Virtual Training Environment with Diagrammatic Models

Alexander Afanasyev
«Computing technique» department
Ulyanovsk State Technical University
Ulyanovsk, Russia
a.afanasev@ulstu.ru

Nikolay Voit
«Computing technique» department
Ulyanovsk State Technical University
Ulyanovsk, Russia
n.voit@ulstu.ru

Dmitry Kanev
«Computing technique» department
Ulyanovsk State Technical University
Ulyanovsk, Russia
dima.kanev@gmail.com

***Abstract*—The authors proposed models, methods, algorithm, structure and scheme of recommendations of the intellectual learning environment, taking into account the dynamic change of knowledge and personalized learning. The complex information and learning system with internal and external sources of knowledge and with intelligent sensors was developed. The main mathematical apparatus of the organization and development of virtual training environment are expert-training and adaptive approaches. Models of virtual simulators are described in set theoretic language. The intelligent environment analyzes the progress of students' work with simulators and computer aided systems. The state machine is used to control student behavior in virtual training environment. The novelty of the work is the formalization of the set-theoretic language of semantic errors of the learning scenario. The experiment of teaching students in the intellectual environment is given, the result of this experiment is given as a proof of reduction of terms of training. We developed a system of virtual simulators that is an integral part of intelligent learning environment and is used in training programs of radio engineering, instrument engineering and computer techniques.**

Keywords—intelligent learning environment, enterprise experience, adaptive learning, virtual simulator system, workflows

I. Introduction

Training with the use of computer technology has been conducted since the 60s of the 20th century, significant progress has been achieved: training machines, systems and complexes (environments) have been developed to automate the learning process, make it subject-oriented and individual. However, the main problem of optimal interaction of subsystems within a single system in the design and organization of automated training systems is not solved, given the variety of its subsystems. The most effective direction in the organization of training systems is the use of intellectual means. The use of such funds for corporate training working professions and engineering specialties in industrial plants using computer aided systems (CAD systems) (for example, KOMPAS-3D, Vertical, Loodsman-PDM, Quartus II, Altium Designer, NX, Ansys) in mechanical engineering, aviation, shipbuilding, military-industrial complex (hereinafter referred to as the project-industrial enterprise), requires an analysis of personal, including professional competence, characteristics of learners and organization on this basis, and taking into account enterprise's knowledge base, automated learning. The solution of this problem is an actual fundamental scientific and technical problem. The purpose of the fundamental research is to develop the theory of development of intellectual means of teaching project activities, including a new scientific approach, principles, complex computer models and methods that are the basis of mathematical, software and information support of training machines, systems and systems (environments), networks in terms of their application in the design and industrial enterprises.

Today the main direction of development of computer-based training systems is their intellectualization focused on individualization and personalization of training, collaborative solving problem, improvement of relationship among participants in the learning process. From a methodological point of view, the organization and development of the modern training systems, based on the network information technologies, are complex intelligent environments with virtual communities. The goals of this research work are studies, development and implementation of intelligent learning environment (ILE). The distinctive features of this environment are the integration of training systems with corporate enterprise knowledge base and availability of the virtual production world. In the frame of practice-based learning these environments are effectively used in both traditional and corporate (within corporations and enterprises) universities. In order to achieve these goals, we need to solve the following tasks: to analyze the computer-based intelligent learning systems; to develop an architecture, methods, models and algorithms of ILEs' operation for universities and business companies; to develop a software and data support of ILEs; to synthesize an experience of ILEs' implementation and use; to analyze and control diagrammatic workflow (e.g. UML, IDEF, BPMN) it is proposed a method based on multilevel automaton grammars [16].

II. Related Work

The main approaches to organization and development ILEs are expert-training and adaptive approaches. Expert-training systems (ETSs) are based on integration of computer-based learning technologies with expert systems. They are intended for solving the low structured problems and use both a reference knowledge and an acquired by students' knowledge.

