



HAL
open science

Graph-Based Method for the Interpretation of User Activities in Serious Games

Iwona Grabska-Gradzińska, Jan K. Argasiński

► **To cite this version:**

Iwona Grabska-Gradzińska, Jan K. Argasiński. Graph-Based Method for the Interpretation of User Activities in Serious Games. 18th IFIP Conference on Human-Computer Interaction (INTERACT), Aug 2021, Bari, Italy. pp.87-96, 10.1007/978-3-030-85613-7_6 . hal-04292348

HAL Id: hal-04292348

<https://inria.hal.science/hal-04292348v1>

Submitted on 17 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Graph-based Method for the Interpretation of User Activities in Serious Games

Iwona Grabska-Gradzińska¹[0000–0002–5799–5438]
and Jan K. Argasiński¹[0000–0002–2992–718X]

Institute of Applied Computer Science; Faculty of Physics, Astronomy and Applied
Computer Science; Jagiellonian University in Krakow, Poland.

{iwona.grabska, jan.argasinski}@uj.edu.pl

Abstract. Computer-based training systems are of growing importance nowadays. Demand for remote skill and knowledge validating solutions is higher than ever before. Serious games provide safe and replayable conditions for teaching and practice. The main issue with education founded on serious games paradigm is the evaluation of the players. Virtual environments allow for various activities that are hard to grasp and quantify automatically. Gameplay log files are usually filled with an excess of trivial activities that don't provide vital information about user's skills or knowledge in the relevant domain. In this paper, the authors propose a novel method for classification and interpretation of user activities in computer-based training systems. The innovation of the presented idea lies in the combination of layered graphs data structures, and Bayesian inference about the user's skills and knowledge.

Keywords: Layered graphs · serious games · training evaluation · educational assessment.

1 Introduction: Player Evaluation in Serious Games

Serious games are usually defined after the classic book of Clark C. Abt as having “an explicit and carefully thought-out educational purpose and (...) not intended to be played primarily for amusement” [1]. Most of the modern computer based serious games are training or educational applications [17]. The main problem with regard to educational games is the assessment. Like in any other scholarly context the information about learners' progress is crucial for evaluating the entire projects. Fortunately, the educational progress of users can be viewed in part in terms of the player's progress in the game. The games have “naturally” built-in progress tracking systems such as points, achievements, character levels etc. They also allow detailed logging of user activity.

Nowadays – not only due to the global crisis caused by the COVID-19 pandemic started in 2020 – there is a growing interest in remote training. More and more companies, training centers and schools are forced to work on-line. Regardless of the future epidemiological situation, this trend will in all likelihood increase. From the point of view of designers and creators of serious games, this

means the need to introduce more and more sophisticated methods of evaluation and the need to automate this process. Generally, procedural assessment based on mechanics is an idea to strengthen one of the most important added values in serious games – the potential variety of scenarios that go hand in hand with the repeatability of the assessment. Procedural story and/or world generation is widely used in entertainment, education and reasoning [8–13].

At this point, attention should also be paid to the distinction between the assessment and evaluation. As authors of [2] indicate – assessment is a formative, diagnostic and process-oriented activity, while evaluation is more often summative, judgmental and product oriented. Nevertheless, the common areas of these two ways of looking at educational progress include the fact that they are data-intensive, utilizing measures, and user-oriented.

In the case of computer games, the players’ behaviors observed in the game are in fact activations of mechanics implemented by the designers. We define mechanics after M. Sicart as “methods invoked by agents to interact with the game states” [4]. The player – within the system – cannot do anything that would not be mechanics (although there is a problem of *subversive play* – exploiting *bugs* and *glitches* [5]). This means that for the purposes of evaluating educational progress in serious games, we can treat mechanics as observables – from their applications and parameters (timing, sequence, speed, context) we can obtain some information about the knowledge, skills, abilities or attitudes of players. Such a framework – combining the ECD (*Evidence Centered Design*) paradigm [6] with the MDA (*Mechanics, Dynamics, Aesthetics*) model [7] is proposed by authors of [4]. In our proposal, mechanics applied in the contexts will constitute the basis for updating the values of evaluation variables.

