Abstract— In this paper we have described game-based learning example that we’ve already implemented. Capitalizing on this model and novel equation for estimating a student’s knowledge level has been also derived. Parameters in derived equation are then optimized by minimizing the root mean square error between classical test results of students and result of estimation model calculated by equation. Knowledge level gain, acquired by using the application is estimated and presented in the function of equation parameters as another performance measure. Sensitivity of obtained performance measures has also been observed. Presented results show the possibility of applying presented method in various game-based learning applications for knowledge level performances determination.

I. INTRODUCTION

Game-based learning (GBL) refers to different kinds of software applications that use games for learning or educational purposes [1]. The interestingness and extensiveness have conditioned research of the educational games applicability models in the process of education especially for those children who have lost interest in certain school courses.

Game-based learning promises to be a new successful approach in conducting university education. Motivating today’s students with traditional teaching methods such as lectures and written materials proves to be a difficult task and consequently, universities are searching for a new method in the changing context of education. There seems to be a growing awareness of the research potential that lies in gaming as a new promising form of interactive content [Pivec, 07]. To a great extent educational games can be referred to as computer science disciplines. Authors [Shabalina, 08] have implemented the educational game concept in the learning game for C# programming language. Their system is based on the common game engine architecture, but it is extended for the use in educational games and consists of two high-level subsystems: a game engine and a learning engine. Course of Programming is found as a difficult and boring subject; thus, learning with games seems to develop students’ motivation on the course. Authors [Roslina, 11] describe the perceptions of students in a Malaysian university (UTM) towards using educational games for self-learning of Programming Introductory course.

A learner model, also called a student model, refers to the model constructed from observation of interaction between a learner and a learning system or instructional environment. A student model must contain important information about the user such as: domain knowledge, learning performance, interests, preference, goal, tasks, background, personal traits (learning style, aptitude...), environment (context of work) and other useful features [Nguyen, 08]. A student model is one of the fundamental components of modern intelligent learning environment, and a lot of research has been devoted to creating student models for various types of computer based support [2]. There is increasing research in learning student models from data [3,4], but most of this research is focused on student models for more traditional Intelligent Tutoring Systems (ITS). Measuring student knowledge in a particular game level is therefore very important. To clarify it – the game is designed in a way to estimate knowledge of a student for current topic (or the level of the game) and to let him/her continue with the next topic if the knowledge for that topic is sufficient. Further behavior of the educational game (go to next level or back to current level) depends on a good diagnosis of student knowledge in the game level. In this paper we present: Firstly, existing learning model that we have tried to implement in our case. Secondly, creation of equation model we have thought would fit as correct model with constant weighting coefficients and finally, creation of equation model with variable weighting coefficients and procedure for determination of coefficients’ values by using simple algorithm.

II. SUBJECT AND METHODS

The student knowledge level in an adaptive hypermedia application [5] is measured using the answers to the questions previously presented to the student. The decision whether a student has solved the task presented in our proposed model is made on the basis of a formula that, besides correct answers includes two additional parameters (time and the number of used Help options[6]).

In the ITS approach [7], user model content variables are used for keeping records on user interaction with the ITS and for adjusting the content presentation to the user profile. These learning style variables are part of a Bayesian Network - BN for making conclusions about the student. There is a list of variables for each topic: spent time, topic deepness level, wrong answers and correct
answers. The variables relevant for deciding on student knowledge in our educational game *ArhiteCOMP* also include spent time and correct answers. However, the spent time needed for solving the tasks in our approach is divided into: 1) time used to read contest of Help window and 2) time needed for giving the answers when the Help windows were not used.

The idea of the Neuro-Fuzzy Reasoner (NFR) system [8] was the initial inspiration for the model for student knowledge diagnosis in our game-based learning system. The NRF system is relatively simple, supports creation of high-level pedagogical strategies, and can be easily adapted to individual teacher’s preferences. The NFR model for student classification is based on test results and the time needed to complete the test. Modification of the NRF system parameters is made by adding a new parameter – time needed for reading the contents of help window.