An individual dynamic learning path is based on the modality of logical conclusions. ETS should be used for solving the following tasks: learning management based on the individual student's training level and his/her personal characteristics (creation of a personalized learning path); diagnosing and forecasting of quality of acquiring the subject information to change the sequence of given learning materials; keeping up a learner's professional level in a given subject domain. The examples of ETS are EDAAS [1] and GCA [2]. The analysis of ETS development and use has revealed a number of problems: to form and formalize knowledge is very difficult in practice. Both qualified experts and engineers on knowledge database are required to solve this problem; a large amount of well-structured qualimetric materials (tests, project tasks) are required; there are no mechanisms of corporate education. To solve modern design and technological problems the approach based on corporate work is fundamental; ETSs have weak explanatory capacities, there is no differentiation of explanation depending on user's qualification and experience. Adaptive training systems (ATSs) implement a feedback between a learner and the systems used for educational process controlling. To create an individual learning path, the methods and models of neural networks theories, graphs, fuzzy data sets, classifications, agent-based systems, and other means are used. In research works [3, 4, 5, 6, 7, 8, 9, 10, 11, 12] the models and algorithms of adaptive learning are proposed. Their disadvantages are: the lack of dynamic content of a domain model that doesn't allow it to automatically include hybrid knowledge and units of an enterprise experience in the learning process, great complexity of filling with training material, the lack of the virtual component (first of all, the lack of simulator systems), which complicates learner's practical competences formation.

In the part of adaptive computer systems of training a number of methods and means of adaptive management of process of training is considered. The paper [8] describes a system of automatic personal training recommendations based on interests through the analysis of visited web pages, using content-based filtering and collaborative. In [9] adaptation takes place after the test; the system calculates the level of knowledge and updates the model of the student. If the student does not score enough after the test, the system offers links to the sections of the current chapter that need to be studied. In [10] the model of adaptation is based on products and is divided into two layers: the choice of concepts from the knowledge space of the subject area for study and the choice of the method of presentation of the material. In the AEHS-LS system [11], the training scenario is chosen based on the style of thinking, learning style and knowledge level. In QuizGuide [19], the learning management subsystem directs students to the most appropriate topics, which are selected based on the learning objective and level of knowledge. The GRAPPLE system [12, 13] model of adaptation is based on the products. In PCMAT [20], the learning management model is based on adaptation rules. The rules are based on student behavior, knowledge, and learning style. In the considered systems, adaptive management of the learning process takes place either on the basis of choosing a suitable, already laid down learning scenario, or on the construction of an individual learning scenario that requires time-consuming manual work to create adaptation rules. The general disadvantages of the considered approaches are: the lack of dynamic content of the domain model, which does not automatically include hybrid knowledge and experience of the enterprise in the learning process; great complexity of filling educational and practical material. The analysis showed that the automated training systems do not take into account the specifics of the training of project activities, they do not integrate with the competence profile of the designer and the assessment of its project activities, as well as communication with the base of design solutions of the enterprise. In the part of recommendation systems, the most important works are considered [21, 22, 23, 24, 25], the main classes of methods for the construction of such systems are: methods of collaborative filtering, methods that analyze the content of objects, and methods based on knowledge. Based on the analysis, it is concluded that none of the presented systems is suitable for making recommendations based on the protocol of project operations. The subject of our work is to solve these problems.

III. Models and Methods of Intelligent Learning Environment

The complex information and learning system with internal and external sources of knowledge and with intelligent sensors was developed to solve the second task. We suggest such an approach of creating the ILEs, which is based on the principles of formal and informal learning methods' integration, of personalized learning and formation of a competency based on both training and real knowledge (the last is an experience of enterprises and organizations acting in real economics sector used as hybrid knowledge bases), the active use of simulators, virtual worlds and augmented reality. The structure of the intelligent learning environment is given in Fig. 1.

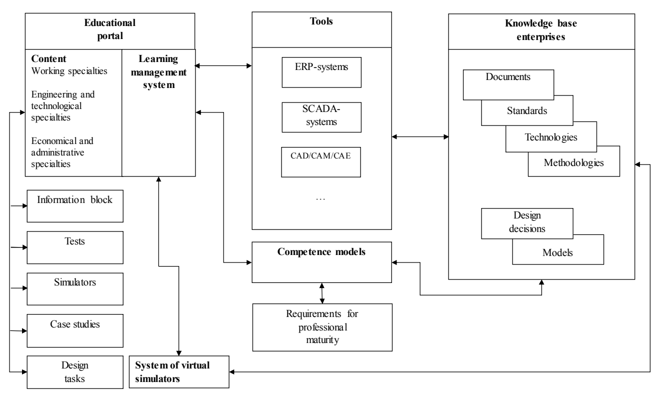

Fig. 1. The structure of the intelligent learning environment.