The spheres of designer’s responsibility are grouped into two separate entities here simply called *the game* and *the educational robot*:

However, the domain content and learning theories, the pedagogy, and the learning profile have a common context representing serious sides which can be grouped into one entity called an educational robot; on the other hand, all game aspects will be represented in one entity called a game. The educational robot and the game should exchange data or information; the game knows what the learner/player is doing, and by communicating relevant information (learner traces) to the educational robot, the last one knows what the player has done, and by analyzing learner’s traces a decision is communicated back to the game (pedagogical order) to be applied such as repeating the same level (...) [15]

In our, presented below, concept – instead of the *communication protocol* between the layers of the system, we propose that the competences of the *educational robot* should be implemented by a properly structured game system, and that the progress in the student’s profile is tracked with the use of variables represented in the graph system, which also describes the player’s possible mechanics with their context. In this way, the system architecture is significantly reduced while maintaining full designer’s control over the application and with

the possibility of procedural content generation, e.g. as repetitions with variants required for accurate testing and predictions [16].

2 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) is a well established algorithm for modeling learners educational progress [32]. BKT is a proven solution for the estimation of student proficiency and the effectiveness of the educational process [31]. Due to its probabilistic construction, it is suitable for prediction and use in [30] recommending systems. A natural development of this solution is its use in intelligent systems, [29] also those based on deep learning [28].

An important element of the solution presented in this article is the use of before-mentioned idea. According to the Bayes Theorem (1), the likelihood of event A under the condition of B: $p(A/B)$ is equal to likelihood of event B under A: $p(B/A)$ multiplied by probability of observing event A: $p(A)$ under condition of B: $p(B)$ [23].

$$p(A/B) = \frac{p(B/A)p(A)}{p(B)} \quad (1)$$

Therefore marginal probability of B: $p(B)$ is equal to sum of joint probability $p(A,B)$ over all possible values of the other variables (in this case a represents particular value of A, (2):

$$p(B) = \sum_{a \in A} p(B, a) \quad (2)$$

The proposed method of constructing evaluation in serious games is based on the idea of Bayesian Knowledge Tracing [19–21] often used in Intelligent Tutoring Systems [22]. These systems can be described as based on the following modules: Student, Tutor, Domain and Interface – just like in the case of before-mentioned educational *robot architecture*, created for the needs of thinking about serious games. For us, the most interesting activity is to model the Student module and the context in which he/she is thrown (game world, *diegesis*). According to [22], the Student module can be considered as a quantitative representation of the recorded behavior of the application user. There are many ways to express the actions performed in the context of a certain domain. We propose a *game*-related view through the prism of mechanics, described by the structure of layered graphs.

In the light of classical Bayesian Knowledge Tracing theory (as described e.g. in [20]), the key parameters are:

- $p(L_0)$ – probability that the student u has the skill k from an external source (it can be estimated, for example, using the initial knowledge test);
- $p(T)$ – probability of transfer of knowledge about the student’s state from *unknown* to *known* after the occurrence of a situation enabling its application;

- $p(G)$ – probability of correct execution of the action despite the lack of the skill (“by accident”, “guess”);
- $p(S)$ – probability of making a mistake despite having the appropriate skill (“accidental mistake”, “slip”).

The initial situation is illustrated by equation (3). Equations (4) and (5) are used to calculate $p(L)$ in the following steps in the case of observed (*obs*) correctly (4) and incorrectly (5) performed operations. At the end (6) and (7) are used to update the probability that student u will apply k correctly when given the opportunity.

$$p(L_1)_u^k = p(L_0)^k \quad (3)$$

$$p(L_{t+1}|obs = correct)_u^k = \frac{p(L_t)_u^k \cdot (1 - p(S)^k)}{p(L_t)_u^k \cdot (1 - p(S)^k) + (1 - p(L_t)_u^k) \cdot p(G)^k} \quad (4)$$

$$p(L_{t+1}|obs = wrong)_u^k = \frac{p(L_t)_u^k \cdot p(S)^k}{p(L_t)_u^k \cdot p(S)^k + (1 - p(L_t)_u^k) \cdot (1 - p(G)^k)} \quad (5)$$

$$p(L_{t+1})_u^k = p(L_{t+1}/obs)_u^k + (1 - p(L_{t+1}/obs)_u^k) \cdot p(T)^k \quad (6)$$

$$p(C_{t+1})_u^k = p(L_t)_u^k \cdot (1 - p(S)^k) + (1 - p(L_t)_u^k) \cdot p(G)^k \quad (7)$$

The dynamics of the described situation is presented in the Fig. 1 below where K stands for the *knowledge* node, and Q for the *quiz* node (or – in our case – *quest* node).

