Adaptive user interfaces based on behavioral analysis

Yevhen Artamonov[1,*,†], Tetyana Okhrimenko[1,2,†], Iurii Golovach[1,†], Daniil Krant[1,†], Andrij Radchenko[1,†], Kostiantyn Radchenko[1,†] and Taras Zaloznyi[1,†]

[1] *National Aviation University, Liubomyra Huzara Ave. 1, Kyiv, 03058, Ukraine*

[2] *State Scientific and Research Institute of Cybersecurity Technologies and Information Protection, Maksym Zalizniak Str., 3/6, Kyiv, 03142, Ukraine*

Abstract

The article reveals the possibilities of developing and applying adaptive interfaces that involve analyzing user behavior in e-learning systems. Methods of content adaptation based on personalization are a relevant direction in e-learning systems, because they allow taking into account the needs and requirements of users in an automatic mode. A user model is presented, which includes a comprehensive approach to representing both objective (progress indicators, performance, etc.) and subjective parameters (likes, user feedback, etc.), which allows personalizing the material and interface in accordance with individual requests and capabilities of users. The Levenshtein method determines the similarity between user behavior vectors and predefined learning templates, which allows the system to dynamically adjust data presentation scenarios to support the optimal learning trajectory. This method was used in the MS SQL Server online learning system and in a hardware-software complex for learning Braille. Changing the scenarios for displaying materials in accordance with a specific user category demonstrated significant improvements in learning success rates. Experiments showed that automatic switching of scenarios allowed the system to adapt to the level of learning material mastery by the user, which increased motivation and productivity. Each scenario of displaying educational materials provided for different levels of task complexity and depth of presentation of educational material, so changing scenarios during training added interactivity and increased the level of friendliness of the learning environment, which adapted to the pace and style of each user. System load assessment showed that the Levenstein method with minimal use of hardware resources solved the problem of measuring the similarity between user behavior and learning templates, which allows the system to adapt content in real time, taking into account the current productivity, interest and progress of the user. Minimization of requirements for system parameters allowed using this method in a hardware-software complex for teaching Braille to the visually impaired. The article separately highlights the tasks and requirements for implementing such systems: the complexity of development, requirements for algorithm performance, ensuring user data security, and constant updating of user models. Solving these problems is crucial for fully realizing the potential of dynamic content visualization in e-learning systems. The constant development and improvement of adaptive technologies have an impact on the future of education because technologies continue to develop, user requests are becoming more diverse, and the use of personalization is introduced into all spheres of life and education is no exception.

Keywords

adaptive interface, behavioral analysis, learning systems, medical diagnostic systems

1. Introduction

Adaptive user interfaces have long become a widespread phenomenon, which is used in almost all software and some hardware-software systems. Developers compete in methods and means of dynamically adjusting parameters for each user, recommendation systems have turned into templates and libraries that are used in the development of new software complexes, and affect the

CH&CMiGIN'24: Third International Conference on Cyber Hygiene & Conflict Management in Global Information Networks, January 24–27, 2024, Kyiv, Ukraine

* Corresponding author.
† These authors contributed equally.
eart@nau.edu.ua (Y. Artamonov); t.okhrimenko@npp.nau.edu.ua (T. Okhrimenko); iurii.golovach@gmail.com (I. Golovach); danbiil988@gmail.com (D. Krant); amentat777@gmail.com (A. Radchenko); radchenko.kn@gmail.com (K. Radchenko); taras.zaloznyi@gmail.com (T. Zaloznyi)
0000-0002-9875-7372 (Y. Artamonov); 0000-0001-9036-6556 (T. Okhrimenko); 0000-0001-9872-8144 (I. Golovach); 0000-0003-4601-2827 (D. Krant); 0009-0003-8288-171X (A. Radchenko);0009-0009-8056-6293 (K. Radchenko); 0009-0002-3012-4840 (T. Zaloznyi)

formation of content, a list of menu items, and product offers in accordance with individual user preferences, thus ensuring intuitive interaction with software systems [1]. In modern systems with adaptive functionality, the ability to analyze and respond to changes in user behavior has become a central issue that affects the formation of interface design, content, and available functionality [2, 3].