We have created the computer application for game based learning and loaded one topic of the course "Computer Architecture and Organization 1" in it. The aim of the course topic ‘*Unary logical operations*’ is to teach students (at the first grade of School of Electrical Engineering and Computer Science Applied Studies in Belgrade) how to perform logical operations at the level of registers in the computer system, through comparing the register binary contents before and after performing the given operation. The content of the educational games is designed as a platform for learning through interactive tasks [9] whose solving is stimulated by the game content and directly facilitated acquisition of knowledge in the course mentioned.

The learning model we have used in this education game is based on the principle of operation of the NFR presented by Z. Sevarac in his work [7]. Our model has been extended with one more input variable – Help window, because this component has been used by students in the educational game to a great extent. The initial model we started with and which we used in the educational game *ArhiteCOMP* was based on the neuro-fuzzy reasoner.

Since the initial model didn’t give expected results, we applied the new model for knowledge level estimation which used coefficients as variable values. The rule for determining the percentage of knowledge applies basic arithmetical operations to input parameters of the model. The significance of the input values for final knowledge estimation is determined on the basis of empirical coefficient values, given by the teacher. The knowledge level that student possesses after playing the education game is given by the following formula:

\[
P_i(X = Mastered) = \left( a \cdot \frac{A_i}{N} - h \times \frac{H_i}{N} - t \times \frac{t_h \cdot (A_i - H_i) - t_h \cdot H_i}{T_{max}} \right) \times 100
\]  

Where:

- \(a\), \(h\) and \(t\) are coefficients that should be estimated,
- \(A_i\) - is the number of correct answers of \(i\)-th student,
- \(H_i\) - is the number of opened help windows of \(i\)-th student,
- \(t_h\) - is the average time a student needs to give an answer without using help window,
- \(t_h\) - is the average time a student needs to give an answer with using help window,
- \(N\) - is the total number of answers,
- \(T_{max}\) - is maximum duration of the game.

The values of coefficients: \(a\), \(h\) and \(t\) can have range from 0 to 1. The first part of the formula has the most significant role in calculating a student’s final knowledge level, since it uses the number of correct answers - \(A_i\). The second important part for knowledge estimation is the second part of the formula, which represents the number of used help windows during playing the game - \(H_i\). This part of the formula has the negative sign, since it decreases probability of the final knowledge level. The part of the formula which depends on a time has the least importance for knowledge calculation. When a student uses the Help window, the time needed for giving an answer increases, which results in reduction of the student’s total knowledge.

The basic terms of knowledge that student has to present on test are written in the help windows in the game. When a student starts learning with the use of the game or faces a difficulty during solving a task generated by the application, Help window serves to accelerate finding the right solution related to particular task. This means that appropriate formulation of definitions and theorems within Help window is the key moment in designing the entire application. Also there is a need to optimize the maximal duration of the game \(T_{max}\). We have selected the approach to limit \(T_{max}\) by setting \(T_{max} \geq N \times t_h\). This admission is based on assumption, that when a student does not know the answer to single question, he will use the help window for each query namely \(N \times t_h\). The inequality in relation is set, because there is some additional time provided for a student to decide will he use the help window, (if he is not sure enough that he has correct answer). If student thinks that he has the correct answer to the query, he will provide answer in shorter time \(t_h\) and estimated knowledge level will be higher according to Equation (1).

Based on teachers’ estimation, the values referring to significance of coefficients in the given formula are estimated as:

- \(a\) - 0.50,
- \(h\) - 0.30,
- \(t\) - 0.15.

Through repeated comparison of results from classical paper test and results obtained in computer game by using the estimation of the knowledge level with the new model (Equation 1 and empirical coefficients) we have come to very encouraging results. With the use of the new model we have managed to reduce the error in estimating student knowledge level by 20%. The new error value we obtain now is 29.97% with the use of the above given formula and with empirical coefficient values.

The goal of our model is to estimate the knowledge of student enough to let him or her pass the current level of game (course topic) and go to the next. Another goal is to
compare and to narrow the difference between results obtained by playing game and results obtained by classical examination methods. The results of classical examinations are graded into three groups: bad, good and excellent so we have graded the results obtained by using our model in the same manner.

