Baltic J. Modern Computing, Vol. 11 (2023), No. 1, 90-113
https://doi.org/10.22364/bjmc.2023.11.1.06

Development of Methods and Models for Generating an Adaptive Learning Plan Based on the User's Level of Knowledge

Svetlana JURENOKA, Janis GRUNDSPENKIS

Faculty of Information Technology, Riga Technical University, Riga, Latvia

svetlana.jurenoka@rtu.lv, janis.grundspenkis@rtu.lv

Abstract. During the e-learning process implementation, there is a knowledge forgetting parameter that determines the remaining level of knowledge after the e-learning process is completed. To assess learner's knowledge, e-learning management systems use formative assessment, which determines the level of knowledge in each examined competence during their acquisition. At the end of completing the study programme, the learner will have a different level of education in the acquired competences included in the study programme. The results of the study are not linked to the specific level of education, but are related to the methodology of the assessment of the learner's level of knowledge and the evaluation of the implementation of the learning plan. The article discusses the development of an adaptive e-learning plan generation algorithm and its integration within the e-learning management system of curriculum, which allows adapting curriculum to the level of knowledge and learning ability of each user by balancing the acquired knowledge in each unit of time, dynamically changing the topology of the learning plan.

Keywords: e-learning, leaning plan, adaptive learning.

1. Introduction

In 2019, due to the spread of COVID-19 in the world, both in Latvia and other countries, education system transitioned to remote format, which contributed to the digitization of teaching aids and the introduction of learning management systems in educational institutions. According to the Skola2030 (WEB, a) concept, the e-learning platform will be integrated as the main storage site for learning resources, which will be available to both educators and learners in preschool educational institutions and schools.

Today there are a large number of diverse systems that provide distance learning support, among which adaptive systems take a prominent place, as they implement a learning process that is centred on the learner and suitable for them.

Adaptive education systems were mentioned already in 1989 (Bruha, 1989). Since then the basic task of the systems has not changed. The task of an adaptive system is to manage and control a learning process adapted to each user or group of users, based on the learner's characteristics and learning speed of the learning process, which ensures user-oriented, supported, and improved learning. The goal of adaptive learning system is to create a flexible environment that supports the provision of the learning process, considering the abilities, needs, interests, limitations and other characteristics of each learner.

The study programme curriculum of a modern learning management system, regardless of its implementation form (intramural, extramural or remote), is adjusted to students with the average level of knowledge and the study plan adjustment is implemented mainly in a manual format, when the study process administrator, based on the learner's level of knowledge, chooses the appropriate rate of obtaining knowledge and the sequence of tasks within the framework of the study plan manually.

For knowledge monitoring in the digital learning systems, tests are used. In their most general meaning, tests are standardised checking of knowledge and understanding with the help of different tasks. Tests are used for introductory assessment, formative assessment and summative assessment. The assessment results indicate the extent to which the curriculum of the study programme has been mastered, which allows determining the speed of learning information and the learner's ability to participate in the learning process.

Regardless of the fact that modern computerised learning management systems can ensure complete educational process by using the information presentation method described above, the form of presenting information is not always acceptable for all groups of learners. By using the adaptive learning management system (ALMS), the users have to choose the study module themselves and adapt to the study course methods. Learning module - the minimum possible amount of educational learning material, which provides a review of a specific topic in the context of the curriculum. A computerized learning process management system can use one of the forms of learning content representation (text content, video content, audio content, and can also contain a set of various interactive activities) to display the module (Refaay et al., 2017). Modern studies confirm (Jurenoks, 2017) that, by using a learning management system with the previously defined study plan, the users cannot choose the learning style appropriate for their needs – in 39% of cases it becomes the reason for terminating studies and in 40% of cases for omitting unclear or hard materials and learning them with the help of other systems or courses. As a result, it can be identified that the lack of ALMS is related to adjustment of the study plan to the user's level of knowledge and ability to acquire the curriculum.

The implementation of the learning plan is influenced by factors related to the time of implementation of the learning process and the level of knowledge of the competences included in the course. The concept of competence denotes the knowledge acquired in the learning process, which gives the ability and opportunities to perform certain tasks, which is provided by a set of performance quality and experience (Maslo and Tilla, 2005).

The article offers adaptive learning plan generation methods, which allow reaching the minimum threshold of the required level of knowledge defined by the study course in each competence included in the course.

2. Related work

Recently, there has been an increase in the number of studies, which are related to the automation of the teaching process and the stage of assessment of results, assimilating the characteristic features of the process to the form of intramural implementation WEB (b), which allows adjusting the curriculum and the teaching strategy of the leaning content for users of different age and social group.

Analysing the influence of the learner model on the operation of the adaptive learning management system, the studies highlight tasks that require to model user characteristics for the selection of the subsequent learning module. Currently, there are no classifications of learner characteristics that ensure the creation of an adaptive learning plan, nor is there a methodology that allows for the assessment of these characteristics. As a result, it can be concluded that there is currently no single scenario for creating a learner model, which describes the selection of necessary properties and priorities for determining the user's level of knowledge. Based on (Volanskaja, 2002; Bul, 2003), for each study programme, depending on the tasks, different approaches are used to build the learner's model, which is influenced by the specifics of completing the study programme and the guidelines for providing knowledge.

Research related to the development of an adaptive learning management system was initiated at the end of the 20th century. Each learner has their own knowledge perception speed and pace of learning information, which depends on the learner's level of knowledge and external factors (Wang, 1983). Within the framework of adaptive learning, there are opportunities to create a learning scenario adapted to each user, considering the individual characteristics of the learner. The works (Santos et al., 2003, Sani and Aris, 2014; Vaca et al., 2012) rightly describe that the main task of adaptive learning is to increase the level of student achievement by bringing the interactive learning process as close as possible to the real lesson with the teaching staff in the classroom. Based on this assumption and the results of the study, it can be concluded that the main difference between adaptive learning and face-to-face classes is the possibility of automation of the learning process based on the learner's level of knowledge (Oxman and Wong, 2014).

In the 1990s, the pace of development related to information technology increased, which contributed to an increase in research related to the possibilities of content customization in learning management systems. With the development of web technologies and the increase in the power of home computer systems, at the beginning of the 21st century, there was an increasing number of studies related to the development of adaptive and intelligent web-based learning management systems based on the creation of a user model (Brusilovsky and Peylo, 2003; Hatzilygeroudis et al., 2005; Wen and Yang, 2005).

Today, research related to adaptive learning mainly investigates issues related to content creation and representation:

- research related to automated content search methods based on the user's level of knowledge, generation of information representation sequence based on the learning goals set by the user and the learner's knowledge levels (Santoso et al., 2014; Sumak et al., 2019);

- research related to the automated generation of learning resources based on the user's level of knowledge and psychological characteristics (Kardan and Noorbehbahani, 2009; Sarkar and Huber, 2021);
- research related to the development of the learner's model and the generation of the knowledge delivery sequence based on the learner's current level of knowledge and the pace of acquiring new knowledge (Bruha, 1989; Nguyen, 2014).