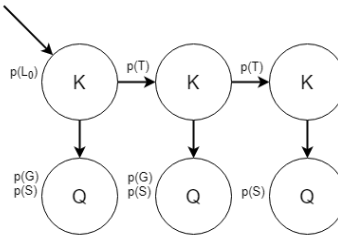


Fig. 1. The Standard Bayesian Knowledge Tracing model (after [24])

3 Layered Graphs in the Description of Gameplay

Players' behaviours and decisions are closely connected with game mechanics and interactive game elements available at the moment of decision. Both for player actions analysis and game design verification, it is useful to represent the gameplay in the formal way. Graphs are very efficient and flexible mechanisms for modelling the system dependencies and user-evoked changes [25]. Moreover, layered graphs are very useful for describing particular points within the story [26]. It is due to the fact that the states of the world elements are divided into separate layers, and graph edges correspond to different dependencies of the system according to the layer type of connected nodes. The formal definition is described in [27].

In presented system the model of a game consists of two separate graphs: a graph model of the world state and a graph model of the gameplay. The graph model of the game world (the World State Graph) is layered and hierarchical. The four layers: layer of locations, layer of characters, layer of items or layer of narrative elements represent objects of the given world. Connections between objects are represented by edges, e.g. the edge between item and character means ownership, the edge between location and location means passage. Node attributes represent object properties. Most player's decisions are influenced by his surroundings: location, elements placed there, as well as his own belongings and attributes. That is why a specific graph structure is assigned – a sheaf subgraph. This structure is most often used in graph operations, called productions. The schematic view of world graph with the division into layers is presented in the Fig. 2.

3.1 User decision as graph production

The events which change the game world, evoked by the player as well as external factors, can be easily represented as graph rules, also known as productions. A production is a pair of graphs called the left-hand side and the right-hand side of the production. The player action is possible if there is in the world graph subgraph identical with the left-side of the production and is implemented as replacing the right side of the production and its embedment.

The structure of the vertex information and relations between vertices in JSON format is shown below.

```
"L_Side": { "Locations": [ {"Name": "Accident scene",
"Attributes": { }, "Characters":
[ { "Name": "Hero", "Items": [ { "Id":"Phone"} ] },
{ "Name": "Victim","Attributes": { "Conscious": 0},
"Characters": [ ], "Items": [ { "Id":"Id card"} ] },
"Items": [ ], "Connections": [ { "Destination":
"Emergency service","Attributes": { "Type":"Phone call" } } ] }
{ "Name": "Emergency service", "Connections":
[ { "Destination": "Accident scene"} ] } ] }
```

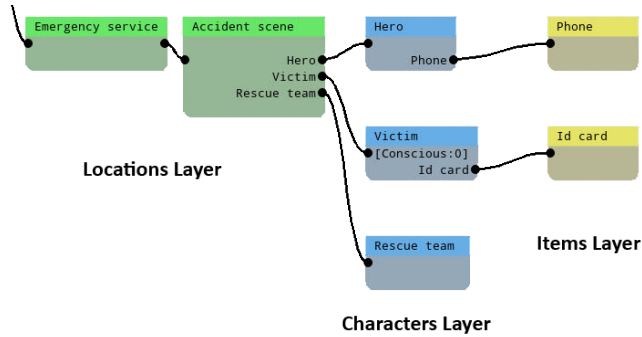


Fig. 2. Implementation of World State Graph in the Godot game engine.

Application of the production to a given World State Graph causes replacement of the nodes of the left-hand side by the nodes of the right-hand side and their embedment in the World State Graph. The first step of applying the production is to match, i.e., to fit the left-hand side to the World State Graph. A successful fit means that there is a subgraph in the WSG which is equal to the left-hand side graph.

3.2 Gameplay graph

The sequence of players action is one of the most important elements of player evaluation. Possible sequences of available productions are used to describe the designed scenarios. The initial state of the world graph allows for these sequences to be generated by player or by testers (manually or semi-automatically, using graph matching functions). The graph of the all possible sequences give us the model of gameplay.

