Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 10, Issue. 9, September 2021, pg.80 – 87

An Adaptive E-learning System for Teaching Mobile Applications

George Pashev
Computer Informatics Department, University of Plovdiv "Paisii Hilendarski", Bulgaria
georgepashev@gmail.com
DOI: 10.47760/ijcsmc.2021.v10i09.008

Abstract- Support for various aspects of personalization and adaptation of the reviewed tools is not at the desired level of abstraction - very often depends on the specific way in which it is assumed that a learning object will be combined, to build a learning path for going through an e-course, or to achieve learning goals. This document describes an innovative and more abstract model and related software system for adaptive e-learning. A software prototype for adaptive teaching of the disciplines "Mobile Applications" and "Programming applications for mobile devices", taught at Plovdiv University "Paisii Hilendarski" is presented. Objectives are discussed, based on an extensive overview in the field of adaptive e-learning systems and the software implementation of the e-learning tool is presented. Results of specific tests with study activities will be presented in future scientific publications.

Keywords— e-learning, e-learning goals, e-learning paths, learning objects, adaptive systems

I. Introduction

In the context of e-learning, the level of personalization is a measure of quality and effectiveness of learning. In other words, learning pathways should be distinguished from each other on the basis of learning objectives, knowledge, motivation and individual learning styles of learners. Based on these personal characteristics, a learner can "pass" the course in one or more learning paths. If it is personalized, the learning content is more appropriate and understandable for the learner in the context of his personal goals and achievements.

Support for various aspects of personalization and adaptation of the relevant tools is not at the desired level of abstraction - very often depends on the specific way in which it is assumed that LEARNING OBJECT will be combined, to build a learning path (LEARNING PATH) for going through an e-course, or to achieve learning goals. The selection of a LEARNING OBJECT for inclusion in the LEARNING PATH is usually based on oversimplified principles, not taking into account the need to model or follow more complex training strategies.

By adaptive software system we mean a system that has some of the following characteristics:

- adapts its behavior according to changes in the work environment or parts / modules of the adaptive system itself;
- building modules are developed in order to maximize compliance with changes in the external environment;
- has the means to monitor and control its objective performance and the means to change its parameters, using a closed loop of actions to improve its productivity or to optimize interactions with the user.

The macroadaptive approach, which can be traced back to the 1970s, addresses the adaptation of instructions at the macro level, by allowing various alternatives in the selection of major multidimensional spatial elements such as learning objectives, levels of detail, delivery system and more. In this approach, learning alternatives are selected mainly on the basis of the learners' learning objectives, basic abilities and levels of achievement in the curriculum structure. [1] provides a taxonomy for systematic leadership, where the choice depends on learning objectives (eg development of new skills or Multidimensional spatial expression of learners 'weaknesses) and learners' abilities, such as intellectual abilities, previous achievements, cognitive and learning styles , academic motivation and personality. [2] considers a more practice-oriented model for macro-adaptive CEs, which supports preconditions for learning content, development of appropriate multidimensional space competencies, adaptation of learners' learning styles and achievement of different types of learning goals according to individual needs and opportunities.

An approach to the interaction of abilities treats the adaptation of learning strategies to specific characteristics of the learner. As noted by [3], an e-learning environment that serves a wide range of learners requires a wide range of environments suitable for optimal individual learning. This strategy offers different types of explanations and even different types of media for different learners. Research has been conducted to find a link between learning and abilities. The most important classes of trainees' characteristics can be summarized as follows: intellectual abilities, cognitive styles, learning styles, prior knowledge, impatience, motivation for achievement, self-conditioned effectiveness.

One aspect of the approach is user control over the learning process, according to the learner's abilities, through full or partial control over the style of explanations or the path through the course. [4] defines three levels of control: complete independence; partial control and fixed tasks with multidimensional space control. Research has shown that the success of different levels of learner control is highly dependent on learners' abilities: for example, it is good to limit the control of learners with a low level of prior knowledge. [5] proposes an (eight) step-by-step model to provide practical guidance for applying the model to systems design. According to this model, the course designer must identify the objectives, define the tasks, define the corresponding characteristics of the learners specific to the target group Learning Path, determine how to adapt the explanations and design alternative interpretations. This model seems to be the most practical in the scope of this approach. Others are considered very theoretical, problematic and time consuming.