The main goal of the study is to implement a new method for determining user categories based on behavior analysis in adaptive e-learning systems. The method involves using the Levenshtein distance, which determines the similarity of rows, to assign a user to a certain category based on the analysis of the trajectory of passing through educational content. The implementation of this method in already operating systems with a constant flow of users will allow us to determine the feasibility of using this approach and the effect of its implementation. The experiment involves the use of two systems with different ideologies of building the structure of the program code and different deployment formats, so the use of a common method for determining the user category will better reveal its advantages and disadvantages.

The experiments will show whether the automatic switching of content display scenarios remains unnoticed by users by assessing the number of users' cancellations of automatic system decisions (the presence of such a fact will indicate confirmation of the correctness of the chosen approach to assessing user behavior, which correctly and expectedly affected the display of educational content). Successful verification of the operation of the Levenshtein method on real data and in real systems will allow us to reveal the potential of this method for use in adaptive information systems that base decision-making on user behavior.

The behavior of a computer system when used by a large number of people is one of the critical factors influencing the choice of a system on the software market, therefore, in research in the field of adaptive interfaces and learning systems, the speed and use of system resources are additional parameters in the evaluation of the proposed methods. That is why the need to provide each user with a comfortable working environment is often ignored, as a rule, remaining at the conceptual level, because the implementation of these methods increases the requirements for system characteristics.

The search for methods for adapting systems to the requirements and needs of users still remains open, and research is being conducted in the areas of providing tools for assessing individual user characteristics, creating user models, and determining the effectiveness of using systems for the tasks set. However, the fact that there is a tendency to automate learning processes (preparation of materials, assessment of the level of knowledge and mastered material, determination of learning trajectories, etc.) does not reduce the role of participation of teachers, instructors or trainers who use traditional tools and methods of learning.

Approaches to building systems with elements of adaptability have demonstrated their applicability in almost all areas of activity. More than 70 articles from various scientific disciplines describing multi-agent and multi-scenario system approaches were reviewed. Articles [4, 5] analyze the organization of training courses that are adapted to the needs and requests of users. In [6], the dissemination of public information using adaptation to user requests is separately considered. The potential of various approaches in decision support systems for healthcare is investigated in articles [7, 8]. Article [9] presents an assessment of the construction of a multi-agent training system. The analysis of scientific research confirmed the relevance of using adaptation to user behavior in software systems and indicated the feasibility of researching new methods for determining the user category depending on his behavior, because practical application distinguishes the described approaches, determining their real relevance and effectiveness.

2. Adaptive user interfaces in the e-learning systems

The use of dynamic interfaces and dynamic content generation in e-learning systems implies a transformation to modern education [10], because the adaptation of content in real-time to the behavior and progress of the student opens up opportunities for personalized learning taking into account the unique needs and preferences of each student. The dynamic nature of interfaces and content takes user interaction with e-learning systems to a new level and ensures that educational content will remain relevant and understandable for a longer time, regardless of the student's basic level of knowledge [11].

Adding new parameters of interface dynamics allows for the continuous development of educational platforms with constant feedback on user actions and reactions [12]. In e-learning systems, adaptation parameters include content presentation, navigation paths, interactive elements, the number and complexity of tasks, and the presence of interactive prompts. For example, if a student constantly faces the problem of finding the last read section of the educational material, then the interface can be adapted by providing a link to this section or its summary in the list of recently viewed materials. In another case, the student could not pass a certain category of tests because he did not have multimedia tools available on the computer, so for him these tests were excluded from the list of tasks.

Although each case of interaction with the system is unique, it is possible to combine users into groups and build standard solutions for adapting the system to the requirements of each group. This approach solves the problem of the complexity of the software implementation of adapting the system to the requirements of each individual user and introduces the concept of display scenarios for user groups.

A user model is used for combining users into groups, and the combination into groups occurs based on the coincidence of certain groups of parameters. The user model used in the developed systems has the following form:

$$U = \{OP, SP\},$$

$$OP = \{op_1, op_2, \ldots, op_m\}, SP = \{sp_1, sp_2, \ldots, sp_n\},$$

where *OP* is the set of objective parameters, *SP* is the set of subjective parameters.

