Estimating the Multiple Skills of Students in Massive Programming Environments

1st Fabiana Zaffalon
PPG Sciences Education
Federal University of Rio Grande
Rio Grande, Brazil
fabinhazaffalon@gmail.com

2nd André Prisco
PPG Sciences Education
Federal University of Rio Grande
Rio Grande, Brazil

3rd Ricardo de Souza
PPG Computational Modeling
Federal University of Rio Grande
Rio Grande, Brazil

4thDavi Teixeira
Center of Computational Sciences
Federal University of Rio Grande
Rio Grande, Brazil

5thMichel Neves
Center of Computational Sciences
Federal University of Rio Grande
Rio Grande, Brazil

6th Jean Luca Bez
Integrated Regional University
Erechim, Brazil

7th Neilor Tonin
Integrated Regional University
Erechim, Brazil

8th Rafael Penna
Center of Computational Sciences
Federal University of Rio Grande
Rio Grande, Brazil

8st Silvia Botelho
PPG Sciences Education
Federal University of Rio Grande
Rio Grande, Brazil

***Abstract*—This Research to Practice Full Paper presents a proposed model to estimate the multiple skills of students in massive online environments that provide programming exercises, whose assessment methods occur automatically without human intervention. The proposed model is based on the M-ERS model and incorporates, from the TrueSkill model, the uncertainty regarding the student's skills. To validate the model, a database from the URI Online Judge platform was used and the M-ERS and TriMElo models were applied to compare the performance and behavior of the two models. The empirical results show that the proposed model updates student's skills more smoothly, according to the correctness or error of the exercise, according to the uncertainty of the skills.**

***Index Terms*—skill, model, TriMElo, M-ERS, TrueSkill**

I. Introduction

There is an increasing number of massive educational environments that offer programming exercises in which assessment methods occur automatically without human intervention. Many teachers and students have incorporated the use of this platform as a didactic tool and as, in most cases, the classes are heterogeneous, that is, there are people with different skill levels, it is important that there is a precise method for monitoring the construction of the student proficiency.

For students in the field of computing, there are massive environments that offer materials and exercises aimed at learning programming, which includes a set of skills to be developed, such as logical reasoning, abstraction, data representation, among others; in addition to skills not directly linked to computing, for example text interpretation [13], [21]. In massive environments for teaching programming, it is common for automatic assessment systems to observe only the final result of the student's interaction with the learning object, not identifying the individual interaction of the multiple skills necessary to solve the problem. Thus, it is understood the importance of assessing this type of problem so that it is possible to identify precisely what skills need to be better developed in students in order to offer them the materials and exercises according to their skills.

Item Response Theory (IRT) [1], Elo Rating System (ERS) [8], TrueSkill [9] and M-ERS [15], are some methods that estimate student's skills. IRT is a set of mathematical models that seeks to represent the probability that an individual will answer an item correctly, depending on the item's parameters and the student's skill [1], [5]. The IRT presents compensatory multidimensional models that contemplate cases in which more than one skill is required for the student to correctly answer an item and that low skills can be compensated for by other higher skills [1], [5].

On the other hand, Elo, widely used for evaluation in international chess rankings, aims to classify competitors through their game histories, using a statistical classification that calculates their skills, relating a player to another player. In education, Elo establishes that a student is considered a player and the problem is considered his opponent and, in this way, updates the student's skill and the difficulty of the problems with each interaction between them [17], [19].

The TrueSkill [9] rating system is a model developed by Microsoft to rate players. It is based on performance expectations, extended from the Elo model to handle games with multiple teams and players. The goal is to identify and

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) – Finance Code 001.

track the skills of players in a game in order to combine them in competitive matches [11].

The M-ERS [15] model uses a compensatory IRT model and the Elo model to estimate the multiple skills of students in adaptive learning systems, assuming that an item may involve more than one skill and that a low skill may be offset by a higher one.