4 Applying graph-based method for the interpretation of user activities in serious games

The example shown below concerns the application supporting the CPR training for OSH (Occupational Safety and Health) courses for employees and helps to evaluate the user's competences after the training. The system is implemented in Godot engine 3.2.3 and graph information is stored in JSON format.

The application of the proposed idea in which we combine the concept of using layer graphs to describe the course of a serious game with the Bayesian Knowledge Tracking method is presented below. A fragment of a serious game is described, in which the protagonist (*the player*) has to undertake a rescue action, taking into account the knowledge and skills related to performing CPR (*cardiopulmonary resuscitation*) (Fig. 3). The following graph productions describe possible actions and their results. In particular, the red node denotes the death

of the injured person and the green one – the success of the rescue operation. The scene (*quest*) starts at the gray node marked with the number (1).

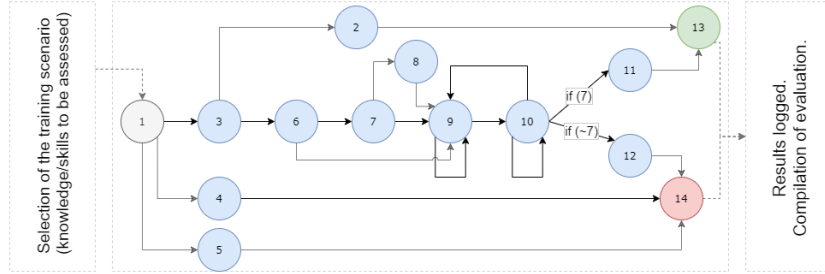


Fig. 3. Gameplay graph describing simple CPR scenario in educational serious game

Description of the nodes: (1) The accident happens; (2) Another actor takes over initiative; (3) Player checks victim’s breath; (4) Player engages in idle activity (eg. plays games on their cellphone, ignoring victim); (5) Player falls into despair (consequently ceasing the aid); (6) Player checks victim’s pulse; (7) Player calls an ambulance by the phone; (8) Player takes instructions from the dispatch; (9) Player delivers rescue breath; (10) Push hard, push fast...; (11) Arrival of the medical assistance; (12) Loss of strength; (13) Patient is saved; (14) Patient dies.

In the challenge presented above, the following production series are possible:

- (a) 1-3-2-13 (success)
- (b) 1-3-6-7-9-10-11-13 (success)
- (c) 1-3-6-7-8-9-10-11-13 (success)
- (d) 1-3-6-9-10-12-14 (defeat)
- (e) 1-4-14 (defeat)
- (f) 1-5-14 (defeat)

It’s worth noting that the scenario 1-3-6-9-10-11-13 is impossible. If an ambulance is not called (7), help will not arrive in time and the action will end in failure (14).

Now we can apply Bayesian Knowledge Tracing methods to provide means for interpreting various outcomes. For the purposes of this example, we will not build a matrix with weights assigned to particular nodes since it would require the expertise of a domain specialist and conducting tests. We will limit ourselves to showing some dependencies in terms of parameters previously defined as interesting for us. Thus we can conclude that:

- in the case of (a) – $p(L)$ remains unknown; $p(T)$ is very low; $p(G)$ is unknown; $p(S)$ is unknown;
- in the case of (b) – $p(L)$ is the greatest among all scenarios; $p(T)$ is highest among all scenarios; $p(G)$ is very low due to length of the chain of productions; $p(S)$ is not applicable;

- in the case of (c) – $p(L)$ is lower than in (b) but higher than in other cases; $p(T)$ is lower than in (b) but higher than in other cases; $p(G)$ is very low due to length of the chain of productions; $p(S)$ is not applicable;
- in the case of (d) – $p(L)$ is highest among all failed scenarios; $p(T)$ is the highest among all failed scenarios; $p(S)$ is very low due to length of the chain of productions; $p(G)$ is high due to only one mistake made that lead to failure (skipping (7));
- in the case of (e) – $p(L)$ is unknown (attitude is low); $p(T)$ is very low; $p(G)$ is practically not applicable; $p(S)$ is very much not applicable;
- in the case of (f) – $p(L)$ is unknown (ability is low); $p(T)$ is very low; $p(G)$ is practically not applicable; $p(S)$ is not applicable;