The microadaptive approach addresses the adaptation of explanations at the micro level, by diagnosing the specific learning needs of the learners, during the performance of the task itself, and not before. Monitoring user behavior and performance, such as the number of wrong responses, response delays, emotional states, and more. can be used to optimize explanatory interpretations [6]. The oldest model of the micro-adaptive approach is the idea of programmed learning, first applied by [7]. A number of microadaptive models differ from the model of the programmed explanation and are developed by multidimensional spatial elementation of a specific model or theory of learning. As described by [8], most microadaptive models "fit" the content of the training during the explanations based on the quantitative presentation of the trait's characteristics. With regard to existing models, such as the mathematical model, the trajectory model, the Bayesian model, the algorithmic approach, etc., micro-adaptive explanatory learning is basically able to adapt to several explanatory variables: for example: the volume of content to be presented, the series of presentations of the content.

In the context of the microadaptive approach, adaptive e-learning is comparable and should be divided into two main processes: The first can be characterized as a diagnostic process that assesses learner characteristics: eg skills, prior knowledge and task indices as a level of learning activity, content structure or conceptual attributes (see [9]). The second can be described as a promising process that optimizes the interaction between learners and the task, by systematically adapting the set and sequence of learning content to the learner's abilities and recent achievements.

Finally, interactive communication is an important element of adaptive learning. The development of a powerful learning system requires a communication model that takes into account the process of interactions between the learner and the learner. Therefore, [10] define two channels for the learning process: the teaching channel for providing content and the assessment channel for monitoring and evaluating the learning process.

The constructivist-interactive approach focuses on contemporary aspects of e-learning can be used in the learning process and follow the constructivist pedagogical approach. An important element of this approach is the Learning Path the need for interactive technologies, which are often considered a natural multidimensional space of e-learning (see [11]).

During the 1980s and 1990s, adaptive multidimensional space-based systems focused mainly on the acquisition of conceptual knowledge and procedural skills (see [12]). Multidimensional space-based learning systems have been criticized by many researchers for their limited scope and adaptability of teaching actions, compared to the rich tactics and strategies used by people - expert teachers. In the 1990s, researchers began to explore techniques such as collaborative and constructivist learning, motivational multidimensional space-etency in adaptive learning systems. By means of constructivist learning theory, the learner plays an active role in the learning process, constructing his own knowledge through experience in the context of the integration of the target area. [13] argues that constructivist learning can be aided by systems intelligence, including mechanisms for presenting knowledge, reasoning, and drawing conclusions. Some new adaptive e-learning systems take motivational factors into account, combining a training plan with a motivational plan. As shown by[14], learning planning can be divided into two streams, content planning for selecting the next teaching topic, and delivery planning are determining how to teach the chosen topic. Motivational multidimensional space elements must be limited in the context of delivery planning.

One of the most popular and well-documented ADAPTIVE E-LEARNING SYSTEMs is the Adaptive Course Generation System (ACGs) [11]. AHA! ([12]), Adaptive Learning Environment (ALE) [13], [14], NetCoach [15], PERSO [16] Each of them was be briefly reviewed and analyzed.

Based on the considered essences of adaptive learning, defined task of adaptive e-learning, we can make the following classification scheme of ASEO:

- Each of ADAPTIVE E-LEARNING SYSTEM can be classified according to combinations of basic parameters in the above discussed Task for training and adaptation chains, incl. Ψ - state of the external environment; Y - state of the object; I Ψ, IY - corresponding meters; Y 'and Ψ' - the measurement results; X - Learning Patrolling (training and controlling) impacts; Dx - resources (limitations of Learning Management); Z * - learning objectives of the Management, necessary for the transition of a learner in state Y *;
- Support for various adaptation circuits
- Supported appearance of MULTIDIMENSIONAL SPACE;
- Level of virtualization and modeling of subjects (Trainee, Teacher, Author, Others);
- Level of diversity of types of learning activities (LEARNING ACTIVITIES): textbooks, tests, forum, announcements, all kinds of learning activities);
- Ability to integrate with existing software systems (SOFTWARE SYSTEMS).