In massive environments where you only have access to the final result of the interaction between the subject and the object (right or wrong), the available models do not present an adequate monitoring of the progress of the skills individually, as happens in the human evaluation in which the teacher evaluates the solution completes the problem and is able to identify the mistakes and the lowest skills of the students. The multidimensional models found in the literature, by admitting only the homogeneous success or failure in relation to the different skills used in the solution, end up simplifying the evaluation so that, according to the result, all the skills involved are updated in the same proportion.

It can be seen, therefore, that in the context of massive online education, where the use of recommendation systems becomes imperative, the adoption of techniques of representation and assessment of skills based on methods based on expectation and performance may not be successful in the case of objects of learning that involve multiple skills. The fact of partially observing the subject's interaction with the object in relation to each required skill makes it difficult to analyze the actual performance versus the one whose expectation predicted the model. This difficulty entails an uncertainty that needs to be addressed for the success of the model.

This article proposes a model, based on the M-ERS and TrueSkill model, so that the skills are updated in different ways, not only according to their importance (or relevance) in the problem, but also taking into account the skills that were decisive for the correct solution of the problem or contributed to the error.

In the next section, the IRT, Elo, TrueSkill and M-ERS models will be detailed. Following the proposed model, TriMElo, will be presented. Then, the performance evaluation of the model will be disseminated through experiments using a database from an Online Judge do Brasil [3] platform. Finally, conclusions and future work are presented.

II. Theoretical Framework

A. Item Response Theory

Item Response Theory (IRT) is considered an important resource in the quantitative educational assessment process. It allows a more precise analysis of each item that makes up the assessment instrument, taking into account the characteristics of the items in the production of the skills [1], [6].

For the calculation of the θ skill estimate, IRT is based on statistical methods and mathematical models that consider not only the responses of individuals but also the properties of the items [1], [6]. The greater the individual's skill, the greater the probability of a correct answer to the item [1].

There are several proposed IRT models, which depend on several factors [1], [6]. For dichotomous items, there are 3 models that differ by the amount of parameters used to describe the item. They are: 1-Parameter Logistic Model or Rasch Model (considering only the item's difficulty), 2-Parameter Logistic Model (considering the item's difficulty and discrimination) and 3-Parameter Logistic Model (considering the difficulty, discrimination and the probability of success at random) [1], [6].

These models consider that the test is a one-dimensional instrument that implies the existence or predominance of only one skill, which does not apply in many practical situations. As an example, one can mention a math test that requires text interpretation before it even requires mathematical development, and in this case, it is a two-dimensional test, as it requires two skills [14].

Research has shown that the Multidimensional Item Response Theory (MIRT) model adapts better to real data than unidimensional models, because in education, subject's responses are determined by more than one skill at the same time [16].

MIRT models can be separated into two classes: compensatory (1) [20] and non-compensatory. A model is said to be compensatory when the probability of an item's success is maintained or increased even if one of the skills is low, which is compensated by another higher skill [14].

$$P(u_i = 1|\theta_j) = \frac{e^{a_{ik}\theta_{jk}+d_i}}{1 + e^{a_{ik}\theta_{jk}+d_i{}'}} \tag{1}$$

where u_i is the response to item i; a_{ik} is the item discrimination parameter i in the k dimension; θ_{jk} is the latent trait of the person j in the k dimension and d_i is a scalar indicating the item's difficulty.

B. Elo Rating System

The Elo classification system was proposed to analyze and classify the performance of chess players [8]. Through statistical methods, each player receives an initial rating of θ_i and, as they participate in the games, this rating is updated according to the results. When used in education, Elo establishes that a student is considered a player and the problem is considered his opponent [17]. The estimation takes place continuously, as the classification update takes place at the end of each resolution [17].