Writing the correct numerical values (multipliers) to the above general statements will automate the process of assessing the probability $p(C)$ – that a given skill will be correctly applied in the next test case. Of course, it is essential to test knowledge, skills, abilities and attitudes many times in different contexts. But the very construction of a serious game based on a generator in the form of a layered graph makes it easier and enables it. In the section 3.1 the example of a JSON code was presented to describe a test situation followed by the example of implementation in a game engine (Fig. 2). Innovation of the presented idea is also in the ability to represent user’s contextual information in the form of gameplay reports generated after the simulation. Graphs and probabilities provide an effective generalized language of communicating the user’s skills, knowledge and abilities in a varied context. In proposed solution input block visible on the left side of the Fig.3 represents the moment of selection of the context of simulation; output block represents the process of writing contextualized data into a generalized database from which the further compilation of the results is possible.

5 Conclusion

The article above presents a way to describe the structure of a serious game using layer graphs. Furthermore, it describes a method proposed for combining mechanics with a statistical measure, which is used to evaluate the educational progress of the player. Of course, there are many possible vectors for further development of the concept above – one can mention such issues as the automation of metrics construction, various possible strategies of the user’s model fitting – including predictions in the scope of the application’s operation. There is also the question of whether is it possible to estimate $p(S)$ – *slip*, and $p(G)$ – *guess* using the context [19]. The next step of the presented system development is to propose the same narrative structure but with the different production sequences to evaluate different fields of the skills and knowledge for different types of users.

As shown above, the graph based system of representation of important game objects and events can be effectively and profitably used to support computer based educational evaluation tools.

References

1. Abt, C. C.: *Serious Games*. Viking Press (1970).
2. Daoudi, I., Tranvouez, E., Chebil, R., Espinasse, B., Chaari, W. L.: Learners' Assessment and Evaluation in Serious Games: Approaches and Techniques Review. *Information Systems for Crisis Response and Management in Mediterranean Countries*, 147-153. doi:10.1007/978-3-319-67633-3_12 (2017).
3. Serrano-Laguna, Á., Manero, B., Freire, M., Fernández-Manjón, B.: A methodology for assessing the effectiveness of serious games and for inferring player learning outcomes. *Multimedia Tools and Applications*, 77(2), 2849-2871. doi:10.1007/s11042-017-4467-6 (2017).
4. Sicart, M.: Defining Game Mechanics. *Game Studies. The International Journal of Computer Game Research*. 8. (2008).
5. Aarseth, E.: I Fought the Law: Transgressive Play and the Implied Player. *From Literature to Cultural Literacy*, 180–188. doi:10.1057/9781137429704_13 (2014).
6. Mislevy, R. J., Almond, R. G., Lukas J. F.: *A brief introduction to evidence-centered design (cse report 632)*, CA: Center for Research on Evaluation, Standards, and Student Testing (2004).
7. Winn B.: *The Design, Play, and Experience Framework*, Hershey (2009).
8. Compton K., Filstrup B., Mateas M.: Tracery: Approachable story grammar authoring for casual users. In *Papers from Seventh Intelligent Narrative Technologies Workshop (INT 07)*, pp. 63–67 (2014).
9. Compton C., Kybartas B., and Mateas M.: Tracery: an author-focused generative text tool. In *Interactive Storytelling (ICIDS 2015)*. *Lecture Notes in Computer Science*, Vol 9445. Springer, Cham. https://doi.org/10.1007/978-3-319-27036-4_14 (2015).
10. Grinblat J., Bucklew C. B.: Subverting historical cause & effect: generation of mythic biographies in Caves of Qud. In *Proceedings of the 12th International Conference on the Foundations of Digital Games (FDG '17)*. Association for Computing Machinery, New York, NY, USA, Article 76, 1–7, <https://doi.org/10.1145/3102071.3110574> (2017).
11. Jason Grinblat and Bucklew C. B.: Warm Rocks for Cold Lizards: Generating Meaningful Quests in Caves of Qud. In: *Experimental AI in Games Workshop (AIIDE 2020)*.
12. Adams T.: Secret identities in dwarf fortress. In: *Proceedings Experimental AI in Games (AIIDE 2017)*, 22–25 (2017).
13. Tarn Adams T.: Simulation principles from Dwarf Fortress. In: *Game AI Pro 2: Collected Wisdom of Game AI Professionals*, pages 519–521. CRC Press (2015).
14. Argasiński, J. K., Wegrzyn, P.: Affective patterns in serious games. *Future Generation Computer Systems*. doi:10.1016/j.future.2018.06.013 (2018).
15. Mestadi, W., Nafil, K., Touahni, R., Messoussi, R.: An Assessment of Serious Games Technology: Toward an Architecture for Serious Games Design. *International Journal of Computer Games Technology*, 2018, 1-16. doi:10.1155/2018/9834565 (2018).
16. Alonso-Fernández, C., Martínez-Ortiz, I., Caballero, R., Freire, M., Fernández-Manjón, B.: Predicting students' knowledge after playing a serious game based on learning analytics data: A case study. *Journal of Computer Assisted Learning*. doi:10.1111/jcal.12405 (2019).
17. Djaouti, D., Alvarez, J., Jessel, J.-P., Rampnoux, O.: *Origins of Serious Games*. 10.1007/978-1-4471-2161-9_3 (2011).