Modern adaptive learning management systems mainly use user-based models, such as the dpc.lv learning platform, which ensures the adaptation of course curriculum to each user's level of knowledge. Each course, which is integrated into the system, consists of a number of components (modules) required for each lesson, which are divided into short 5–10-minute video lectures. The modules under consideration are supplemented with metrics that determine the relevance of the module to the learner's level of knowledge. During the learning process, the choice of topic and the scope of the topic to be taught are adjusted to the learner's level of knowledge, which allows choosing an appropriate learning pace, respecting the number of hours allocated to the learning process.

Currently, it is believed that real-time continuous monitoring of the learner's knowledge level and adjustment of the learning scenario based on the current knowledge level is an integral part of every adaptive system (Zhang et al., 2020). The works of some authors (Anil and Abdul Moiz, 2019; Klett and Pharow 2006) are devoted to research related to the individualization of the learning scenario using the previously developed study course. For example, the online theoretical learning platform macam.lv uses a model, which ensures the acquisition of the course for obtaining Category B and C driving license by differentiating the content according to user's level of knowledge, thus raising the level of the vehicle driving category. The learning platform adjusts the sequence of topics during the implementation of the study programme to the level of knowledge of each learner, based on the data obtained from the test performance results.

In Latvia, a solution to the implementation of adaptive learning process for Moodle learning management system was proposed in 2017 (Jurenoks, 2017). The article defines the criteria necessary for the validation of the learner's adaptive knowledge level before starting the study course. Based on the conducted research, the author uses Dijkstra's algorithm for generating assessment test questions, which allows testing wide areas of the learner's knowledge, using the possible limitation of the number of questions asked in test systems.

During the implementation of the study programme, contemporary learning management systems use a model that allows for the selection of the subsequent learning module from an existing curriculum based on previous actions during the learning process.

As a result, it can be concluded that there are currently no unified solutions that ensure the generation of a learning plan for each user, which does not allow adapting the study program to each learner's level of knowledge.

3. Development of adaptive learning plan reconfiguration method

The learning plan reconfiguration method described in the article includes several processes that determine the need to activate the reconfiguration module. The learning management system ensures the validation of the learning plan through the feedback of information processing, which includes the validation of the available resources, learner model and knowledge level.

Using a real-time controlled learning plan generation method, the article proposes a set of algorithms that ensure the verification of each learning plan element for the generation of a new learning plan for the acquisition of defined competences. The article also describes the algorithms that ensure the development of a module acquisition sequence plan using the learning module acquisition time parameter.

In situations when the acquisition of real-time feedback of the learning process is limited, it is proposed to use the genetic algorithm for the inclusion of the learning module in the curriculum.

3.1. Substantiating the need for learning plan change

Other studies (Novikoff et al., 2011; Haritonov and Krushel, 2012) demonstrate that during the learning process the level of each competence is reduced if issues related to the competence acquisition are not addressed in other learning modules. Figure 1 illustrates the level of knowledge acquired by the learners as a result of learning the course using one of the three defined curricula.

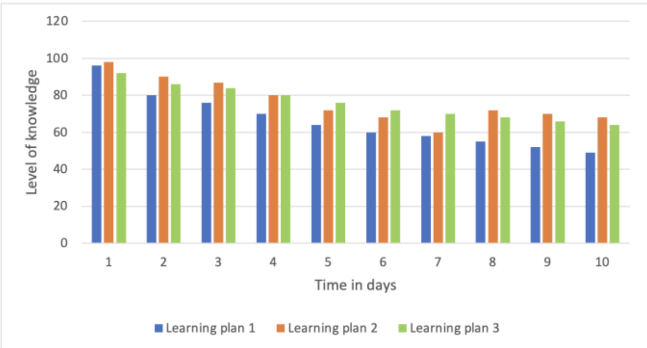

Fig 1. The level of knowledge of the remaining competences during the acquisition of the study course

As shown by Fig. 1, the level of learners' knowledge decreases during the learning process using the previously defined learning plan (learning plan 1). By providing manual adaptation of the learning plan to the learner's level of knowledge (learning plans 2 and 3), it can be seen that the level of knowledge acquired by learners depends on the selected learning plan. According to Fig. 1, the changes in learning plans 2 and 3 are minor, and they depend on the placement of the learning module during the curriculum acquisition.

Other studies (Srinivasa et al., 2022; Mejanova et al., 2022) have defined that the level of knowledge during the learning process is a dynamic value that can change depending on the chosen learning plan scenario. A learning plan can be considered viable as long as the following conditions are met:

- the time required to acquire the remaining learning modules is greater than the sum of the time of the unlearned modules;
- the learner's level of knowledge is higher than the limit value defined in the study programme;
- the learner does not continue progressing within the syllabus of the course, and the time for the acquisition of learning modules is greater than the defined limit value.

3.2. Setting adaptive learning plan generation tasks

The task of generating the adaptive learning plan is the construction of a trajectory for acquisition of the learning content, which will allow increasing the level of each competence included in the learning process during the final tests within the study programme. The learner, while acquiring the study programme, improves their competences, which are defined in the study course and are linked to study modules, where the level of acquired knowledge is determined using test modules.

Acquiring a learning module, the learner can improve one or more competences, as a result of which it can be assumed that each learning module provides knowledge for improving at least one competence. Input values for module initialization (required competences) are optional and may not be specified for initial learning modules.

The learning plan, which includes the acquisition of learning modules and improving the level of competences, can be represented as a bipartite graph, where the vertices are divided into two categories – modules and competences, and each arc of the graph connects a vertex from one part of the graph with a vertex from another part; namely, there are no links between vertices of the same part of the graph.

The implementation time of the learning process is limited by using the course acquisition time provided for each study course *t*. By acquiring the learning module, the learner acquires the competence K[c], which is measured as a percentage on a scale from 0% to 100%. It is assumed that when acquiring the study course, at the end of the course, the users have achieved the overall level of competence $\sum_{i=1}^{n} K[c_i]$, which combines the results of each competence level. The task of the learning plan algorithm is to generate such a path in the graph P so that the level of competence achieved at the end of the course is the highest in the specified time period *t* (Eq. (1)).

$$\sum_{i=1}^{n} K[c_i]\,(P,t) \rightarrow max, \qquad (1)$$

where $K[c_i]$ – competence included in the study programme, P – acquisition sequence of learning modules (path in the graph), *t* – the time required to determine the level of competence.

There is limitation during the generation of the study plan, which is related to the acquisition time of the learning modules.

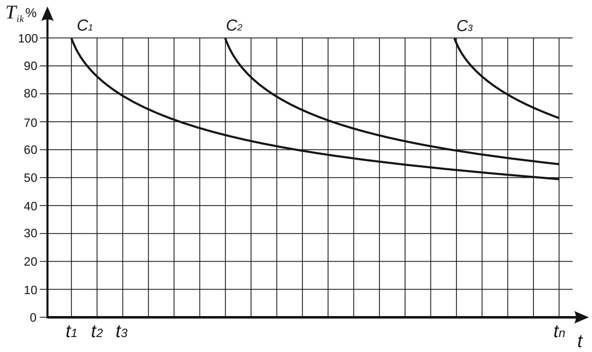

Fig 2. Extent of information forgetting for learning competences C1->C2->C3

For example, the learner fully acquires competences sequentially C1, C2, C3, where the time required for acquiring competences is C1[t]=10; C2[t]=40; C3[t]=20. Taking into account the rate of forgetting information, there is a situation that at the end of the course, the level of knowledge within each specific competence will be different (Fig. 2).