Based on the review, we can identify the following areas for modeling and creating ASEO:

- Creating adequate models of:
  - o entities, information resources, material resources, processes, process instances, messages, metascience (SA);
  - o ADAPTIVE E-LEARNING SYSTEM framework as a process of adaptive learning as a more general case of a Learning Path Management process of different subjects / processes / instances of processes, and not only from the point of view of Learning Path Management on learners, as done in [17];
  - o each SA both virtually and as a real object, and through the GPI.
- Functionality for:
  - o setting management goals;
  - o description and compliance with MULTIDIMENSIONAL SPACE;
  - o flexibility in determining the corresponding meters I Ψ, IY and the method of measurement;
  - o easy integration with existing SOFTWARE SYSTEMS;
  - o wide variety of LEARNING ACTIVITIES.
- Feedback:
  - o for flexible correction of the parameters of the subjects and their properties in Adaptive E-learning Systems;
  - o adaptive identification of these parameters during the operation of the system (for example, a level in Bloom's Taxonomy can be dynamically calculated for a trainee after passing a test procedure, as well as other parameters);
  - o structural identification, incl. and automated collection of metadata for various objects and subprocesses in the process of adaptive e-learning. Examples of this are automated generation of tests to assess a learner after passing a certain LEARNING ACTIVITIES, in the context of achieving a certain learning goal (LEARNING GOAL), as well as automated annotation of LEARNING ACTIVITIES (collection of their metadata) , by generating TE to be solved by trainees;
  - o adjustment of learning objectives.

In the implementation of the prototype used for training in the discipline "Mobile Application" and the discipline "Programming Applications for Mobile Devices" at Plovdiv University "Paisii Hilendarski", we tried to achieve most of the areas mentioned in the previous paragraph.

II. OUR APPROACH

Figure 1 shows an architectural scheme of the application, where it can be seen how the various multidimensional space elements are integrated in the scheme, both conceptually and in the current experimental setting. Users access through the PeuPortal portal, which is uPortal based. In it, the user can access it with his single university account through a central authentication service (CAS Server).

In the overall architectural framework, the interface of most of the applications is automatically generatied by the workflow execution system TinyWf, which is a result of the current study (see ([18]). However, it is permissible to use views from web applications that are not implemented with the TinyWf workflow framework, including those implemented with other workflow execution frameworks.

When a user logs into the portal, through his e-portfolio will appear the menus and portlets in them, for which he personally or as a participant in a user group has the right to view (visualize) and use. Each uPortal portlet corresponds to exactly one workflow that is run by the TinyWf environment. Most of the workflows turn to web methods of the respective .NET REST API.

REST services to university systems such as PeU (information system of Paisii Hilendarski University), Moodle (for open source e-learning based on PHP), Jasper Reports Server (for creating and managing reports with dynamic content with open source based on Java), etc.

Jasper Reports Server is accessed through library extensions in TinyWf, Moodle REST API - directly from TinyWf, and the various REST services consumed by TinyWf access the DBMS of the application systems for which they are built. [18]