The Elo works according to the expectation and the result, the expected probability that the player will win the game is given by the logistic function in relation to the difference in the estimated ratings (2) [17]:

$$P(R_{ij} = 1) = \frac{1}{1 + 10^{\frac{\theta_j - \theta_i}{400}}} \tag{2}$$

wherw $R = \{0,1\}$ is the set of results of a game: 1 (win) and 0 (lose). Given a match between the player i and the player j, with Elo θ_i and θ_j, respectively, at the end of the game new Links are calculated according to the expectations of the

results, the previous Links and a constant k. The higher the k, the greater the Elo change (3) [17]:

$$\theta_i = \theta_i + k(R_{ij} - P(R_{ij} = 1)) \tag{3}$$

The use of the Elo classification system offers advantages, such as simplicity of use in online environments and implementation in educational systems, in addition to presenting a low number of parameters to adjust [17]. However, it is intended to track only a single skill [15].

C. Multidimensional Elo

There is an adaptation of the Elo model to estimate the multiple skills of students [Anonymous 2018]. The objective is to provide means to combine learning objects and students in a recommendation environment with programming problems, considering that each student has some more developed skills and others that need to be improved.

Elo's original approach to education considers students and learning objects to be a one-dimensional model, assuming that the theme of the proposed challenges can be reduced to a skill level. However, in the case of programming exercises, it is assumed that they are composed of different levels of maturity and skills. Thus, the multidimensional link assumes that the relationship between the student and the learning objects comprises a set of skills that can be relatively independent. In this context, a learning object may require more advanced levels in one skill and more basic levels in others, and may not even require some skill. Likewise, a student has different levels for each skill [18].

In the classic model, Elo is a scalar value for each student and for each learning object. The extended model of the Elo metric to make it multidimensional, considers that each dimension is a skill or concept that the student must have for himself at some level. Each student has his/her abilities as (4) [18]:

$$\overrightarrow{\theta_s} = (S_1, S_2, ..., S_N) \tag{4}$$

onde S_i is the Elo in the i skill. Each L learning object has its requirement level represented by $\overrightarrow{\sigma}$ and by relevance $\overrightarrow{m_l}$, where n is the number of learning objects. The set of learning objects can be represented by (5):

$$L = \{(\overrightarrow{\sigma_1}, \overrightarrow{m_1}), (\overrightarrow{\sigma_2}, \overrightarrow{m_2}), ..., (\overrightarrow{\sigma_n}, \overrightarrow{m_n})\} \tag{5}$$

Relevance m is a real number, between 0 and 1, which indicates how important a skill is for interacting with a learning object.

Considering the interaction of the student i with the learning object j, the adaptation to the model is represented by (6). For each s skill:

$$\begin{aligned} \theta_{i_s} &= \theta_{i_s} + m_{j_s} k(R_{ij} - P(R_{ij} = 1)) \\ \sigma_{j_s} &= \sigma_{j_s} + m_{j_s} k(R_{ji} - P(R_{ji} = 1)) \end{aligned} \tag{6}$$

Each interaction is a tuple I = (S, L, R), where R ={0.1 }, with 1 indicating that the student got the problem right or 0 indicating that he got it wrong.

D. TrueSkill Classification System

Microsoft developed the TrueSkill rating system to rate games with multiple teams and players. It is an extended Elo model for the purpose of identifying and tracking the skills of players to match them in competitive matches [11].

While in the Elo model the skill is determined by a single value, TrueSkill assumes that the player's skill follows a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where the skill is characterized by a normal curve in which the player's average (μ) represents the skill and the variance (σ) represents the uncertainty regarding the skill. The lower the uncertainty, the greater the confidence in the average value of the skill and, consequently, there is greater accuracy in the measure of the skill [4], [11]. An example of a TrueSkill skill is shown in Fig. 1, where the green area represents the confidence of the model that the player has a skill between 15 and 20.

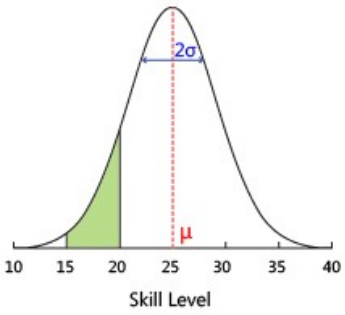

Fig. 1. Example of the skill of the TrueSkill model.