The process of filling the user model should have the following classification features: implicit accumulation, individuality, dynamism, durability, and descriptiveness [10]. To adapt the system itself, its model is also introduced, which is divided into two sets of parameters:

– interface parameters (IP): these parameters cover all elements related to the presentation and interaction of educational resources. They include layout design, navigation structures, accessibility functions, and any other aspects that affect the user's direct interaction with educational content;

– functional parameters (FP): these parameters are related to the main functions of the e-learning system. They include content delivery mechanisms, performance tracking, feedback systems, and any other operational capabilities that support the learning process.

Then the e-learning system model can be represented as follows:

$$ESE = \{IP, FP\},$$

where

$$IP = \{ip_1, ip_2, \ldots, ip_k\},\ FP = \{fp_1, fp_2, \ldots, fp_l\},$$

where *IP* – the set of interface parameters, *FP* – the set of functional parameters.

To implement the method of dynamic visualization of educational content, it is necessary to evaluate the user parameters U and assign the user to a certain group. Then the system can personalize both the material and the interface according to the individual needs of a given user group. This approach offers several significant advantages and potential problems. The key feature of this method is the mechanism of automatic scenario change, which involves dynamic adjustment of content and learning paths (order of training material and/or test tasks) based on the analysis of the user's previous interaction and learning progress.

It was decided to use the Levenshtein distance to measure the similarity between the user's current behavior and predefined learning patterns, determining when to switch scenarios to optimize the learning process. The Levenshtein distance between two data sets is a measure of the difference between them, which quantifies the similarity, which in turn helps to decide which group to assign the user to and which scenario to use [13, 14].

The Levenshtein distance between two strings is the minimum number of single-character edits (insertion, deletion, replacement) required to transform one string into another [15]. Formally, it is defined recursively as follows:

$$D(i,j) = \begin{cases} max(i,j), & \text{if } min(i,j) = 0, \\ min\begin{cases} D(i-1,j)+1 \\ D(i,j-1)+1 \\ D(i-1,j-1)+[s[i-1] \neq t[i-1]] \end{cases}, & \text{otherwise,} \end{cases}$$

where $D(i,j)$ is the Levenshtein distance between the first i characters of string s and the first j characters of string t.

Consider an undirected connected graph with 10 vertices, represented by an adjacency matrix. The graph has the following structure:

$$\begin{array}{cc} & \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}.$$

We will treat the paths as strings of characters "0149", "0379", "159", "269", "0269", "389".

Example of Calculation of Levenshtein Distance Between paths 1 and 2 ("0149" and "0379") have few steps.

Step 1. Initialization: we start by initializing the dynamic programming matrix D with dimensions $(n+1) \times (m+1)$, where n is the length of the first string and m is the length of the second string. In this case, both strings have a length of 4, so the matrix will be 5×5.

$$\begin{array}{cc} & \begin{matrix} \square & 0 & 3 & 7 & 9 \end{matrix} \\ \begin{matrix} \square \\ 0 \\ 1 \\ 4 \\ 9 \end{matrix} & \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Step 2. Filling the Matrix: we fill in the matrix D by calculating the minimum cost of edit operations (insertion, deletion, substitution) required to transform the substring of the first string into the substring of the second string. We proceed cell by cell:

- $D(1,1)$ compares "0" with "0", which are the same, so the cost is 0. $D(1,1) = min(D(0,1)+1, D(1,0)+1, D(0,0)+0) = min(2,2,0) = 0$.
- $D(1,2)$ compares "0" with "03", the cost is 1. $D(1,2) = min(D(0,2)+1, D(1,1)+1, D(0,1)+1) = min(3,1,2) = 1$.
- $D(1,3)$ compares "0" with "037", the cost is 2. $D(1,3) = min(D(0,3)+1, D(1,2)+1, D(0,2)+1) = min(4,2,3) = 2$.
- $D(1,4)$ compares "0" with "0379", the cost is 3. $D(1,4) = min(D(0,4)+1, D(1,3)+1, D(0,3)+1) = min(5,3,4) = 3$.
- $D(2,1)$ compares "01" with "0", the cost is 1. $D(2,1) = min(D(1,1)+1, D(2,0)+1, D(1,1)+1) = min(1,3,2) = 1$.
- $D(2,2)$ compares "01" with "03", the cost is 1. $D(2,2) = min(D(1,2)+1, D(2,1)+1, D(1,1)+1) = min(2,2,1) = 1$.
- $D(2,3)$ compares "01" with "037", the cost is 2. $D(2,3) = min(D(1,3)+1, D(2,2)+1, D(1,2)+1) = min(3,2,2) = 2$.