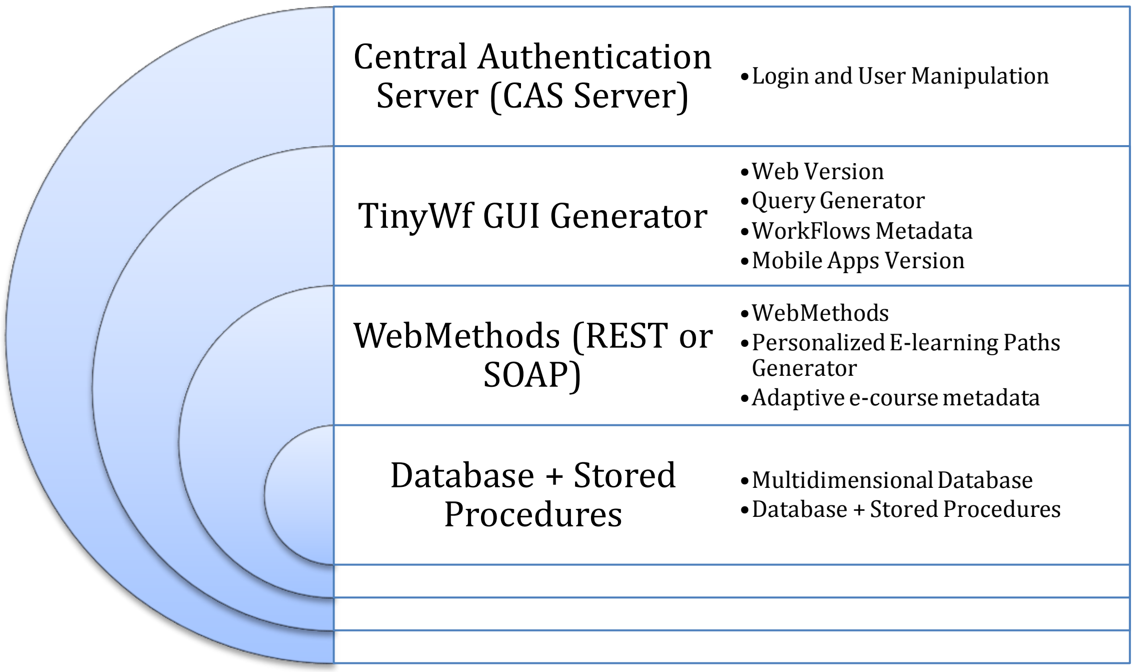

Figure 1.Application view of the Adaptive E-learning System

Jasper has a powerful tool for creating and displaying reference templates in various formats (pdf, xls, doc, html, etc.). Templates can be stored on Jasper Server and accessed remotely. [18] Thearchitecture of the application is 4-layer and includes:

Layer 1. Relational database containing references, stored procedures, views and functions. Database queries sent from higher layers are located here in order to achieve the highest possible query execution speed. The principle of the closest placement of data and means for their processing, applied in optimization algorithms, is applied;

Layer 2. Transport layer (for connection between layer 1. and layer 2.);

Layer3. Defining work processes, incl. to generate an interface via REST or SOAP protocols. URL templates are used to define business process nodes;

Layer 4. A user layer that provides access to workflows (depending on user roles, rights, and personal settings) to processes on the system itself.

A relational model is chosen for the realization of layer 1.. The structure of the database in a third normal form. To facilitate the process of generating the interface in layer 3., a convention for naming the objects in the database has been adopted. The name of the field that represents a foreign key to a table begins with "ID_" and then the name of the linked table is displayed. The user actions related to the data in the database are encapsulated in stored procedures, views and functions (Fig. 2).

Much of the business logic is stored in a similar way. This allows the upper layers to focus on the types of activities that are more appropriate for their location.

Transport layer 2. is essentially a service provider. The web service is a method of communication between two electronic devices in a network, which in practice provides functionality available at a network address in the web space. The W3C defines a web service as a software system designed to support communication between interoperable machines over a network. Protocols such as REST, SOAP, etc. are supported.

Layer 2. is an important part of the integration because it provides Layer 3 with web services for consumption. The latter can be from different types of applications, which is the basis of their integration.