By changing the acquisition sequence of learning competences or by integrating modules related to a certain competence in other competence acquisition processes, it is possible to balance the remaining amount of knowledge among all the competences at the end of the study course. As a result, it can be argued:

1. If only one competence is defined as the learning outcome for the learning module, then during the curriculum acquisition it is necessary to learn the modules that ensure the achievement of the relevant competence. In cases when the remaining level of the competence is less than the minimum requirement defined in the study programme, it is necessary to choose modules that have not been covered before to repeatedly acquire the competence.
2. If maximum time *t* is defined for acquiring the curriculum, it should be observed that the total time for learning the modules to achieve the competences cannot be greater than the total time of the course $K(t) > \sum_{i=1}^{n} M_n(t)$.

3.3. Learning plan generation algorithms

The learning plan generation algorithm consists of two interconnected parts: inclusion of learning modules for acquisition of competences and determination of the acquisition sequence of competences. Taking into account the rate of forgetting the learned information (Lange, 1983), it is determined that if the interval between classes is increased without changing the number of classes, the amount of knowledge at the end of the course decreases and time is needed to repeat the learning material. As a result, it is believed that when developing a learning plan, the learner will operate within the framework of one competence until they fulfil the requirements of the defined learning plan.

Learning module sorting algorithm

Let us suppose that there is learning plan configuration graph G consisting of a set of vertices V= {1,2…n} representing learning modules and sets of edges, as well as the time required to acquire a learning module. In the process of content development or improvement, new learning modules may appear that provide information that is necessary for acquiring certain competences. The learning module is described using the entry requirements - necessary knowledges for completing the module as well as the learning outcomes to be achieved. Each learning module provides only one outcome, that is linked with module topicality. Module can have one or more requirements (not limited to specific competence) that indicate possible position of this module in learning curricula. For example, let us describe the available modules that are included to acquire *K1* competence (see Table 1).

Table 1. Acquisition Pre-requisites of Learning Modules

Necessary knowledges for completing the module	Title of a learning module	Learning outcomes (knowledges)	Competence	Time required for module acquisition (units of time)
-	M0	R0	K1	5
R0	M1	R1	K1	14
R0	M2	R2	K1	12
R1	M3	R3	K1	8
R2	M4	R3	K1	5
R1	M5	R9	K1	22
R2	M6	R6	K1	8
R3	M7	R9	K1	7
R3	M8	R9	K1	5
R6	M9	R9	K1	7
R9	M10	R10	K1	20

Each module has a minimum learning time value. As a result, the interaction of the K1 competence module can be represented using a directed graph, where the element of the graph is the learning module and the weight on the arc is the time required to acquire the module (see Fig. 3).

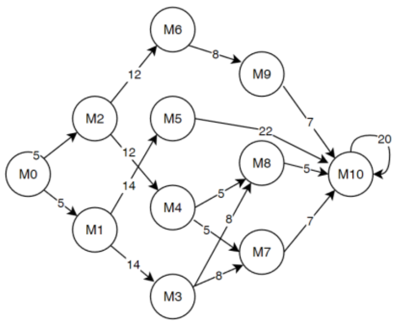

Fig. 3. Interaction of learning modules for competence acquisition

When developing a learning plan, it is necessary to identify what is the minimum and maximum time required to acquire each competence using the data set of the learning module. To determine the shortest path between two vertices in the graph, it is possible to use one of two algorithms: Bellman-Ford or Dijkstra's algorithm. In the considered graph, the weights of the edges have a positive value, as a result of which the authors propose to use Dijkstra's algorithm for determining the shortest path.

Let us implement the procedure ***get_k_time()***, which determines the minimum time required for acquiring learning competences.

```
def get_k_time(self, start_vertex):
  D = {v:float('') for v in range(self.v)}
  D[start_vertex] = 0
  pq = PriorityQueue()
  pq.put((0, start_vertex))
  while not pq.empty():
   (dist, current_vertex) = pq.get()
    self.visited.append(current_vertex)
    for neighbor in range(self.v):
      if self.edges[current_vertex][neighbor] != -1:
        distance = self.edges[current_vertex][neighbor]
        if neighbor not in self.visited:
          old_cost = D[neighbor]
          new_cost = D[current_vertex] + distance
          if new_cost < old_cost:
            pq.put((new_cost, neighbor))
            D[neighbor] = new_cost
return D
```

The algorithm uses the validation of the learning module relevance, which determines the possibilities of acquiring the learning module in a certain period of time, based on the module selection process and the methodology for determining the impact factors of technical support (Novikoff et al., 2011; Rahimi, 2020; Mohamad et al., 2021).

As it has been mentioned above, when developing a learning plan, it is necessary to balance the remaining level of knowledge in each competence acquired in the study course. As a result, by determining the time needed to acquire learning competences we define a set of competences, which allows arranging the initial acquisition sequence of the unrelated competences, taking into account the time that should be spent to acquire the learning competences. It has been experimentally proven (Novikoff et al., 2011) that in case of alternative sequence choice during the development of a learning plan, it is recommended to use competences where the time required for the acquisition of the competences tends to the maximum, but does not exceed the maximum time allocated for the acquisition of the competences.

Let us determine all possible paths that ensure the achievement of learning outcomes. In total, the graph (Fig. 3) has 6 paths with path length being 27, 29, 32, 34 and 41:

- Path 1: 0→2→4→8→10 , length 27;
- Path 2: 0→2→4→7→10, length 29;

- Path 3: 0→2→6→9→10 , length 32;
- Path 4: 0→1→3→7→10 , length 34;
- Path 5: 0→1→3→8→10 , length 32;
- Path 6: 0→1→5→10, length 41.

Based on the information presented in Fig. 3, it can be seen that the choice of path does not affect the achievement of the goal. As it has been mentioned above, the time required to complete the learning module without interrupting the learning process does not affect the scope of the learning outcome to be achieved but varies depending on the presentation type of information or the pedagogical method used in the learning process.

When a learning path (for example, path 4) is selected from the path set *path* $=\{1,2\ldots n\}$, the learning modules used in constructing the path are identified as executed (for example, M0, M1, M5, and M10). As a result, when the level of knowledge in a particular competence is lower than the minimum limit $K[c_i(min)]$, the system includes modules related to the specific competence in a learning plan. When generating a new learning plan, only previously unseen modules are selected from the data set.

For example, if the learner has acquired the learning competence $K[c_i]$ using the sequence 0→1→5→10, then when repeatedly acquiring the competence, the system will use a new learning plan, generating five possible paths (see Fig. 4).

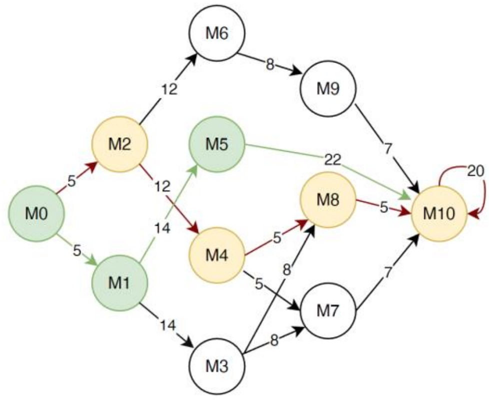

Fig. 4. Selection of learning modules for the repeated acquisition of competence $M[c_i]$

As it has been mentioned above, when developing a new learning plan, the previously used modules are no longer included in the new learning plan. Module M10 is the final module for competence acquisition, which also includes a test to assess a learner's knowledge. Therefore, this module will be incorporated in a new learning plan.