- $D(2,4)$ compares "01" with "0379", the cost is 3. $D(2,4) = min(D(1,4) + 1, D(2,3) + 1, D(1,3) + 1) = min(4,3,3) = 3$.
- $D(3,1)$ compares "014" with "0", the cost is 2. $D(3,1) = min(D(2,1) + 1, D(3,0) + 1, D(2,0) + 1) = min(2,4,3) = 2$.
- $D(3,2)$ compares "014" with "03", the cost is 2. $D(3,2) = min(D(2,2) + 1, D(3,1) + 1, D(2,1) + 1) = min(2,3,2) = 2$.
- $D(3,3)$ compares "014" with "037", the cost is 2. $D(3,3) = min(D(2,3) + 1, D(3,2) + 1, D(2,2) + 1) = min(3,3,2) = 2$.
- $D(3,4)$ compares "014" with "0379", the cost is 3. $D(3,4) = min(D(2,4) + 1, D(3,3) + 1, D(2,3) + 1) = min(4,3,3) = 3$.
- $D(4,1)$ compares "0149" with "0", the cost is 3. $D(4,1) = min(D(3,1) + 1, D(4,0) + 1, D(3,0) + 1) = min(3,5,4) = 3$.
- $D(4,2)$ compares "0149" with "03", the cost is 3. $D(4,2) = min(D(3,2) + 1, D(4,1) + 1, D(3,1) + 1) = min(3,4,3) = 3$.
- $D(4,3)$ compares "0149" with "037", the cost is 3. $D(4,3) = min(D(3,3) + 1, D(4,2) + 1, D(3,2) + 1) = min(3,4,3) = 3$.
- $D(4,4)$ compares "0149" with "0379", the cost is 2. $D(4,4) = min(D(3,4) + 1, D(4,3) + 1, D(3,3) + 0) = min(4,4,2) = 2$.

Final Matrix:

$$\begin{pmatrix} \square & \square & 0 & 3 & 7 & 9 \\ \square & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 1 & 2 & 3 \\ 1 & 2 & 1 & 1 & 2 & 3 \\ 4 & 3 & 2 & 2 & 2 & 3 \\ 9 & 4 & 3 & 3 & 3 & 2 \end{pmatrix}$$

The final cell $D(4,4) = 2$ indicates that the Levenshtein distance between "0149" and "0379" is 2.

The Levenshtein distance was used in our adaptive learning systems to speed up calculations of user group membership and measured the similarity between the sequence of user interactions with the system and predefined patterns of expected behavior. This distance helps determine the need to change the content display scenario if there is a change in the type of user group membership, thereby ensuring that the level of complexity of the learning material is appropriate for each student.

3. Experimental evaluation of automatic scenario switching in adaptive learning systems

To evaluate the performance of the Levenshtein method in determining the user category, two systems were used:

1) the online learning system MS SQL Server, which was used to support the educational process in courses at a private educational center;

2) the hardware-software complex for teaching Braille, which was used in children's state institutions and individually was made in the form of a toy with a sound interface.

These systems have been used for more than 5 years and the current experiment is a continuation of the experiment, which is highlighted in the work [16], where the training complexes themselves are described in more detail. The new method for determining the user category replaced the previous method, which was based on the number of errors made during testing, and was the number of participants in the experiment:

1) 145 participants, when studying MS SQL Server, changed between two scenarios for 13 days (Table 1 shows the results of scenario changes);

2) 170 participants, when using the hardware-software complex for studying Braille, changed between three scenarios during 23 lessons (Table 2 shows the results of scenario changes).