Each integrated application includes a set of workflows. To design and create an integrated application, it is necessary to implement a number of workflows, accompanied by appropriate user interfaces. The following approach has been adopted to automate these activities:

Based on a formal workflow model, tools are provided to describe specific work processes;

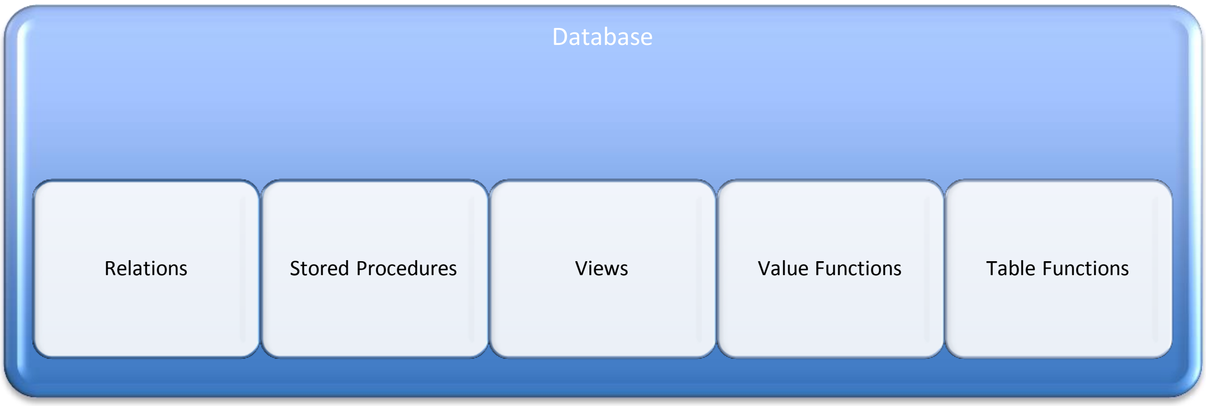

Figure 2. Database layer objects

Layer 3. User Interface Generation Layer

Each state of a business process is represented by a node of a set N and is characterized by certain sets of input and output attributes (data). In terms of our architecture, each workflow state corresponds to at least one web method call from the transport layer. Business processes are set within a user group (configurations of different user groups are possible). The input data (attributes) of the state are determined manually (input by an authorized user) or automatically - by projecting a session variable or "mapping" attributes of the output state to the previous state through which the process was executed. The output data (attributes) of the status is determined after the user selects one of the possible results of the implementation of the web method corresponding to the status. The attributes of this result are the output attributes of the state. Output (as well as input) attributes of a state are designed as session variables of input attributes of other states. Some of the possible subsequent states are filtered on the basis of conditions that can be set when describing the respective business process.

Layer 4. User layer

To implement layer 4. of the architecture requires a web portal with the following minimum functionalities:

- simultaneous access of a user to different types of systems depending on his roles and rights in them;
- general view of the available functionalities, respectively work processes in the electronic portfolio of the user;
- support for LDAP requests and relational databases;
- providing consumer services, multidimensional space elements and frameworks for connecting more complex systems;
- flexible user group management system, incl. realization of invested groups;
- possibilities for using API, etc.

Based on research and analysis of various web portals, uPortal was chosen for the experimental implementation of layer 4.. In addition to the above features, uPortal has other useful features: it is

- based on open standards, Java, XML, XSLT, JSP and J2EE;
- used by millions of users;
- has options for user identification, incl. LDAP and database;
- offers links to libraries and integration tools such as CAS;
- single sign-on, etc.

TinyWf (from Tiny Workflow Framework) is a software framework implemented in PHP and Python 3 [18], on the basis of which applications can be designed, integrating functionalities of heterogeneous systems and using formally defined business processes.

The main features that TINYWF offers to designers of integrated applications are:

- tools for describing business processes (in the form of relatively standardized PHP code to represent states and transitions);

- automatic generation of input and output graphical interface in all current states of the business processes available to a given user (eg filling in list fields from certain nomenclatures);
- automatic provision of transitions to subsequent states based on the descriptions of the respective business processes;
- automatic matching (mapping) of output data from one state to input data to another state;
- maintaining memory for user selection in previous states (implemented with session cookies);
- clear memory for previous selection when going through a recursive connection to an automatic memory state.
- Possible perspectives for development of the presented approach for integration, incl. for the creation of integrated SEs are related to:
- implementation of machine learning in Layer 3., which will allow "smart" rearrangement of possible subsequent transitions;
- adding an abstract sublayer to layer 1. to allow cross-dimensional spatialization for different DBMS by defining database objects in a platform independent language and automatically generating DDL, DML, DCL for target DBMS;
- use of a graphic editor to define business processes, etc.