When repeatedly acquiring the competence, based on the available time limitations of the study programme, the task of the system is to re-include in the programme learning topics related to the learning competence, where the remaining level of knowledge is lower than the course limit $K[c_i(min)]$. The article proposes to choose the shortest path for repeated acquisition of competences, which is justified by the time limit after acquiring basic competences. Figure 5 shows the learning module sorting algorithm for competence acquisition.

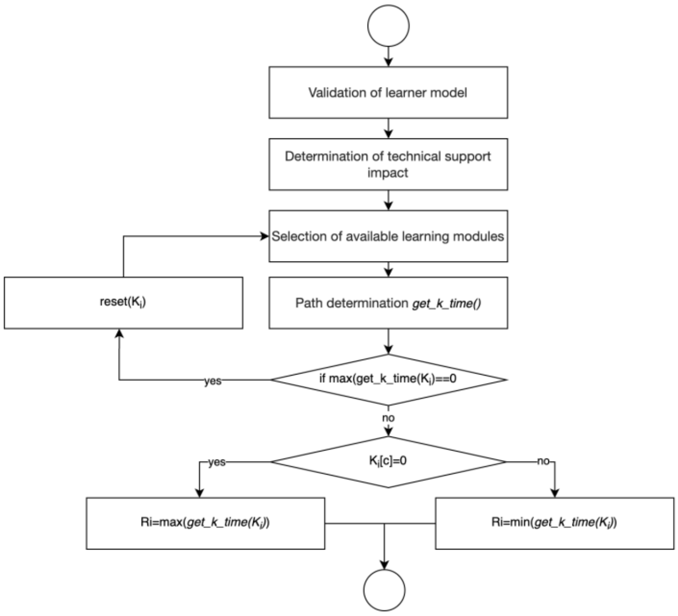

Fig. 5. Learning module sorting algorithm for competence acquisition

Learning competence sorting algorithm

One of the tasks of the learning competence sorting algorithm is to balance the levels of knowledge obtained in the learning process among all the covered competences. Let us assume that each competence is acquired when all the modules included in the learning competence are completed. Besides, it should be noted that the modules are selected using the learning module sorting algorithm described above (Fig. 5). The result of module selection will be different for each competence, but it cannot be less than 1 and greater than the number of all modules needed to acquire the competence. Let us define an array *MP[k,m]*, where *k* is the available number of competences and *m* is the maximum number of modules of one competence that are used in the study course. The study programme may include competences where the process of acquiring knowledge depends on the previously acquired competence in the study plan, as well as competences that are viewed as an individual object. Competences can be divided into three basic categories: basic competences, optional competences and additional competences. Thus, the curriculum *MP* consists of three sets (Eq. (2)).

$$MP_i = \{Kmain_1, Kmain_2 \dots, Kmain_n\}\{Kadd_1, Kadd_2, \dots, Kadd_n\}\{Kopt_1, Kopt_2, \dots, Kopt_n\} = \overline{1,n}\ , \qquad (2)$$

where *Kmain* – basic competence included in the study programme;
Kadd – optional competence included in the study programme.
Kopt – additional competence included in the study programme.

The task for inclusion of learning competences in the learning plan can be divided into two parts:

- to select the number of competences to be included in the learning plan from the competences available in the system, which is regulated by the requirements of the study programme or learning administrator. As a result, a data set is created, which contains information about the competences to be included in the study programme and the time required for their acquisition, which is determined using the learning module sorting algorithm (Fig. 5);
- to select additional competences that complement the scope of knowledge to the competences included in the study programme.

Let us assume that all the competences that are included in the study programme can be divided into categories by assigning them an evaluation factor. Competences are defined in the description of the study programme, and a place for acquiring the competences within the course is suggested. As a result, it can be considered that the set of learning competences can be represented as a directed graph by dividing the competences according to their acquisition sequence. For example, within the study programme "Python Programming Language", the study course "Programming Languages and Compilers" (15 hours) includes 10 competences, which are divided into three basic blocks (Fig. 6).

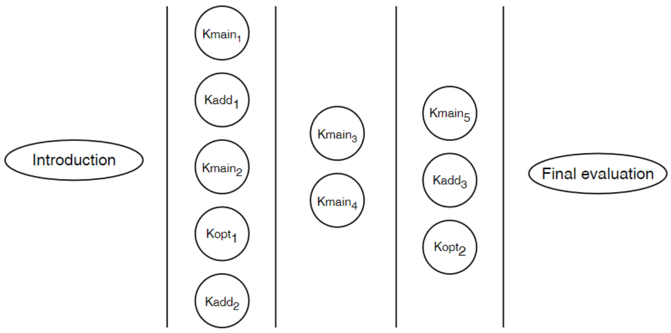

Fig. 6. Distribution of competences based on the sequence defined in the study programme

Basic competence selection algorithm

During curriculum development, the system determines the maximum time required for acquiring each competence using the module sorting algorithm (Fig. 5). In the beginning, a set of basic competences is distinguished, which is mandatory for acquiring the study programme $MS = \{Kmain_1, Kmain_2 \dots, Kmain_n\}$.

When selecting a set of basic competences, the following condition must be met: $K(t) > \sum_i^1 K_n main_i\,(t)$, which means that such a way of acquiring competences should be chosen so that all basic competences $K_n main$ can be acquired in the time period $K(t)$ allocated for the study course. In case if $\sum_i^1 K_n main_i(t) > K(t)$, the system replaces the module with the largest time value with the next possible selected learning path for competence acquisition. The basic competence selection algorithm is shown in Fig. 7.

The task of the basic competence selection algorithm is to select the number of competences by determining the maximum possible learning time for each competence, which in total for all selected competences does not exceed the time allocated for acquiring the study programme.

As it has been mentioned above, the basic task of the learning management system is to ensure the full acquisition of basic competences. During the execution of the algorithm, the system performs the validation of the remaining level of knowledge of the acquired competences using the information forgetting rate (Lange, 1983).

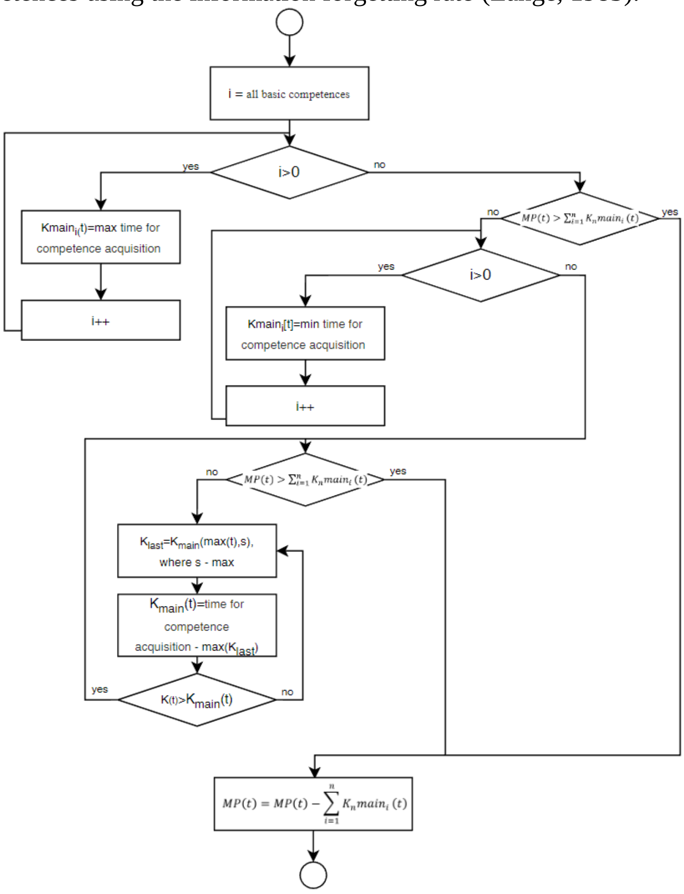

Fig. 7. Basic competence selection algorithm

During the implementation of the learning process, the basic learning competences can be divided into two parts (Letina, 2020):