18. Orun, A., Seker, H.: Development of a computer game-based framework for cognitive behaviour identification by using Bayesian inference methods. *Computers in Human Behavior*, 28(4), 1332–1341. doi:10.1016/j.chb.2012.02.017 (2012).
19. Pelánek, R.: Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, 27(3-5), 313–350. doi:10.1007/s11257-017-9193-2 (2017).
20. Yudelson, M. V., Koedinger, K. R., Gordon, G. J.: Individualized Bayesian Knowledge Tracing Models. *Artificial Intelligence in Education*, 171–180. doi:10.1007/978-3-642-39112-5_18 (2013).
21. Baker, R. S. J. d., Corbett, A. T., Alevan, V.: More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. *Lecture Notes in Computer Science*, 406–415. doi:10.1007/978-3-540-69132-7_44 (2008).
22. Manrique, R., López, C., Leon E.: Student Modeling via Bayesian Knowledge Tracing: a case study. 10.13140/RG.2.1.3646.4167 (2014).
23. Englebienne, G.: Bayesian Methods for the Analysis of Human Behaviour. *Computer Analysis of Human Behavior*, 3–20, doi:10.1007/978-0-85729-994-9_1 (2011).
24. Pardos, Z., Heffernan, N.: KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. 243-254. 10.1007/978-3-642-22362-4_21 (1970).
25. Ślusarczyk, G.: Graph-based representation of design properties in creating building floorplans. *Computer Aided Design* 95, (February 2018), 24–39. <https://doi.org/10.1016/j.cad.2017.09.004> (2018).
26. Grabska-Gradzińska I., Nowak L., Palacz W., Grabska E.: Towards Automatic Generation of Storyline Aided by Collaborative Creative Design. In *Cooperative Design, Visualization, and Engineering (CDVE 2020)*. *Lecture Notes in Computer Science*, Vol. 12341. Springer, Cham. https://doi.org/10.1007/978-3-030-60816-3_6 (2020).
27. Grabska-Gradzińska I., Nowak L., Palacz W., Grabska E.: Automatic Story Generation Based on Graph Model Using Godot Engine. In *Artificial Intelligence and Soft Computing (ICAISC 2020)*. *Lecture Notes in Computer Science*, Vol. 12415. Springer, Cham. https://doi.org/10.1007/978-3-030-61401-0_37 (2020).
28. Gervet, T., Koedinger, K., Schneider, J., Mitchell, T.: When is Deep Learning the Best Approach to Knowledge Tracing?. *Journal of Educational Data Mining*, 12(3), 31-54. <https://doi.org/10.5281/zenodo.414361> (2020).
29. Minn, S.: BKT-LSTM: Efficient Student Modeling for knowledge tracing and student performance prediction. arXiv preprint arXiv:2012.12218 (2020).
30. Zhao, J., Bhatt, S., Thille, C., Zimmaro, D., Gattani, N.: Interpretable Personalized Knowledge Tracing and Next Learning Activity Recommendation. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pp. 325-328, (2020).
31. Corbett, A. T., Anderson, J. R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*. 4 (4): 253–278 (1995).
32. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: Individualized bayesian knowledge tracing models. *Artificial Intelligence in Education* (2013).