Module "Trainee in a course"

In the beginning the student has the opportunity to choose one of the adaptive e-courses designed for him. When choosing a current e-course, the visualized[1] lists of achieved and not achieved (yet) learning goals, as well as the learning units from the PM, on which the learner has "worked" - with the possibility for re-access or refinement. The learner may also choose a new learning goal to achieve. The goals are arranged according to "proximity" to the current state of the learner, represented by a dot in the multidimensional space.

In this sense, the first goal in the list of unachieved learning goals is the easiest to achieve. However, the learner can also choose a more distant goal as a consequence of his motivation to achieve a "higher" goal faster and in a more intense mode of learning. When choosing a specific goal, we move on to generating a personalized learning path for the learner. The generated personalized learning path is a list of new learning objects that the learner must "pass" successfully in order to achieve the learning goal. Each of the training sites in the PM may be accompanied (or not) by sufficiently comprehensive metadata that determines its location in space. If the unit has been recently added, it is assumed that the accumulated metadata is probably not yet adequate to its content.

To determine the condition for the adequacy of the available metadata, the following metric is used: number of generating accumulative tests performed for the respective study unit. With a relatively small number, the trainee is offered an automatically generated cumulative test. After the student completes the test, a signal is sent to the teacher that a new solution has been proposed, which must be checked and evaluated. When the answer is assessed by the teacher as correct, a function / predicate (and metadata) is automatically generated, with which the respective learning unit will continue to participate in e-learning and in the automatic generation of learning paths.

The algorithm for generating custom learning paths (eLP_GA) starts with the initialization of some variables such as *studPoint* (the current presentation of a learner in a MULTIDIMENSIONAL SPACE). For this trainee, a value of the maximum reach radius is initialized *maxReachRadius*, checking whether there is already a dynamically calculated one for the trainee, and if not, it sets a default value (lines 2-3 of Fig. 3).

Lines 5-7 perform the achievement of unfulfilled goals for the current trainee within the current training application. If such targets are not found, the algorithm is exited with the corresponding message.

In the presence of unfulfilled learning goals, the first of them becomes current (line 9). In addition, an empty list of Learning Activities is initiated (on line 11), which will eventually, after the completion of the algorithm, contain Learning Activities with which the learner could achieve the current Learning Objective.

Line 13 checks whether the selected learning objective contains predicates that indicate the presence of compulsory learning activities to achieve this learning goal, and if they do, they are added to the list of Learning activities On line 15 in the temporary object of Learning Objective In memory, those dimensions that contain these predicates are removed so that the algorithm can work properly afterwards.

Each of the Learning Activities that are already on the list may contain additional predicates / functions in the dimensions, such that when the learner passes through them, his radius-vector in MULTIDIMENSIONAL SPACE achieves a LEARNING OBJECTIVE. That is why its point "moves" into the dynamic memory of MULTIDIMENSIONAL SPACE, as if it had passed through the available Learning activities (line 19). Line 21 in MULTIDIMENSIONAL SPACE is looking for such LEARNING ACTIVITY that would be potential for achieving a LEARNING OBJECTIVE: at the same time they are arranged close to the LEARNING

OBJECTIVE and are in the field of reachability defined by the reach radius and the central point of learning. in MULTIDIMENSIONAL SPACE. Demand is further optimized by adding to the filter a criterion that the learning activity should not be offered to students more than 2 times when generating different learning paths.

Lines 23-25 check whether the point of the learner in MULTIDIMENSIONAL SPACE has not moved to MULTIDIMENSIONAL SPACE so that the LEARNING OBJECTIVE has already been achieved. If so, exit the algorithm with a success message and the available list of LEARNING ACTIVITIES.