Table 1
Scenario Usage in the E-Learning System for Studying MS SQL Server

Day	Scenario 1 at the beginning of the day	Change of scenario (end of the day)	Scenario 2 at the beginning of the day	Change of scenario (end of the day)
1	85	9	15	5
2	81	8	19	4
3	77	5	23	3
4	75	8	25	2
5	69	7	31	3
6	65	4	35	1
7	62	3	38	1
8	60	7	40	1
9	54	8	46	1
10	47	7	53	2
11	42	6	58	1
12	37	5	63	0
13	32	4	68	0
Result	28		72	

Table 2
Scenario Usage in the Hardware-Software Complex of Learning Braille

Lesson	Scenario 1 (beginning of the lesson)	Change of scenario (end of the lesson)		Scenario 2 (beginning of the lesson)	Change of scenario (end of the lesson)		Scenario 3 (beginning of the lesson)	Change of scenario (end of the lesson)	
		1 to 2	1 to 3		2 to 1	2 to 3		3 to 1	3 to 2
1	170	8	6	0	0	0	0	0	0
2	156	5	4	8	3	2	6	2	1
3	152	6	3	9	4	2	9	4	3
4	151	7	2	12	3	3	7	4	3
5	149	5	3	16	3	5	5	5	2
6	149	5	2	15	2	3	6	3	4
7	147	7	1	19	4	5	4	5	2
8	148	8	3	19	5	4	3	3	3
9	145	4	3	21	5	3	4	2	3
10	145	5	2	20	3	3	5	3	1
11	144	5	4	20	3	4	6	2	1
12	140	6	3	19	2	5	11	2	2
13	135	8	2	20	2	2	15	2	1
14	129	9	3	25	2	3	16	2	1
15	121	8	4	30	1	2	19	2	1
16	112	8	4	36	1	3	22	2	1
17	103	7	3	41	3	4	26	1	2
18	97	9	2	43	1	4	30	0	2
19	87	9	2	49	0	5	34	1	3
20	77	8	4	56	2	4	37	0	1
21	67	7	2	59	1	2	44	1	1
22	60	8	2	64	2	3	46	1	1
23	53	8	2	68	2	2	49	0	1
Result	45			73			52		

In the MS SQL Server e-learning system, users participated remotely, devoting an average of 3 hours per day to the training course. Braille was not studied every day, but the entire training course took no more than 2 months, and students were allocated approximately 1 hour per lesson.

The graphs (Figure 1 and Figure 2) with selected scenarios at the end of the day/lesson show the trend of scenario changes, which corresponds to the progress of the complexity of the tasks for most users and the minimum number of reverse scenario switching.

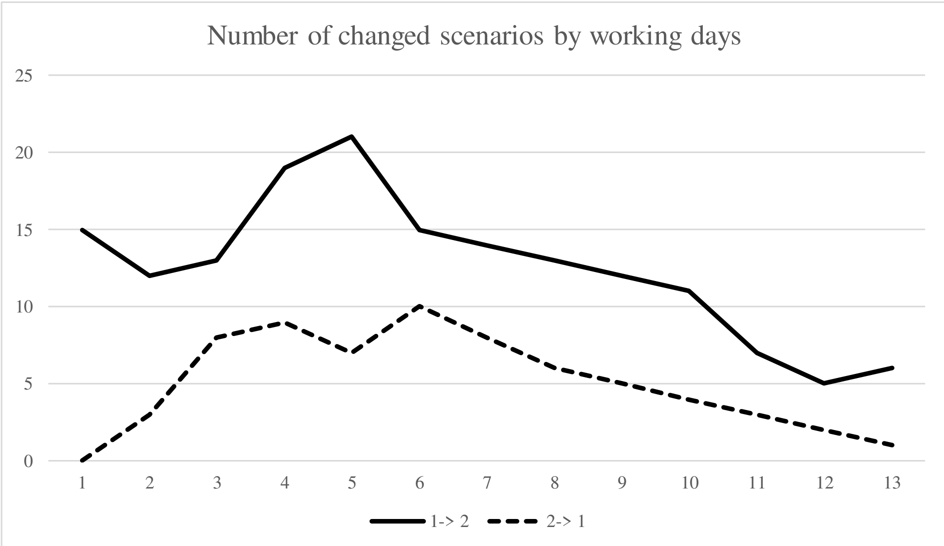

Figure 1: Number of changed scenarios by working days in the E-learning System for Studying MS SQL Server.