- individual competence – competence that has no relation with other competences included in the study course;
- group competence – competence that is linked to other competences included in the study course.

Linking learning competences with other competences included in the study course indicates repetition of information, repeating knowledge that has been acquired previously. Let us suppose that the learning process is implemented using the competence K[c], then the method of determining the amount of information forgetting (Lange, 1983) is not applied to all the lower-level related competences.

As a result, during the generation of the basic competences, the system can identify the time required to acquire all the competences, taking into account the balancing of the remaining level of knowledge among all the competences included in the study programme.

Optional competence selection algorithm

The operation of the optional competence selection algorithm is related to two influencing factors on the user's side:

- time required to acquire basic competences;
- the remaining level of knowledge of basic competences.

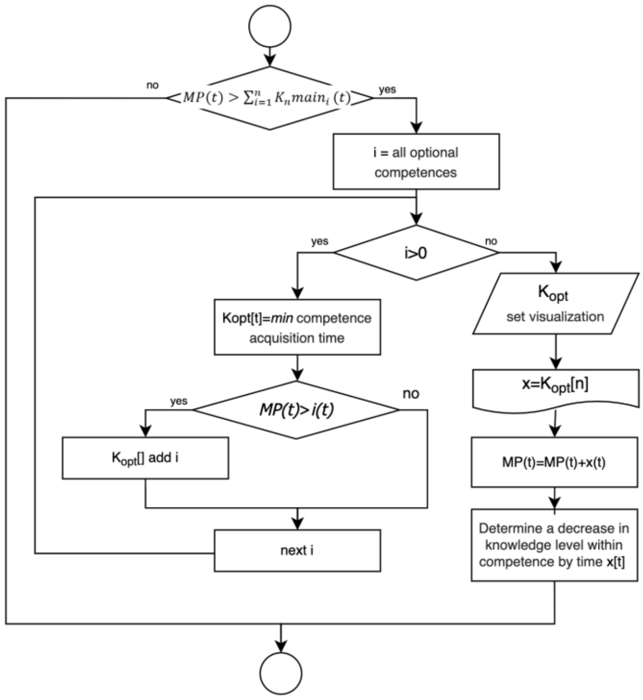

Fig. 8. Optional competence selection algorithm

The set of optional competences is determined by the remaining time of implementation of the learning process and the maximum number of optional competences that can be acquired in the specified period. The optional competence selection algorithm is shown in Fig. 8.

In the described algorithm (Fig. 8), during the acquisition of optional competences, the information forgetting determination algorithm is applied to all basic competences. As a result, the process of selecting optional competences is implemented as long as it is possible to fulfil the condition related to the time needed to acquire information and the level of knowledge of the existing basic competences.

In cases when the knowledge level of basic competences is lower than the limit level defined in the system, the basic competence selection algorithm is initialized for information repetition.

Additional competence selection algorithm

The time of acquiring additional competences is the time that does not affect the total learning time of the study course $K[c_i(t)] \notin K(t), kur\ K[c_i] \in Kopt$. As a result, the number of additional competences does not affect the structure of the study programme and can be integrated at any point of the study course within the study programme. To select the number of additional competences, two approaches are proposed:

- a unit linked to other competences included in the course syllabus, where the link is defined during the development of the study material;
- a unit linked to other competences included in the course syllabus, where the link is determined using a comparative analysis of the descriptive text.

In cases where a link is found with a competence in the curriculum, the level of knowledge of the competences used during the acquisition of the selected competences is renewed during the course.

Using metrics in module selection

To make the additional competence selection algorithm effective, it is necessary to ensure a closer link with the subject of the curriculum when selecting optional or additional learning modules in the system, as a result of which the completion of the modules increases the remaining level of knowledge both in the competence under consideration and in the related competences. Let us suppose that a set of additional learning competences is assigned metric *Kl[s]*, a value that is located in the indexed set and can be compared, which represents the number of links of the additional competences with other basic competences included in the curriculum (Eq. (3)).

$$Kl(s) = MP[\mathrm{k}_1, \mathrm{k}_2, \dots, \mathrm{k}_n], \ \ k_i \in \ Kmain, \tag{3}$$

where *MP* – a set of components that the learner acquires during the implementation of the curriculum.

Let us consider that the competence *Kl[s]* is useful for the competence *Kmain*, if the condition is fulfilled:

$$\begin{pmatrix} Kl(s_1) \\ Kl(s_2) \\ \dots \\ Kl(s_n) \end{pmatrix}_{con} \xrightarrow[Sort]{} \begin{pmatrix} Kl(s_{i_1}) \\ Kl(s_{i_2}) \\ \dots \\ Kl(s_{i_n}) \end{pmatrix}_{con}, \tag{4}$$

where $Kl\left(s_{i_j}\right) > Kl\left(s_{i_{j+1}}\right), j = 1,2, \dots, n-1.$

This condition plays an important role. All metrics with high scores are selected from the metric list *Kl[s]*. Thus, the possibility of choosing a learning competence that is not related to acquiring the basic part of the study course will be reduced.

Using genetic algorithm for curriculum generation

A genetic algorithm is a problem-solving strategy to optimize and search for correct or approximate solutions to a problem. It is a search algorithm based on Darwin's theory of evolution, that in nature there is a chance for survival of individuals that are better suited to a specific environment (change of generations, competition among individuals) (Nikitenko, 2006; Zuters, 2022).

Genetic algorithms use operators based on randomness and the individuals of each subsequent generation will be different from the initial population (Takahasi, 2004).

In the article, combinations of modules are called chromosomes, or individuals. Each iteration of a genetic algorithm is called a generation, and better chromosomes are involved in generating the next generation. When generating the chromosomes of a new generation, the chromosomes of the previous generation are used based on the crossover and mutation method (Nikitenko, 2006).

Mutation is a method that changes one or more genes in a chromosome, and the crossover method involves several chromosomes and their fragments that form a new individual (Zuters, 2022).

Genetic algorithms can be described by three algorithm steps (Nikitenko, 2006; Takahasi, 2004):

- to generate and evaluate the starting chromosome (solution) set P;
- to generate new chromosomes for the set P_i using the chromosomes of the previous set P;
- to repeat the generation of new chromosomes if the final condition is not fulfilled.