The structure of the intelligent learning environment contains the following components: *Information block* is office documents; *Tests* are question sets; *Simulators* are the applications for skills testing; *Case studies* are laboratory works; *Design tasks* are course works; *System of virtual simulators* are training systems; *ERP-systems* are planning systems; *SCADA-systems* are control and data collection systems; *CAD/CAM/CAE* systems are design systems; *Competition model* is the profile of the user; *Requirements for professional maturity* are skills; *Documents* are files in the format DOC, PDF, etc. *Standards* are a set of standards, such as ISO; *Technologies* are a set of technologies such as Component Object Model; *Methodologies* are a variety of methodologies such as Rational Unified Process. To implement the principles of intelligent environment organization and functioning we propose a number of models and methods: a dynamic multilevel domain model; an overlay learner model; a model of learning process scenario as a dynamic oriented graph; a method for controlling and diagnosing of a learner's qualitative characteristics; a method for generating the dynamic adaptive personalized path of a learner; a method for integration of based learning platform with social networks and online learning technologies.

The authors have proposed and investigated two approaches to the development and integration of models and methods. The first approach was given in [13]. It is concerned with the representation of a domain model as an associative multilevel model, the use of modified classification method based on the fuzzy Kohonen maps as a basic tool for monitoring and diagnosis of learner's dynamic competences, and the proposed method of generating personalized learning paths. The second approach was given in [14]. It is associated with the ontological representation of the subject domain and the development of the method of generating dynamic learning path based on it. The algorithm dynamically generating the ontological domain model has been proposed. It provides the "relation" of the learning process with real knowledge of an enterprise. The formalized experience of an enterprise is represented as ontology of a subject domain which definitions connect with a hybrid knowledge base. It stores the workflow models, design solution models, technical process models, activity models (software libraries used to reuse), and documents. These artifacts of experience have the metadata for their description and inclusion in creating a personalized learning path. We developed a system of virtual simulators that is an integral part of ILEs and is used in training programs of radio engineering, instrument engineering and computer techniques [15, 17, 18]. The basis of the implementation of simulator's system is the author's model of a virtual automaton-based simulator. A distinctive feature of this model is the availability of functional blocks (groups of states) and the error states, allowing the authors to extend the class of learner's diagnosable errors. The model of a virtual simulator is as follows: *The simulator* = (*a set of states*, *a set of control actions*, *a set of rules*, *a set of functional blocks*, *a set of error states*) where *a set of states* are the states determined by the space of simulator settings, *a set of control actions* are a set of possible user's actions with a simulator; *a set of rules* are a condition for change of a simulator's state from one to another; *a set of functional blocks* is separate parts of a simulator, that independently performs its task and does not overlap with other blocks; *a set of error states* are a set of states that characterize the nonfunctional simulator. *Rule* = (*condition*, *control action*, *next state*) where *state* is a state in which the simulator should apply the rule, *control action* is an action of a trainee which starts changing the state, *next state* is a state in which the simulator will be after the application of the rule. *Functional block* = (*a set of rules*, *a set of target rules*) where *a set of rules* are rules that are executed within the functional block; *a set of target rules* means that each block has its own purpose (e.g., to set the start mode), the last rule that is applied to achieve the purpose of the block is the target rule. A description of an algorithm of simulator development is the following: select a set of a simulated object's controlling components, create a set of states for each component, select simulator's parameters and make the dependence of states on parameters, select a set of control actions on the simulated object, develop the rules for changing parameters depending on the control action, set the initial settings. To check learner's skills in a simulator he (she) is given exercises. They are characterized by the initial and final states: *Exercise* = (*an initial state*, *a final state*) where *an initial state* is a state from which the exercise should be started; *a final state* is the state in which the exercise is considered to be done. During the exercise, the learner generates a control action in the simulator. Start the step counter, every control action increments the counter by 1. The control action initiates a change of the simulated object's state. The simulated object's state after the clock cycle t is denoted as $S(t)$. The rule used for transition from state $S(t\text{-}1)$ to $S(t)$ is denoted as $R(t)$. The authors classified learners' errors and developed rules to define them. The loop: $S(t) == S(t\text{-}n)$, $n=1...t$. Back to the state that it was in the past, a special case – there is no change of state after the control action. Unchanged state $S(t) == S(t\text{-}1)$. The control action doesn't change the simulator's state. The state of simulated object's is equal to the previous state, i.e. a special case of loop. A simulated object's error $S(t) \in$ *a* set of error states. The user set the simulator in a faulty state. The current state belongs to a set of error states. Achieving the goal again: $\exists b \in$ a set of functional blocks $\wedge R(t) \in$ a set of target rules of block $b \wedge \exists n \neq t$: $R(n) \in$ a set of target rules of block b. A learner changed a parameter the second time, e.g. he (she) first set a 10-second period, then an 8-second period. Transition between blocks $\exists b1$, $b2 \in$ a set of functional blocks $\wedge R(t) \in$ a set of rules of block $b1 \wedge \exists n \neq t$: $R(n) \in$ *a set of target rules* of block $b2$. The learner first performed the exercise within one functional block, and then he started performing tasks in the framework of another block, e.g., he began to set the period to 15 seconds, didn't finish and began to set the internal start mode. Reusing a functional block: $\exists b \in$ a set of functional blocks $\wedge R(t) \in$ a set of rules of block $b \wedge \exists n \neq t$: $R(n) \in$ *a set of target rules* of block b. A learner began to change the parameter changed before he did it. Using a functional block without achieving any goal: $\exists b \in$ *a set of functional blocks*, $\exists k$: $R(k) \in$ *a set of rules* of block $b \wedge \nexists n$: $R(n) \in$ *a set of target rules* of block $b \wedge S(t) \in$ *a set of final states* of the exercise. A learner performed an unnecessary action that does not influence on the result of the exercise, e.g., a learner started to adjust the amplitude of the signal when it was not required to complete the task successfully. An important issue in the organization of ILEs is their relation with the software tools used in education, such as computer-aided design (CAD). Bellow the authors proposed the method of forming a recommendation when working with CAD. To make guidelines in CAD authors should analyze the sequence of operations carried out by the user. The general scheme is shown in Fig. 2.