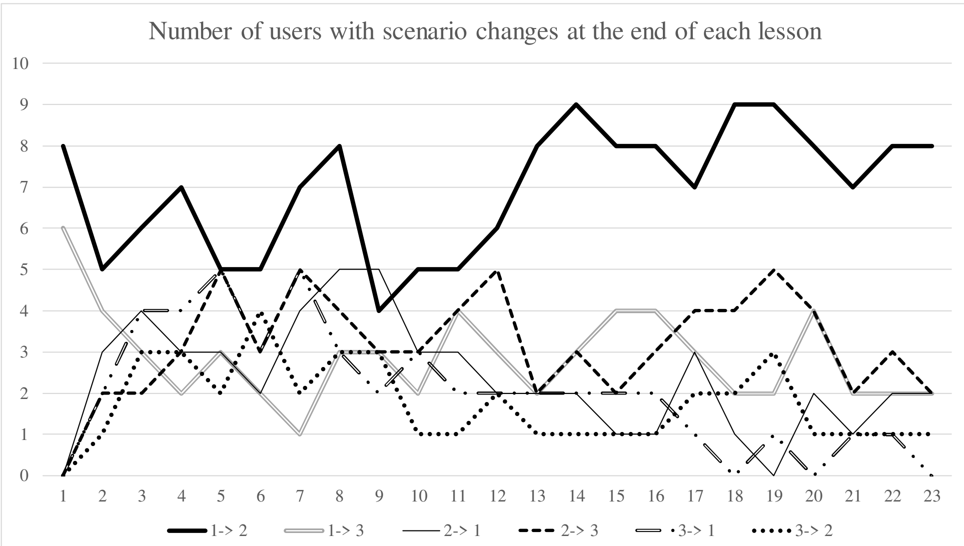

Figure 2: Number of changed scenarios by working days in the Hardware-Software Complex for Learning Braille.

During the experiments, the use of system resources was monitored during the operation of the new algorithm for determining the user category, which showed a minimal increase in resource requirements, which allowed the systems to be run on microcomputers (Braille learning).

Analysis of the obtained data shown:

1) in the e-learning system for studying MS SQL Server, the number of participants in Scenario 1 decreases over time, and in Scenario 2 increases (Figure 3), similarly, in Braille learning there is a gradual increase to Scenario 3 (Figure 4), which indicates that as the course progresses, more and

more users move from the initial scenario to the next one, which confirms the correctness of the work of determining the category by the new method;

2) the decrease in the number of transitions back to scenarios with less complexity also indicates the correctness of the choice of the user category, which corresponds to the logic of learning and adaptation to more complex content.

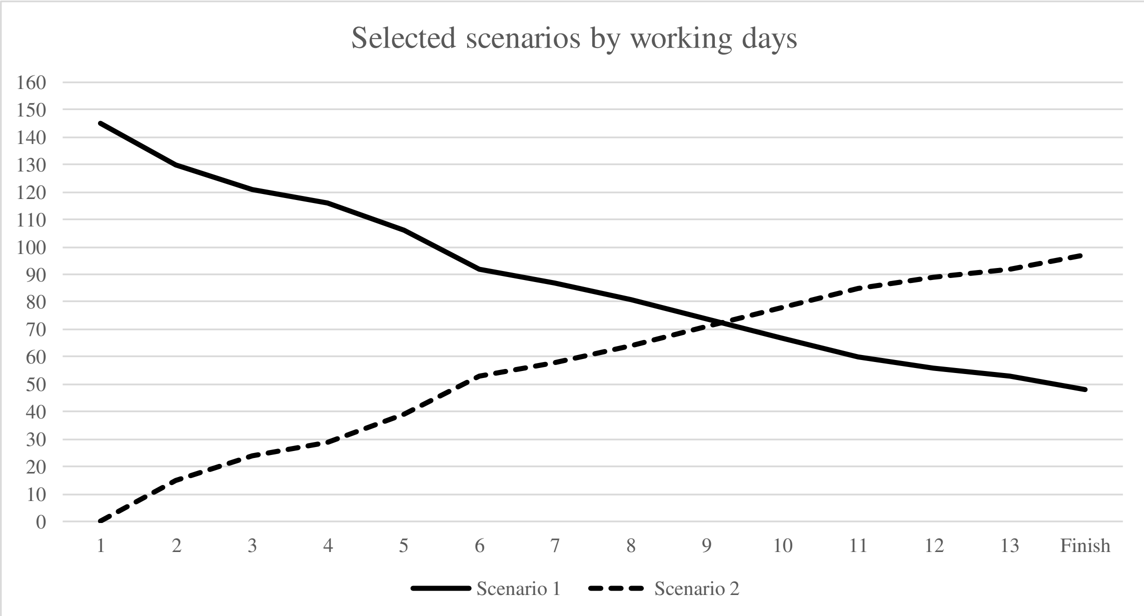

Figure 3: Selected scenarios by working days in the E-learning System for Studying MS SQL Server.