As it has already been mentioned above, the mutual interaction of learning modules is represented by a graph in the article. The task of learning plan generation is to arrange the learning modules on a linear plane, respecting the requirements related to the study programme. Let us assume that the attitude of the learning module in the graph is determined by the weight defined by the edge. The task is to determine the order of the arrangement of the vertices for their placement in a linear learning plan, ensuring a minimum sum of the weights of the edges of the graph between all the vertices. Let us assume that there is graph G of learning competence modules consisting of 5 vertices and 4 edges. The time needed for each learning module completion is the same, resulting in the weight of edges equal to 1.

Genetic algorithm has two approaches:

- genetic crossover method;
- mutation method.

The authors propose to employ the genetic algorithm, which uses the crossover method, when there are no reference plan models for acquiring a study course.

Using a random or predefined arrangement of vertices on a linear plane, it is possible to determine time that is needed to acquire a learning competence (Fig. 9).

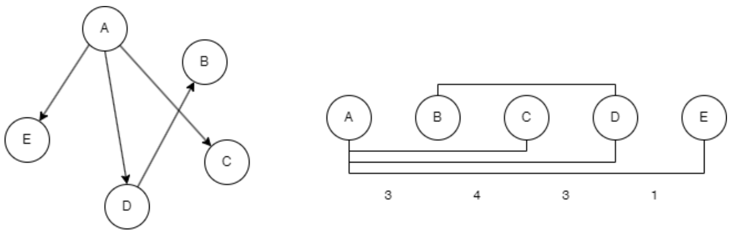

Fig. 9. A random arrangement of the vertices of the graph on a linear plane

Using the base step arrangement of the module, the time required to acquire the learning competences is equal to 11 units. The task is to determine the minimum time required for the acquisition of competences in order to include as many subject-related additional competences as possible in the study programme.

When working with a genetic algorithm, it is necessary to define an initial arrangement of possible vertices on a linear plane. In the genetic algorithm, this combination is called chromosomes. Let us define three randomly generated initial chromosomes P1 = 1,2,3,4,5; P2=2,1,3,4,5; P3=5,2,3,4,1, where the length of the chosen path of each chromosome is equal to P1=11 units, P2=P3=9 units.

It can be considered that in the current state, the first best chromosome is selected, to which the mutation algorithm is applied. The mutation algorithm performs the inversion process using the first element. Let us create a new path Px=2,5,4,3,1 by positioning it in place of the inferior chromosome. When fulfilling the task, a new set of chromosomes P1=2,5,4,3,1; P2=2,1,3,4,5; P3=5,2,3,4,1 is created. By defining the maximum number of generations, the system continues running the mutation algorithm as long as all the chromosomes are not within the defined boundaries and the curriculum does not include all the basic competences.

When it is necessary to generate a curriculum using several input parameters (number of modules of one topic, acquisition time of learning modules), then the genetic algorithm of the crossover method is used. Let us define a function (Eq. (5)) that evaluates the suitability of a module for its inclusion in the study programme, using the ratio of the total time for module acquisition and the number of modules for competence acquisition.

$$f(p) = \frac{\sum_1^n M_n(t)}{\sum_1^i M_i}, \tag{5}$$

where M – a learning module included in the system;
M(t) – implementation time of the learning module included in the system.

The task is to determine the extremum value of the function (depending on the task, one of the limit values of the function, *min* or *max*, is searched for). The genetic algorithm performs two tasks during operation:

- determines the limit values of the number of modules under consideration, which can be acquired in a certain time interval;
- determines the time limit required for acquiring certain competences.

Possible combinations of modules for acquiring the learning course are used to define the initial chromosomes. Modules are selected from all the competences included in the study programme, following only the order of their acquisition rather than adherence to the category of competences. As a result, if there is a certain amount of time allocated for acquiring one competence during the study course, then the task of the genetic algorithm is to find the optimal set of modules that can be implemented in a time interval that approaches the specified time limit. The number of chromosomes is determined taking into account the maximum number of paths from the initial graph position to the final vertex and the learning module time required to acquire the learning material. Chromosomes consist of mutually independent genes: time to acquire modules and number of modules. The crossover algorithm uses the best chromosome-shuffling technique to create a new population at each generation step. During the crossing process of the genetic algorithm, it has been found that in cases where the values are close to the optimal solution, it may happen that other chromosomes that do not contain the genes of the optimal solution are involved in the mixing process. As a result, continuing the mutation process, it may happen that a high concentration of population individuals will form at a point that does not correspond to the solution of the problem (Fig. 10).

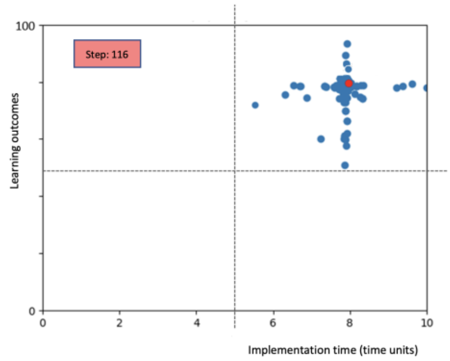

Fig. 10. Concentration of individuals in a population

During the operation of the algorithm, a situation may arise that the system will determine the smallest number of modules necessary for the acquisition of the study course, replacing the basic competence modules with modules from optional or additional competences. In order to reduce the inclusion of an unimportant module in the population mutation process, the authors propose to introduce strong individuals that will determine the guidelines for learning plan generation, including modules from the basic competences in the mutation process. During the operation of a genetic algorithm, individuals of basic competences are included in the stage of each population formation, which will limit the algorithm deviation from the learning plan of the basic course.

To diversify the learning plan, it is necessary to ensure parallel acquisition of learning competences. It means that learning modules to be included in the learning plan are selected from all the competences based on the module entry requirements. To ensure the diversification of the study course for crossing two parent specimens, let us define the limit of the number of generations, which will determine the interval when it is necessary to introduce the strong individual to form a new population. Strong individual is selected using the shuffle method when the next learning module is selected from the set of basic competences.

4. Experimental validation of an adaptive learning plan generation model and methods

The chapter describes one real experiment for the approbation of an adaptive learning plan generation algorithm in a study course with determined time necessary to achieve a certain level of knowledge. This experiment has been executed using the adaptive learning control module developed during the research, which ensures management of the Moodle learning platform.

Using the learning platform das.lv, which provides online training classes for adults, an experiment was conducted with the aim of increasing the knowledge level of learners by completing professional development distance learning programmes related to information technology. During the experiment, the study course "Financial Data Analysis and Drawing up Reports in the Excel Environment" was used. It ensures the acquisition of 160-hour study programme in the extramural form of study. The extramural form of study requires at least 30% to be implemented in person, and the rest of the learning process takes place in the extramural form using the available learning resources. In accordance with the license of the study programme, theoretical classes in the amount of 44 hours and practical classes in the amount of 60 hours are implemented in the extramural form.

The aim of the experiment

1. To check the use of the learning plan reconfiguration method for the selection of competences to achieve results in all competences in the shortest possible period of time.
2. To evaluate the influence of the method of balancing the level of competence knowledge on the implementation time of the learning process.
3. To evaluate the influence of the learning plan reconfiguration method on the total amount of the acquired knowledge in all competences.