Line 26 checks whether the list of potential LEARNING ACTIVITIES is empty and if so, we enter the hypothesis of impossibility to generate a list of LEARNING ACTIVITIES and exit the algorithm with such a message.

```
1  studPoint = RVec(SO); {locate student point in MDS }
2  maxReachRadius=((a=V_L (studPoint,[(maxReach,[SO])]))===null)
3  ?MAX_RADIUS_VAL:a;
4
5  uncompCGs=orderTo(studPoint,searchGE(studPoint, maxReachRadius, "isCourseGoal = 1 AND
   CompletedBY("+SO.ID+") <> 1"))
6
7  if(incompCGs.length===0) exit("No course goals available right now.");
8
9  selCG=uncompCGs[0];
10
11 retLAs=[];
12
13 V_L(selCG,[(obligLA,[:id_CG])]).foreach(function(dimObject){retLAs.push(getLA(dimObject.val))});
14
15 V_L (selCG,[(obligLA,[:id_CG ])]).
16 foreach(function(dimObject){dimObject=null});
17     while(1){
18
19         foreach(curRetLA  in
           retLAs){V_L(curRetLA).foreach(function(dimObject){studPoint.addOrUpdate(dimObject)})}
20
21         potentialLAs=orderTo(studPoint,searchGE(studPoint, maxReachRadius, "isLA = 1 AND
           prevPresentedTo("+SO.ID+")<3"));
22         C: /*Check if studPoint has all dimension values greater or equal than those of selCG,
           i.e. studPoint has accomplished selCG;*/
23         if(isGE(studPoint, selCG)){
24             exit("successful path generation", retLAs);
25         }
26         if(potentialLAs.length===0) exit("Error: no suitable LAs found");
27         curLA = potentialLAs[0];
28         potentialLAs.shift();
29         curLA.push(curLA);
30         V_L(curLA).foreach(function(dimObject){studPoint.addOrUpdate(dimObject)});
31 goto C;
32 }
33
```

Figure 3. Algorithm for Personalized eLP Generation

In the presence of LEARNING ACTIVITY in the list of potential, on the principle of the queue is removed the first in the list and entered in the list of LEARNING ACTIVITY, which should achieve a LEARNING OBJECTIVE. Subsequently, the learner's point in MULTIDIMENSIONAL SPACE is moved as if it has passed through this LEARNING ACTIVITY and a new iteration of the algorithm is repeated.

It should be noted that some optimization of the generated list of LEARNING OBJECTIVES is achieved, because the list of potential LEARNING ACTIVITIES is arranged in close proximity of LEARNING ACTIVITIES to LEARNING OBJECTIVES. Another optimization is related to checking before entering the cycle of the algorithm whether in the field of reachability of the trainee there is not at least 1 LEARNING ACTIVITY, which is as close as possible to LEARNING OBJECTIVE, but with greater or equal coordinates for the dimensions of LEARNING OBJECTIVE. achieving a LEARNING OBJECTIVE with only 1 LEARNING ACTIVITY).

A set of Course goals has been defined in account to *the Training of Mobile Applications*. Goals are sorted in order of their achievability from a beginner in the Adaptive e-course. Furthermore, Goals are defined with a set of Concepts for each goal, and a level of difficulty for each concept:

Goal 1: LinearLayout, EditText, TextView, Button;
Goal 2: ListView, Database, SQLite3, CrudOperations;
Goal 3: "Database pro" Database, SQLite3, CrudOperations, DBHelper, DBActivity;

Goal 4: RestServices, RestFul, RestIsh, SyncCommunication, AsyncCommunication, Threads, ThreadPool, AsyncTasks;
Goal 5: "Map Wizard" GPS, GoogleMaps, LocationListener, LocationRetrieve;
…..