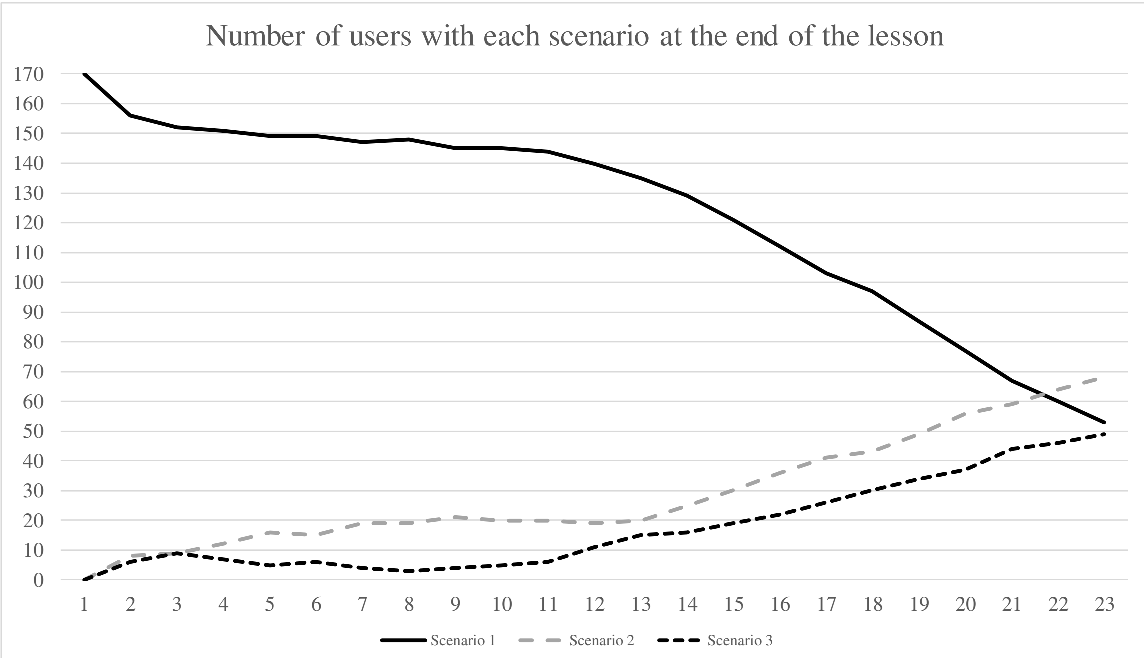

Figure 4: Selected scenarios by lessons in the Hardware-Software Complex for Learning Braille.

The results obtained when testing both systems emphasize the correctness of the choice of the method for evaluating user categories based on their behavior and the effectiveness of generating dynamic content in adaptive learning environments, which emphasizes the possibility of implementing such solutions in adaptive learning systems.

4. Conclusions

The practical implementation of the Levenshtein method for determining the user category by his behavior allowed to control the change of scenarios of displaying educational content in e-learning systems. This confirmed the correctness of the hypothesis of using the trajectory of the educational material among a set of objective and subjective parameters in the user model, which allowed to provision of individual settings of content and interface for each user. The Levenshtein distance with minimal use of hardware resources solved the problem of measuring the similarity between user behavior and learning patterns, which allows the system to adapt the content in real time, taking into account the current performance, interest, and progress of the user. Previous studies [16] have proven that dynamic adaptation of the display of educational materials improves learning outcomes by hiding too complex or too simple tasks, which maintains student motivation and contributes to better educational achievements.

Experiments have shown that automatic switching of content display scenarios remains unnoticed by users due to the small number of recalls of automatic decisions.

Continuous monitoring of user interaction with the system and adjustment of the user model depending on his behavior allowed to improvement the user's impression of the content, which is always consistent with the changing needs of the user, his state, or interest. The introduction of the structure of interface parameters and functional parameters provided a flexible approach to managing various components of the e-learning system and allowed to integration of the proposed approach in different forms of teaching and among different groups of users. At the same time, the implementation of adaptive systems involves the complication of development, the use of complex algorithms, an increase in the level of protection of user personal data, and constant refinement of the user model. The degree of use of the potential of dynamic visualization of content in e-learning systems depends on the solution of these tasks.