The object of the experiment

The object under investigation is a study course developed in the Moodle learning management system, which ensures the delivery of knowledge related to information technology in an automated mode (www.das.lv). The study course consists of 16 basic learning competences. The course includes theoretical modules in the amount of 80 hours and practical modules in the amount of 150 hours. The time allocated to the learning process is determined and cannot exceed that specified in the study programme (44 academic hours for theoretical lectures and 60 academic hours for practical classes). The in-person and online learning process takes place in parallel, which requires the renewal of the learner's knowledge level before and after each learning activity. Intramural and extramural classes cannot overlap. The study programme is considered acquired if the learner's knowledge level for all basic competences included in the programme exceeds the 55% limit.

Determination of the reference value of the scope of the learning plan execution

In the experiment, it is considered that the learning plan is completed when the learner fulfils all the activities envisaged in the learning process, reaching the minimum requirements in all basic competences. The reference value is considered the learning plan implementation time t_s, which is needed by the learner to acquire the competences included in the study programme at the appropriate level, using the sequence of the learning modules defined in the study programme.

During the experiment, the learner cannot repeatedly acquire the learning material if it has already been acquired. During the experiment, the minimum knowledge level threshold c_{min}=55(%) was determined for the target audience group. Let us define the knowledge level threshold c_{min}=30(%), which determines the need to adjust the learning plan to balance the knowledge level among all competences.

Let us introduce the efficiency coefficient of learning competence acquisition δ_i, which indicates the deviation of the level of knowledge obtained in the considered competence from the average level of knowledge of the competences acquired in the entire study course.

During the experiment, the amount of teaching of the basic competences was not changed. The adaptive learning process was implemented when the learner had acquired the basic competences at the initial level ranging between 30% and 40% during face-to-face lectures, as result three groups of 12 students were selected for experiment. The results of the experiment are shown in Table 2.

Table 2. Results of the Second Experiment

	Static learning plan			Learning plan generation algorithm to increase the knowledge level			Balanced knowledge level		
Competence	Acquired knowledge level %	Number of learning modules	δ_i	Acquired knowledge level %	Number of learning modules	δ_i	Acquired knowledge level %	Number of learning modules	δ_i
K1	69%	5	8%	66%	7	4%	60%	6	0%
K2	61%	7	0%	61%	5	-1%	62%	3	2%
K3	63%	8	2%	63%	7	1%	59%	3	-1%
K4	64%	9	3%	64%	4	2%	62%	5	2%
K5	61%	7	0%	55%	7	-7%	63%	3	3%
K6	66%	8	5%	62%	6	0%	60%	6	0%
K7	62%	9	1%	60%	6	-2%	58%	5	-2%
K8	63%	4	2%	60%	5	-2%	60%	6	0%
K9	57%	5	-4%	57%	4	-5%	61%	6	1%
K10	65%	6	4%	67%	4	5%	60%	7	0%
K11	55%	8	-6%	69%	3	7%	61%	3	1%
K12	57%	7	-4%	56%	6	-6%	61%	4	1%
K13	55%	8	-6%	64%	5	2%	58%	4	-2%
K14	61%	9	0%	57%	5	-5%	63%	3	3%
K15	55%	4	-6%	64%	6	2%	59%	3	-1%
K16	58%	1	-3%	61%	3	-1%	59%	5	-1%
Average level of acquired knowledge within the programme	61%	55%	69%	62%	55%	69%	60%	58%	63%
Programme implementation time in minutes	6660			5715			7020		
	61%			62%			60%		

Figure 11 illustrates the deviation of the level of knowledge obtained for each competence in the learning process from the average level of knowledge among all competences acquired in the study course.

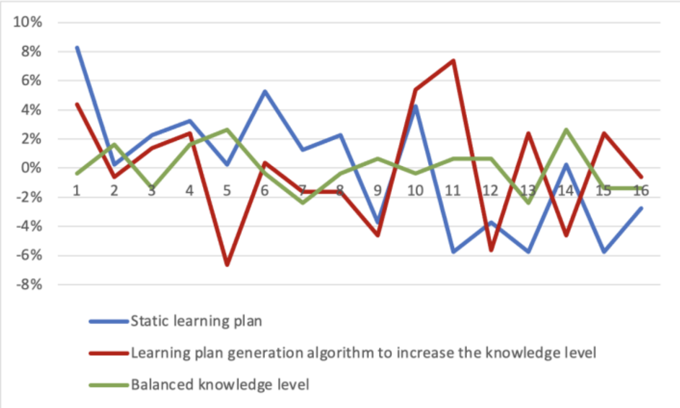

Fig. 11. Deviation of the acquired knowledge level from the average knowledge level within the course

Summary of experiment results

1. Using the static learning plan generation method, it is possible to acquire the learning content, fitting in the time allocated for the learning process and meeting the defined requirements of the study course.
2. Using the learning plan reconfiguration method, it is possible to reduce the time allocated for the learning process by excluding learning modules that do not provide a significant increase in the level of knowledge of the already acquired competences.
3. Using the method of balancing the level of knowledge acquired by the learner, it is possible to reduce the value of the efficiency coefficient of learning competence acquisition δ_i by balancing the differences in the level of knowledge among all competences.
4. Using the knowledge level balancing method, the learning time of programme acquisition is increased, which during the experiment reached the maximum possible time for acquiring the learning material.

5. Conclusion

The classification of competences included in the learning process has been described in the article. The authors of the research have developed the methodology for the inclusion of competences in the learning plan, which provides the inclusion of the initial mandatory competences, ensuring the acquisition of the mandatory part of the study programme to a full extent. By ensuring the inclusion of basic competences, the learning plan generation algorithm selects the maximum possible time for acquiring learning competences, which does not exceed the time allocated for acquiring competences. In cases when the learner’s knowledge level of competences falls below the threshold defined in the programme, it is necessary to choose the sequence of learning modules with the shortest learning time for repeatedly acquiring the competences in order to increase the number of optional competences.

The algorithm for the inclusion of optional competences has been proposed, using two scenarios: a competence selected by the learner that is not related to other competences included in the study course and a competence related to the study programme that is selected based on the remaining knowledge of the basic competences.

In order to avoid useless changes in the learning plan, the authors of the research have developed the learning plan reconfiguration process activation method, which is based on the assessment of the time allocated for the study process and the determination of the remaining level of knowledge. Based on the research results, the primary conditions for changing the learning plan have been set:

- the time required for acquiring the remaining learning modules $M[t]-t_{main}-t_{opt}$ is less than the sum of the time of the unlearned modules;
- the learner's level of knowledge is lower than the threshold c_{min};
- the learner does not continue progressing within the syllabus of the course, and the waiting time t_{max} is greater than the defined limit value.

For the acquisition of learning competences, the authors of the research have developed algorithms, which determine the selection of available learning modules for learning plan generation. The article describes algorithms for the selection and sorting of modules and competences, which determine the choice of an appropriate learning path using Dijkstra's algorithm.

To make the additional competence selection algorithm effective, it has been proposed to introduce additional learning module metrics that ensure a closer link with the subject of the study programme, as a result of which the acquisition of the modules increases the remaining level of knowledge both in the considered competence and in the related competences.

In cases, when working with an adaptive learning management system, a situation arises when receiving feedback in real time is not possible, it has been proposed to use a genetic algorithm for generating a learning plan reference model, determining the optimal learning module acquisition strategy.