The selection of optimal coefficients is performed by using algorithm for minimizing the root mean square (rms) error (shown at equation 2) between classical test results of students and result of estimation model calculated by Equation 1:

\[
\alpha_{\text{opt}}, h_{\text{opt}}, t_{\text{opt}} = \min_{(a,h,t) \in (0,1)} \sqrt{\frac{1}{N} \sum_{i=1}^{N} (R_{i,\text{classic}} - R_{i,\text{est}})^2}
\]

(2)

where \(R_{i,\text{classic}}\) is result of \(i\)-th student obtained by classical paper-based examination and \(R_{i,\text{est}}\) is result of \(i\)-th student estimated by our model.

The C++ like code of algorithm implemented is given in Figure 1. At the output of this algorithm the optimal values of \(a, h\) and \(t\) are calculated according to rms rule. The similar results may be obtained if other error criteria (like minimum averaged absolute error, maximal absolute error, etc) are selected.

The values of coefficients \(a, h\) and \(t\) calculated by equation 2 are 0.5, 0.3 and 0.15. This leads to error of 25.35%. The graphical presentation of how classical paper-based test results match with the one obtained by our estimation model is shown in Figure 2.

In our experiments we have some results of classical paper-based tests that deviate a lot from results obtained by playing computer game. For example, some students passed the classical examination test with best results (100%) but they answered less than 3 out of 8 tasks in playing game during maximum time of game duration (6 of 71), or inversely – some students achieved very bad results in classical examination tests (0%) and they answered correctly more than 5 out of 8 tasks in playing game (8 of 71).

Figure 2. The ratio of estimated values given by model of variable coefficients and the values obtained through a practical task

This is the main reason why the error of our estimation model is even lesser. Generally, the results obtained by playing the game are better than results obtained in classical paper-based tests. In Figure 3, can be seen comparison of the three models of assessment of student’s knowledge with real results.

Figure 3. Curves of the three models and practical score

III. RESULTS

It has been said, that some students, that achieved very bad results in classical examination tests (0%), have answered correctly more than 5 out of 8 tasks in playing game (8 of 71). This could be explained observing properties of Equation (1). By using help menu, even when incorrect answers are provided, a student upgrades his level of knowledge. Knowing what incorrect answer is, a student gets an information what might be correct answer to the query.

For the case when student is using \(H_i\) help windows and provides all incorrect answers, \(A_i = 0\), the level of knowledge that student possesses after playing the education game is not zero, since he knows what are incorrect answers to the queries. Concerning this he can now direct his way of thinking and conclude what might be correct answers. So expected level of knowledge which student possesses is now measured as:

\[
P(X)_{A=0} = H_i \left( \frac{t + t_a}{t_{\text{max}}} - \frac{1}{N} h \right)
\]

(4)

For the case when a student is using help windows and
provides all correct answers, \( A_i = H_i \). Then the application fully meets its goal and expected level of knowledge is measured as:

\[
P_i(X) = H_i \left( \frac{a-h}{N} + \frac{t_a}{T_{\text{max}}} \right)
\]

Now we will introduce auxiliary performance measure - the knowledge level gain (KLG). KLG can be acquired by using the application, and is estimated as:

\[
P_i(X) - P_i(X) \approx H_i \left( \frac{1}{N} - \frac{t_a}{T_{\text{max}}} \right)
\]

It can be seen from Equation (6) that KLG, obtained by using help, increases for the higher values of parameter \( a \), and decreases when parameter \( t \) grows.

Selection of optimal coefficient values, performed by using algorithm for minimizing the rms between classical test results of students and results of estimation model calculated by Equation (1), meets the presumption of Equation (6) that parameter \( a \) should have higher value than parameter \( t \).