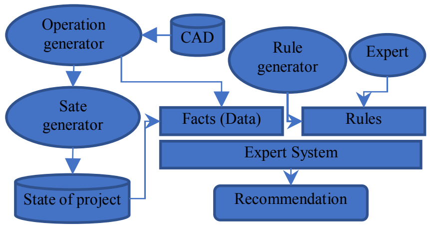

Fig. 2. The general scheme of forming a recommendation when working with CAD.

Operation generator is embedded in CAD, processes user's actions, and encodes them as follows: *Operation*= (*code*, *version* (*kind*), *SettingsOFoperations*), where *code* is a unique identifier of the operation; *kind* is the type of operation in CAD, eg. extrusion, rotation, kinematic operation, loft, sketches, Boolean operation, cut, fillet, chamfer, round hole, slope, copy, paste; *SettingsOFoperations* is a multiplicity of *SettingsOFoperations*. *SettingsOFoperations* = (*Key*, *Definition*) where *Key* is a line, name of setting; *Definition* is a (*Figure*|*Line*|*Object*), setting's definition. *Operations* are sent to expert system (ES) as facts (data). State generator generates a state of the project on the basis of operations *StateofProject* = (*MultiplicityofOperations*). A learner's operation under the number t is denoted by an *Operation*(*t*), and the state took over the project, is as *StateofProject*(*t*). *Rule* is a function defined on the set of operations and states. Each recommendation has a rule which activates this recommendation *Recommendation*= (*Message*, *Rule*). The message is the text of the recommendation *Rule*=*Operation*→{0,1}. The rules are filled either manually by the expert or generated automatically for a class of recommendations. The example of recommendations based on operations after *i*-th operation (do not build a chamfer for each edge separately): *Message* = "do not build a facet for each edge separately". If it is possible, when you create the chamfer you should register as much as possible the number of edges, the chamfer settings for which are the same: *Rule* = *Operation*(*i*).*kind*=*Chamfer*)^∃*t*^*Operation*(*t*).*Settings*=*Operation*(0).*Settings*. Example recommendations on the basis of the state after *i*-th operation (Loop): *Message* = "back to the state in the past", *Rule* = ∃*t*^*StateofProject*(i)=*StateofProject*(*t*)^*i*<>*t*. The algorithm for generating recommendations:

1) *A student gets started working with a project.*
2) *If a project is new, go to step 5.*
3) *Generating the operations based on an available project.*
4) *Adding an operation to the sequence of operations.*
5) *Reading learner's control actions.*
6) *Generating operations on the learner's actions.*
7) *Adding an operation to the sequence of operations.*
8) *Forming the state of the project on the basis of the sequence of operations.*
9) *Adding the project state to the sequence of project states.*
10) *Searching a rule that matches to the sequence of the operation.*
11) *If a rule is not found go to step 13.*
12) *Adding recommendations to the learner's individual list and outputting them on display.*
13) *Searching a rule that matches to the sequence of states.*
14) *If a rule is not found go to step 16.*
15) *Adding recommendations to the learner's individual list and outputting them on display.*
16) *If the project is not finished go to step 5*
17) *Output.*