The constant development and improvement of adaptive technologies have an impact on the future of education because technologies continue to develop, user requests are becoming more diverse, and the use of personalization is introduced into all spheres of life and education is no exception.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] R. Kostyrko, T. Kosova, L. Kostyrko, L. Zaitseva, O. Melnychenko, Ukrainian market of electrical energy: Reforming, financing, innovative investment, efficiency analysis, and audit, Energies 14 (16):5080 (2021). doi: 10.3390/en14165080.

[2] I. Ostroumov, et al., Modelling and simulation of DME navigation global service volume, Advances in Space Research 69(8) (2021) 3495–3507. doi: 10.1016/j.asr.2021.06.027.

[3] S. Zhyla, et al., Practical imaging algorithms in ultra-wideband radar systems using active aperture synthesis and stochastic probing signals, Radioelectronic and Computer Systems 1(105) (2023) 55–76. doi: 10.32620/reks.2023.1.05.

[4] D. I. Chatzopoulou, A. A. Economides, Adaptive assessment of student's knowledge in programming courses, Journal of Computer Assisted Learning 26(4) (2010) 258–269. doi: 10.1111/j.1365-2729.2010.00363.x.

[5] J. Melesko, E. Kurilovas, Personalised intelligent multi-agent learning system for engineering courses, in: Proceedings of IEEE 4th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), IEEE, Vilnius, Lithuania, 2016, pp. 1–6. doi: 10.1109/AIEEE.2016.7821821.

[6] J. Yost, M. Dobbins, R. Traynor, K. DeCorby, S. Workentine, L. Greco, Tools to support evidence-informed public health decision making, BMC Public Health 14:728 (2014). doi: 10.1186/1471-2458-14-728.

[7] D. Schaumann, S. Breslav, R. Goldstein, A. Khan, Y.E. Kalay, Simulating use scenarios in hospitals using multi-agent narratives, Journal of Building Performance Simulation 10(2) (2017) 1–17. doi: 10.1080/19401493.2017.1332687.

[8] I. R. Diamond, R. C. Grant, B. M. Feldman, P. B. Pencharz, S. C. Ling, A. M. Moore, P. W. Wales, Defining consensus: a systematic review recommends methodologic criteria for reporting of Delphi studies, J Clin Epidemiol 67(4) (2014) 401–409. doi: 10.1016/j.jclinepi.2013.12.002.

[9] T. Kristensen, M. Dyngeland, Design and development of a multi-agent e-learning system, International Journal of Agent Technologies and Systems 7(2) (2015) 19–74. doi: 10.4018/IJATS.2015040102.

[10] E. B. Artamonov, O. O. Zholdakov, Concept of creating a software environment for automated text manipulation, Scientific journal "Proceedings of the National Aviation University" 3 (44) (2010) 111–115.

[11] S. Jimaa, The impact of assessment on students learning, Procedia-Social and Behavioral Sciences 28 (2011) 718–721.

[12] A. M. Khan, H. Batool, S. Ashraf, A comparative study to evaluate assessment facilities at government special education schools, Pakistan Journal of Humanities and Social Sciences Research 02(01) (2020).

[13] A. Backurs, P. Indyk, Edit distance cannot be computed in strongly subquadratic time (unless SETH is false), SIAM J Comput 47(3) (2018) 1087–1097.

[14] E. Zorita, P. Cuscó, G. J. Filion, Starcode: Sequence clustering based on all-pairs search, Bioinformatics 31(12) (2015) 1913–1919.

[15] L. Yujian, L. Bo ,A normalized Levenshtein distance metric, IEEE Trans Pattern Anal Mach Intell. 29(6) (2007) 1091–1095.

[16] Y. Artamonov, I. Golovach, D. Krant, H. Rosinska, O. Nechyporuk, S. Stanko, Dynamic content generation methods based on user behavioral ranking, in: Proceedings of IEEE 4th International Conference on Advanced Trends in Information Theory (ATIT), IEEE, Kyiv, Ukraine, 2022, pp. 313–318. doi: 10.1109/ATIT58178.2022.10024196.