Let us now discuss the sensitivity of Equation (1) over parameters \( a \), \( h \) and \( t \). Denoting sensitivities as \( \delta_a \), \( \delta_h \), and \( \delta_t \), respectfully, it can be written:

\[
\begin{align*}
\delta_i &= \frac{a}{N} - \frac{t_a}{T_{\text{max}}} \\
\delta_i &= -\frac{\partial P}{\partial h} = -\frac{H}{N} \\
\delta_i &= \frac{\partial P}{\partial t} = -\frac{t_a}{T_{\text{max}} - t_a (A_i - H_i)}
\end{align*}
\]

Since \( H_i \leq A_i \), it is clear that \( \delta_h \leq \delta_a \). Also, \( \delta_t \) can be written as the linear combination of two sensitivities above:

\[
\begin{align*}
\delta_t &= \delta_a \delta_a + \delta_h \delta_h \\
\delta_t &= \left( t_a - \frac{t_a}{T_{\text{max}}} \right) \delta_h \frac{N}{T_{\text{max}}}
\end{align*}
\]

Concerning (7) and (8) it is clear that sensitivity of Equation (1) is highest over parameter \( a \) and, that parameter \( a \) should not be in the area close to its maximum value, in order to prevent \( \Delta a \) to be significant factor. Optimal coefficient values, obtained by using algorithm for minimizing the rms, also meet this presumption.

IV. DISCUSSION/CONCLUSIONS

One of the most common solutions for student diagnosis in ITS is testing. Generally, test-based diagnosis systems use heuristic solutions to infer student knowledge. In contrast, Computerized Adaptive Testing (CAT) is a well-founded technique, which uses a psychometric theory called Item Response Theory (IRT) [10]. IRT supplies several methods to estimate student knowledge. All of them calculate a probability distribution curve \( P(\theta|u) \), where \( u=u_1, \ldots, u_i \) is the vector of items administered to students. When applied to adaptive testing, knowledge estimation is accomplished every time the student answers each item posed, obtaining a temporal estimation. The distribution obtained after posing the last item of the test becomes the final student knowledge estimation. One of the most popular estimation methods is the Bayesian method [11]. It applies the Bayes theorem to calculate student knowledge distribution after posing an item in.

Modeling students’ knowledge in educational games involves a high level of uncertainty [12]. For this reason, the presented model in this paper could be applied in a game educational application as a student knowledge diagnosis engine. In the modern age the educational process uses multimedia technologies increasingly. Instead to let students pass examinations on paper-based tests teachers may prefer to use the software application and let students show their knowledge through it. Sometimes, the application might stimulate student to find answers by themselves especially if application enables help windows with short description of current topic.

Game-based learning represents one kind of software applications that use games for learning or educational purposes. The aim of such application is to help student to understand topics by visual representations of processes that cover that topics. We have designed the GBL application that evaluates student knowledge related to topic of shifting bits in microprocessor. In order to evaluate student knowledge we have created the model that takes three input parameters: number of correct answers, number of help windows opened by students and duration needed to complete test. The rule for evaluating knowledge is calculated based on a simple formula with entering parameters such as the number of correct answers, time taken and how many times student has asked for help. This formula (1) is very important contribution in evaluating student’s knowledge in the process of game-based learning. The output of model estimates student’s knowledge graded into three grades: bad \((P(X)\leq45)\), good \((45<P(X)<85)\) and excellent \((P(X)\geq85)\).

In our model, each of input parameters is weighted and the main task has been to calculate weighting coefficients in order to results estimated in our model match the results obtained by classical paper-based test. By comparing the results with the same set of data in the archive data, we also came to a conclusion that the values in formula 1 are not very sensitive to certain changes. The accuracy of the evaluation stays almost the same if, for example, we change the value of the rate \( a \) from 0.15 to 0.30 and the value of the rate \( h \) from 0.30 to 0.55. Slightly larger sensitivity of evaluation has been noticed while changing the value of the rate \( t \). In order to maintain the model accuracy in student's knowledge evaluation, the value of the rate \( a \) can vary from 0.50 up to 0.75.

Once the coefficients are calculated with minimal error rule applied to results of testing group of students, they can be used in model for all students. The mistake made by our model compared to the results of classical examinations on a paper test was very small: \( \varepsilon = 0.18 \). We got encouraging results on the precision of students’ knowledge estimation generated by our model based on values for weighting coefficients. Thanks to model presented we have performed the evaluation of student knowledge of all student groups that attend the presented course.
ACKNOWLEDGMENT
This paper was supported by the Serbian Ministry of Education and Science (subproject 3 in project No. III47016, project No. TR35026 and project No. III44006).

REFERENCES