Figure 4. A part of the set of Course Goals of Adaptive Course for training Mobile Applications

Each of the enlisted concepts for each goal, some of which are shown in Figure 4, are assigned with difficulty coefficient by the teacher, who has prepared the adaptive e-course. Optionally a name is assigned to some of the goals. If a student wants to build a simple DB Application, then she/he can choose to achieve Goal 2 in the adaptive e-course and the system chooses Learning Activities that are suitable for this goal. However, if a student wants to build more sophisticated DB Apps with map localization features, then she/he can choose Goal 5 "Map Wizard"combined with Goal 3 "Database pro", so that a corresponding e-learning path can be generated. The UI supports selection of multiple course goals by the trainee.

III. CONCLUSIONS

The paper demonstrates an innovative approach towards the generation of adaptive E-learning Courses, based on personal learning goals. A prototype in the field of education of "Mobile Applications" is discussed. Further testing with a large number of students needs to be performed. The algorithm for generation of learning paths can be improved in account to performance, by using graphs with distance metrics instead of vector calculus in multidimensional spaces, for instance.

ACKNOWLEDGEMENT

The paper is supported within the National Scientific Program "Young scientists and Post-doctoral students" in accordance with Appendix No. 11 of Council of Ministers Decision No. 577 of 17 August 2018.

REFERENCES

[1]. Zuckerman, M. P. (1983). Sensation seeking and arousal systems. Personality and Individual Differences 4.4
[2]. Glaser, R. (1977). Adaptive education: Individual diversity and learning. Holt McDougal.
[3]. Cronbach LJ. The two disciplines of scientific psychology. American psychologist. 1957 Nov;12(11):671.
[4]. Snow CC, Hrebiniak LG. Strategy, distinctive competence, and organizational performance. Administrative science quarterly. 1980 Jun 1:317-36.
[5]. Carrier J, Greengard L, Rokhlin V. A fast adaptive multipole algorithm for particle simulations. *SIAM journal on scientific and statistical computing*. 1988 Jul;9(4):669-86.
[6]. Federico, P.-A. (1983). Changes in The Cognitive Components of Achievement as Students Proceed .*In Journal of Computer-Based Instruction.*
[7]. Pressey LC. Proficiency in silent reading. School & Society. 1926.
[8]. Suppes, P. J. (1976). Models of individual trajectories in computer-assisted instruction for deaf students. In *Journal of Educational Psychology 68.2.*
[9]. Rothen W, Tennyson RD. Application of bayes' theory in designing computer-based adaptive instructional strategies. Educational Psychologist. 1978 Jan 1;12(3):317-23.
[10]. Wasson, B. (1990). Determining the focus of instructions: Content planning for intelligent tutoring systems. *In Dissertation, University of Saskatchewan.*
[11]. Viet AN, Si DH. Acgs: Adaptive course generation system-an efficient approach to build e-learning course. *InThe Sixth IEEE International Conference on Computer and Information Technology (CIT'06)* 2006 Sep 20 (pp. 259-259). IEEE.
[12]. De Bra P, Smits D, Stash N. The design of AHA*!. In Proceedings of the seventeenth conference on Hypertext and hypermedia* 2006 Aug 22 (pp. 171-195).
[13]. Adank P. The neural bases of difficult speech comprehension and speech production: Two Activation Likelihood Estimation (ALE) meta-analyses. *Brain and language*. 2012 Jul 1;122(1):42-54.
[14]. Methods and techniques of adaptive hypermedia, 1996
[15]. Loewenstein GF, Weber EU, Hsee CK, Welch N. Risk as feelings. *Psychological bulletin*. 2001 Mar;127(2):267.
[16]. Guettat B, Chorfi H, Jemni M. Customized learning environment based on heterogeneous traces. *In2010 International Conference on e-Education, e-Business, e-Management and e-Learnin*g 2010 Jan 22 (pp. 193-197). IEEE.
[17]. Rastrigin LA. Adaptation of complex systems. Riga: Zinatne. 1981; 386.
[18]. Pashev G, Alexander T, Evgeniya A, Georgi T. Integration of heterogeneous information systems in the field of e-education. *In Proceedings of the Annual Conference of the Union of Scientists Plovdiv - Bulgaria.* 2015;17.