Traditional adaptive learning methods are based on a modular principle. Each module contains a certain number of elements that have a learning element adapted to the specific level of the trained engineer's knowledge. Formula for efficiency evaluation is given as:

$$\frac{L_s}{L_p} = \frac{element_{modul} \times element_{submodul}}{element_{scenariy} + element_{cadmodel}} \qquad (1)$$

where $element_{modul}$ is the number of modules, $element_{submodul}$ is the number of learning elements in a module, $element_{scenariy}$ is the number of scenario's learning elements, $element_{cadmodel}$ is the number of learning elements of a subject domain model. There is a graph of the application efficiency of a new adaptive planning and control method of a learning path in Fig. 3. The author's method gain, lettered *P*, is 2 times compared to *S*.

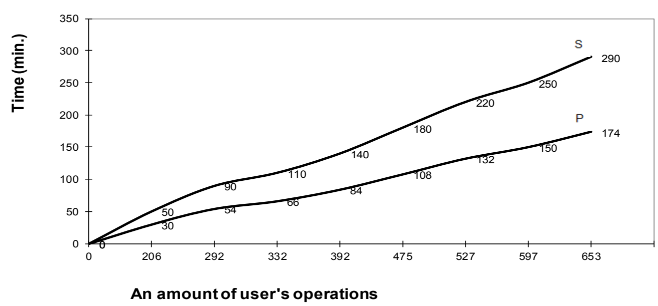

Fig. 3. Dependence training time on the number of concepts and operations: *S* is known adaptative methods, *P* is the author's adaptative method.

Fig. 4 shows the screen form of the proposed simulator. The effectiveness of the development of the approach, methods and models is confirmed by the positive results of the training of employees with the help of the author's system. The experiment was conducted as a proof of the reduction of the training period. In the experiment, was attended by 210 employees, among whom 80 radio engineering, 70 controllers, 60 installers of electronic equipment and devices. In this experiment, the system has reduced the duration of training from 3 months to one.

Fig. 4. Virtual workplace radio installer electronic equipment and devices.

ACKNOWLEDGMENT

The reported study was funded by RFBR and Government of Ulyanovsk Region according to the research project No. 16-47-732152. The research is supported by a grant from the Ministry of Education and Science of the Russian Federation, project No. 2.1615.2017/4.6.

CONCLUSIONS AND FUTURE WORKS

The system of virtual simulators is developed by the Institute of Distant and Further Education of Ulyanovsk State Technical University (Russia) on basis of the Moodle platform. ILE is embedded in the learning process of the University. The experiments over the past three years for groups of 150 students and 60 employees have been shown good results: the period of studying subjects of 72 hours have been decreased on average by 20%, while the quality of education, measured as compliance with the required list of competences have been increased by 22%. In modern conditions of the collaborative task decisions, in particular of designing the complex technical objects, developing the technological processes, etc. the important issue within studied by this research work theme is group training. The main point of it is role training and relationship among learners that solve their challenges and are gathered by a common goal. As for virtualization today, the authors develop virtual workplaces for radio-wiremen, regulators and fitters. In the future authors will plan to integrate workplaces into a virtual manufacturing department, and then into the workshops.

REFERENCES

[1] J. L. FEINSTEIN FREDERICK SIEMS, “EDAAS: an expert system at the US Environmental Protection Agency for avoiding disclosure of confidential business information,” Expert Systems, vol. 2, no. 2, pp. 72–85, Apr. 1985, doi: 10.1111/j.1468-0394.1985.tb00451.x.

[2] M. Valtorta, B. Smith, and D. Loveland, “The graduate course advisor: a multi-phase rule-based expert system,” Proc. IEEE Workshop on Principles of Knowledge-Based Systems, IEEE Computer Society. IEEE Computer Society Press, 1109 Spring Street, Silver Spring Md, 1984.

[3] O.A. Shabalina, “The learning process model and its interpretation in educational computer game,” Vestnik SSTU [University magazine], 2(70), pp. 158-167, 2013. (in Russian).

[4] V. B. Kudryavtsev, P. A. Aliseychik, K. Vashic, “Modeling of the learning process,” Fundamental’naya i prikladnaya matematika [Fundamental and applied mathematics], vol. 15, no. 05, pp. 111-169, 2009. (in Russian).

[5] E.V. Kasyanov, “WAPE is an adaptive system support distance learning of programming,” Proc. The International conference. «Computational and informational technologies in science and education» on Proceedings, Pavlodar, vol. 1, pp. 606-615, 2006. (in Russian).

