is equal to 90%, then the algorithm will mark that exam profile to be considered as well as a training profile for the next evaluation.

To test the quality of this system, three students were asked to use this application (Table no. 1). Each of them completed three seminars and then started the exam ("Authorized person"), but they were also switched (replaced) in between for the second exam ("Unauthorized person"). When all the students tried to validate their identity, only student A was rejected. However, the second time, another person took the exam in their place, the system did not validate the identity for any of them.

**Table no. 1. The results of the test for all students**

| | **Student A** | **Student B** | **Student C** |
|---|---|---|---|
| **Authorized person** | ✗ 70% | ✓ 75% | ✓ 80% |
| **Unauthorized person** | ✓ 70% | ✓ 60% | ✓ 70% |

*Source: Table implemented based on the algorithm tested*

For example, having all the profiles of the student B, the system calculates the Euclidean distances and analyse those values to get the final score 75% as follows (Table no. 2):

**Table no. 2. The results of the test for student B**

| **Seminars** | | **Exam** | |
|---|---|---|---|
| **dist(ps1,ps2):** | 0,1606780541 | **dist(pe,ps1):** | 0,0604770062 |
| **dist(ps1,ps3):** | 1,00200201 | **dist(pe,ps2):** | 0,1745344426 |
| **dist(ps2,ps3):** | 1,0131008805 | **dist(pe,ps3):** | 1,0030986262 |

*Source: Table implemented based on the algorithm tested*

Based on the results of the students A, B, and C, we can only validate the solution using the FAR (False Acceptance Rate) and FRR (False Reject Rate) indicators (Nguyen, Le and Le, 2010):

$$FRR = \frac{Total\ false\ rejection}{Total\ attempts} = \frac{1}{6} * 100 = 16,(6)\% \qquad FAR = \frac{Total\ false\ acceptance}{Total\ attempts} = \frac{0}{6} * 100 = 0\% \tag{3}$$

By testing the algorithm using this set of rules, we obtained the most favourable case, FAR=0%, and FRR=16,66%. In this paper (Zamfiroiu, et al., 2020) a detailed analysis is made on several user detection techniques, which are classified into two categories: static and continuous authentication. The static authentication behaviour refers to how the user types the username and password, but the results are not as good as those of continuous authentication thus, the values obtained by our algorithm are better than the ones from the first category. Hence, because continuous authentication uses the random text written by the user, it is more accurate than analysing the static one. The examples analysed for this category in this paper (Zamfiroiu, et al., 2020) are based on different methods such as using Euclidian distance and probabilities together or the method of nearest neighbour. All the methods are using keyboard dynamics and mouse usage dynamics, so multiple user behavioural indicators are covered. As a result, for the nearest neighbour the FAR value is 3,17% and FRR is 0,03%. By comparing these with our results (16 and 0), there are still many possibilities to improve the algorithm as using more user characteristics (e.g., mouse dynamics). In addition, if the number of sessions (seminars and exams) is more and more increasing, then we will see a decrease in FRR and FAR indicators over time, which determines this algorithm to become more accurate.

## Conclusions

In this paper, our motivation was to build a complex eLearning system that can be used to assist students in their educational process. Statistics have proven, time after time, that we need to use new technologies to improve our quality of life, not just in educational field. In this regard, we have formulated two different components for computer-assisted education. A system that is based on assisted education, we see it as a puzzle that contains several pieces. In this paper, through the solution of detecting the user through the typing mode, we have the remote evaluation component that gives us the certainty that there are no chances to cheat. Also, through the adaptive assessment component, we offer an interactive way for students to learn and progress. But despite all these components, there are still missing puzzle pieces. Many studies showed that there is a strong relationship between learning style and eLearning and the learning outcomes are improved if the learning style motivates the students. For future work, we aim to implement the assisted eLearning system based on the framework described in this paper (El-Sabagh, 2021). In addition, we continue to investigate how computer technologies can improve the field of education in Romania.

**Acknowledgements**

This work was supported by project PN19370401 “New solutions for complex problems in current ICT research fields based on modelling and optimization”, funded by the Romanian Core Program of the Ministry of Research and Innovation, 2019-2022.

**References**

Abdullah, S.C., 2003. *Student Modelling by Adaptive Testing - A Knowledge-Based Approach*. [pdf] Available at: <https://www.cs.kent.ac.uk/pubs/2003/1719/content.pdf> [Accessed 6 March 2022].

Douhou, S. and Magnus, J.R., 2009. The reliability of user authentication through keystroke dynamics. *Statistica Neerlandica*, [online] 63(4), pp.432–449. https://doi.org/10.1111/j.1467-9574.2009.00434.x.

El-Sabagh, H.A., 2021. Adaptive e-learning environment based on learning styles and its impact on development students' engagement. *International Journal of Educational Technology in Higher Education*, 18(53), pp.1-24.

Liang, K., Zhang, Y., He, Y., Zhou, Y., Tan, W. and Li, X., 2017. Online Behavior Analysis-Based Student Profile for Intelligent E-Learning. *Journal of Electrical and Computer Engineering*, [online] 2017, pp.1–7. https://doi.org/10.1155/2017/9720396.

Molins-Ruano, P., González-Sacristán, C., Díez, F., Rodriguez, P. and Sacha, G.M., 2014. *Adaptive Model for Computer-Assisted Assessment in Programming Skills*. [online] https://doi.org/10.48550/ARXIV.1403.1465.

Mtebe, J.S. and Raisamo, R., 2014. A Model for Assessing Learning Management System Success in Higher Education in Sub-Saharan Countries. *The Electronic Journal of Information Systems in Developing Countries*, [online] 61(1), pp.1–17. https://doi.org/10.1002/j.1681-4835.2014.tb00436.x.

Nguyen, L., 2014. Evaluating Adaptive Learning Model. In: *2014 International Conference on Interactive Collaborative Learning (ICL)*. Dubai, United Arab Emirates: IEEE. pp.818–822. https://doi.org/10.1109/ICL.2014.7017878.

Nguyen, T.T., Le, T.H. and Le, B.H., 2010. Keystroke Dynamics Extraction by Independent Component Analysis and Bio-matrix for User Authentication. In: *Zhang BT., Orgun M.A. (eds) PRICAI 2010: Trends in Artificial Intelligence. PRICAI 2010. Lecture Notes in Computer Science*, vol 6230. (pp. 477-486) Publisher: Springer, Berlin, Heidelberg.

Rani, M., Nayak, R. and Vyas, O.P., 2015. An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage. *Knowledge-Based Systems*, [online] 90, pp.33–48. https://doi.org/10.1016/j.knosys.2015.10.002.

Zamfiroiu, A. and Ciurea, C., 2017. A model for users' profile recognition based on their behavior in online applications. *Economic Computation and Economic Cybernetics Studies and Research*, 51(2), pp.181-194.

Zamfiroiu, A., Constantinescu, D., Zurini, M. and Toma, C., 2020. Secure Learning Management System Based on User Behavior. *Applied Sciences*, [online] 10(21), p.7730. https://doi.org/10.3390/app10217730.

Pluralsight, 2022. *Pluralsight Test*, [online] Available at: <https://www.pluralsight.com> [Accessed 6 March 2022].

Typingdna, 2022. *Authenticate people by the way they type*, [online] Available at: <https://www.typingdna.com> [Accessed 6 March 2022].