References

Anil, G. R., Abdul Moiz, S. (2019). Personalized Dynamic Learning Plan Generator for Smart Learning Environments, *International Journal of Recent Technology and Engineering (IJRTE)*, doi: 10.35940/ijrte.B3806.078219.

Bul, E. (2003). Comparative analysis of student models, *Telematics -2003*, pp 362–364.

Bruha, I. (1989). Defining adaptive and learning systems, *Cybernetics and Systems: An International Journal*, Volume 20. pp. 77-88.

Brusilovsky P., Peylo C. (2003). Adaptive and Intelligent Web-based Educational Systems, *International Journal of Artificial Intelligence in Education, 13*

Haritonov, I.M., Krushel, E.G. (2012). Forecasting the level of development of competencies by university graduates based on the ratings of applicants, *Modern problems of science and education 2012*.

Hatzilygeroudis I., Giannoulis C., Koutsojannis C. (2005). Combining expert systems and adaptive hypermedia technologies in a Web based educational system, *Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, pp. 249-253, doi: 10.1109/ICALT.2005.86.

Jurenoks, A. (2017). Adaptive e-learning system based on student activity skills in Moolde system, *Society. Integration. Education. Proceedings of the International Scientific Conference*. 3. 492. doi:10.17770/sie2017vol3.2399

Kardan, Ahmad A., Noorbehbahani, F. (2009). Using Learners' Annotations to Produce Feedbacks for Selecting and Editing Learning Contents in e-Learning Systems, *2009 Sixth International Conference on Information Technology: New Generations*, pp. 1128-1133, doi: 10.1109/ITNG.2009.229.

Klett, F., Pharow, P. (2006). How to Achieve User Satisfaction in Complex E-Learning Environments, *2006 7th International Conference on Information Technology Based Higher Education and Training*, pp. 773-785, doi: 10.1109/ITHET.2006.339699.

Lange, V. (1983). On the Speed of Forgetting. No. 4., pp. 142–145.

Letina, A. (2020). Development of Students' Learning to Learn Competence in Primary Science, *Education Scince*, https://doi.org/10.3390/educsci10110325

Maslo, I., Tilla, I. (2005). Competence as an ideal and analytical category of parenting. Teacher, 3 (51), 4. 9

Mejanova, S., Lebedeva, M., Assylbekova, L., Ldokova, G. (2022). Personalized distance-learning experience in determining students' performance, *World Journal on Educational Technology: Current Issues*, pp. 1428-1437, doi: 10.18844/wjet.v14i5.7086.

Mohamad, M. A., Amron, M. T., Md Noh, N. H. (2021). Assessing the Acceptance of E-Learning via Technology Acceptance Model (TAM), *2021 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1-5, doi: 10.1109/ICRAIE52900.2021.9704019.

Nguyen, L. (2014). A user modeling system for adaptive learning, *2014 International Conference on Interactive Collaborative Learning (ICL)*, pp. 864-866, doi: 10.1109/ICL.2014.7017887.

Novikoff, Timothy P., Kleinberg, Jon M., Strogatz, S. H. (2011). Education of a model student, *PNAS*, vol. 6, https://doi.org/10.1073/pnas.1109863109

Nikitenko, A. (2006). Development and implementation of a hybrid intellectual system, *PhD thesis, Riga Technical university*, Riga, Latvia

Oxman, S., Wong, W. (2014). White Paper: Adaptive Learning Systems. Integrated Education Solutions, *Open Journal of Modern Linguistics*. Vol. 10, No. 3.

Rahimi, R. A. (2020). A Survey of Technology Acceptance Models in the Creative Industry: Exploring Key Limitations, *13th International Conference on Developments in eSystems Engineering (DeSE)*, pp. 9-14, doi: 10.1109/DeSE51703.2020.9450774.

Refaay, E., Ahmed, A., Salem, N. (2017). Development of Web- Based Nutritional Health Education Module. *IOSR Journal of Nursing and Health Science.* 06. 62-74. 10.9790/1959-0603046274

Sani, S., Aris, T. N. M. (2014). Proposal for Ontology Based Approach to Fuzzy Student Model Design, *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, pp. 35-37, doi: 10.1109/ISMS.2014.14.

Santos, J. M., Anido, L., Llamas, M. (2003). On the use of e-learning standards in adaptive learning systems, *Proceedings 3rd IEEE International Conference on Advanced Technologies*, doi: 10.1109/ICALT.2003.1215203.

Santoso, H. B., Isal, R. Y. K., Basaruddin, T., Sadita, L., Schrepp, M. (2014). Research-in-progress: User experience evaluation of Student-Centered E-Learning Environment for computer science program, *2014 3rd International Conference on User Science and Engineering* (i-USEr), pp. 52-55, doi: 10.1109/IUSER.2014.7002676.

Sarkar, S., Huber, M. (2021). Personalized Learning Path Generation in E-Learning Systems using Reinforcement Learning and Generative Adversarial Networks, *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 92-99, doi: 10.1109/SMC52423.2021.9658967.

Srinivasa, K., Kurni, M., Saritha, K. (2022). Pedagogy for E-learning. 10.1007/978-981-19-6734-4_12.

Sumak, B., Podgorelec, V., Karakatic, S., Dolenc, K., Sorgo A. (2019). Development of an Autonomous, Intelligent and Adaptive E-learning System, *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics*, pp. 1492-1497, doi: 10.23919/MIPRO.2019.8756889.

Takahasi, A. (2004). Genetic algorithms in the optimization of complex functions, *PhD thesis, Riga Technical university*, Riga, Latvia

Vaca, J. M., Agudo, J. E., Rico, M. (2012). Adaptive Competence-Based System in an E-learning Platform: Adapting Support Tasks to Students, *2012 IEEE 12th International Conference on Advanced Learning Technologies*, pp. 720-721, doi: 10.1109/ICALT.2012.173.

Volanskaja, T.A. (2002). Methods and technologies of adaptive hypermedia, *Modern problems of program design*, pp. 38–68.

Wang, A. Y. (1983). Individual differences in learning speed, *Journal of Experimental Psychology: Learning, Memory and Cognition*, pp 300–311, https://doi.org/10.1037/0278-7393.9.2.300.

WEB (a). Education for Modern Literacy (Skola 2030): A Curriculum and Approach Description, URL: https://skola2030.lv/admin/filemanager/files/2/prezentacija_izgl_musdienigai.pdf.

WEB (b). Regulations of the Cabinet of Ministers No. 111, (2022) The procedure for organizing and implementing distance learning, (Nr. 6 35.), https://likumi.lv/ta/id/329849-attalinato-macibu-organizesanas-un-istenosanas-kartiba.

Wen, J. R., Yang, S. M. (2005). The effects of learning style and flow experience on the effectiveness of e-learning, *Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, pp. 802-805, doi: 10.1109/ICALT.2005.270.

Zhang, B., Li, Y., Shi, Y., Hou, L. (2020). Recognition and Application of Learner’s Cognitive Ability for Adaptive E-learning, *2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT)*, pp. 62-64, doi: 10.1109/ICALT49669.2020.00025.

Zuters, J. (2007). Neural networks lesson list in the composition model, *PhD thesis, University of Latvia*, Riga, Latvia

Received January 17, 2023, revised February 16, 2023, accepted March 2, 2023