[6] A. G. Dorrer, “Modelling of interactive adaptive learning course,” Sovremennie problemi nauki I obrazovaniya [Modern problems of science and education], no. 5, pp. 52-59, 2007. (in Russian).

[7] M.J. Huang, H.S. Huang, M.Y. Chen, “Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach,” Expert Systems with Applications, vol. 33, pp. 551-564, 2007.

[8] M. K. Khribi, M. Jemni, & O. Nasraoui, “Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval,” Educational Technology & Society, vol. 12 (4), pp. 30-42, 2009.

[9] V. Esichaikul, S. Lamnoi, and C. Bechter, “Student Modelling in Adaptive E-Learning Systems. Knowledge Management & E-Learning,” An International Journal, vol. 3, pp. 342-355, 2011.

[10] Y. E. A. Mustafa, S. M. Sharif, “An approach to adaptive e-learning hypermedia system based on learning styles (AEHS-LS): Implementation and evaluation,” International Jour-nal of Library and Information Science, vol. 3, no. 1, pp. 15-28, 2011.

[11] L. Mahnane, M. Tayeb Laskri, and P. Trigano, “A Model of Adaptive e-learning Hypermedia System based on Thinking and Learning Styles,”.IJMUE, vol. 8, no. 3, pp. 339-350, 2013.

[12] E. Ploum, “Authoring of adaptation in the GRAPPLE project,”. Master Thesis, Eindhoven University of Technology, 2009.

[13] P. De Bra, D. Smits, K. van der Sluijs, A.I. Cristea and M. Hendrix, “GRAPPLE: Personalization and adaptation in learning management systems,” World Conf. edmedia on Proceedings, pp. 3029-3038, 2010.

[14] A.N. Afanasyev, N.N. Voit, “Intelligent learning environments for corporations,” The 9th IEEE International conference on Application of Information and Communication Technologies: AICT on Proceedings, pp. 107-112, 2015.

[15] A. Afanasyev, N. Voit, D. Kanev, T. Afanaseva, “Organization, development and implementation of intelligent learning environments,” 10th International Technology, Education and Development Conference: INTED-2016 on Proceedings, pp. 2232-2242, 2016, https://library.iated.org/view/AFANASYEV2016ORG, last accessed 2018/05/14.

[16] A. Afanasyev, N. Voit, “Intelligent Agent System to Analysis Manufacturing Process Models,” Proceedings of the First International Scientific Conference «Intelligent Information Technologies for Industry» (IITI’16), vol. 451, pp. 395-403, 2016, doi: 10.1007/978-3-319-33816-3_39.

[17] A.N. Afanasyev, N.N. Voit, D.S. Kanev, “Development of intelligent learning system based on the ontological approach,” 10th IEEE International Conference on Application of Information and Communication Technologies (AICT) on Proceedings, pp. 690-694, 2016, doi: 10.1109/ICAICT.2016.7991794.

[18] A. Afanasyev, N. Voit, D. Kanev, “Intelligent and virtual training environment,” International conference on Fuzzy Logic and Intelligent Technologies in Nuclear Science - FLINS2016 on Proceedings, pp. 246-251, 2016, doi: 10.1142/9789813146976_0041.

[19] S. Sosnovsky, P. Brusilovsky, “Evaluation of topic-based adaptation and student modeling in QuizGuide,” User Modeling and User-Adapted Interaction, vol. 25, no. 4., pp. 371-424, 2015.

[20] P. Couto, M. Fernandes, C. Martins, and L. Faria, “PCMAT, an adaptive learning platform,” ICIPT 2013: 8th International Conference on Information Processing, Management and Intelligent Information Technology (ICIPM, ICIIP). Advanced Institute of Convergence Information Technology (AICIT), 2013.

[21] M. Deshpande, G. Karypis, “Item-based top-n recommendation algorithms,” ACM Transactions on Information Systems (TOIS), vol. 22, no. 1, pp. 143-177, 2004.

[22] P. Marschik, “Evaluation of collaborative filtering algorithms,” Doktorska disertacija, 2010.

[23] M. Kula, “Metadata Embeddings for User and Item Cold-start Recommendations,” the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), 2015.

[24] S. Reiterer, A. Felfernig, M. Jeran, M. Stettinger, M. Wundara, W. Eixelsberger, “A Wiki-based Environment for Constraint-based Recommender Systems Applied in the E-Government Domain,” UMAP Workshops, 2015.

[25] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, “Recommender systems: an introduction,” Cambridge University Press, 2010.