In the case of competitions between two players without a tie, The TrueSkill update procedure can be summarized in (7) [10], [12].

$$\begin{aligned} \mu_i &= \mu_i + \frac{\sigma_i^2}{c} \cdot v\left(\frac{\mu_i - \mu_j}{c}\right) \\ \mu_j &= \mu_j - \frac{\sigma_j^2}{c} \cdot v\left(\frac{\mu_i - \mu_j}{c}\right) \\ \sigma_i^2 &= \sigma_i^2 \cdot \left[1 - \frac{\sigma_i^2}{c^2} \cdot w\left(\frac{\mu_i - \mu_j}{c}\right)\right] \\ \sigma_j^2 &= \sigma_j^2 \cdot \left[1 - \frac{\sigma_j^2}{c^2} \cdot w\left(\frac{\mu_i - \mu_j}{c}\right)\right] \\ c^2 &= 2\beta^2 + \sigma_i^2 + \sigma_j^2 \\ v(x) &= \frac{\mathcal{N}(x)}{\Phi(x)} \\ w(x) &= v(x).[v(x) + x] \end{aligned} \tag{7}$$

where i is the winner, j is the loser, μ is the mean of the estimated, σ is the standard deviation of the estimated, $\mathcal{N}(x)$ is the probability density of a standard normal distribution, $\Phi(x)$ is the cumulative density of a standard normal distribution, and β is the distance that guarantees 76 per cent chance of winning [10].

The authors suggest values for initializing the parameters: μ_0 = 25; $\sigma_0 = \mu_0/3$; $\beta = \sigma_0/2$ [9].

E. M-ERS Model

The M-ERS [15] model, acronym for Multidimensional Extension of the ERS, presents an approach that incorporates the compensatory MIRT model to the Elo model, to track the estimates of the skill parameters in adaptive learning systems.

Rather than assuming a one-dimensional trace of item responses, the approach assumes that a single item can involve more than one skill. Thus, the model allows a simultaneous update of m different skills, estimating the probability of a correct answer P_{ij} through the compensatory MIRT (8) and updating the skill parameters, with the model Link (9), after each answer given to the item.

$$P_{ij} = P(Y_{ij} = 1) = \frac{e^{(\sum_{m=1}^{M} \alpha_{jm}\theta_{im} - \beta_j)}}{1 + e^{(\sum_{m=1}^{M} \alpha_{jm}\theta_{im} - \beta_j)'}} \tag{8}$$

where θ_{im} is the student's m skill i textit (m = 1, .., M), α_{jm} is the breakdown of the corresponding j item à m skill dimension and β_j is the general difficulty level of the item j.

$$\begin{aligned} \hat{\theta}_{im(t)} &= \hat{\theta}_{im(t-1)} + D_{m(t)} K\{Y_{ij(t)} - P_{ij(t)}\} \\ \hat{\beta}_{j(t)} &= \hat{\beta}_{j(t-1)} - D_{m(t)} K\{Y_{ij(t)} - P_{ij(t)}\} \end{aligned} \tag{9}$$

where $D_{m(t)}$ is a weight to specify whether the skill m is indicated by the item given in the t-th step. For the skill that is indicated by the item, $D_{m(t)}$ is equal to 1. For the skill that is not indicated by the item, the weight takes values between 0 and 1. K decreases linearly, between 0.4 and 0.1, depending on the total number of items answered.

Models that use the Elo classification system, decrease student's skills when there is a mistake (or defeat, in the case of a game) and increase their skill's when there is a correct answer in the exercises (or the player's victory).

III. TriMElo Model

In this section, the TriMElo model is proposed, which aims to simultaneously estimate the skills of students using massive online environments. The motivation for the TriMElo proposal arises from the perception that in the existing models that estimate the multiple skills of students and that use the MIRT and Elo models, they estimate the skills in a simplified way, varying the values of the skills according to the importance of the skills in the problems, without taking into account the same skills in the students. That is, if the student misses a certain problem that requires three skills, all of these skills will fall in their values, in the same proportion as the importance of these skills in the problem. Thus, if the student already has a high skill in one of them, when he misses the problem, that skill will have a relevant drop.

In massive problem-solving environments, where the platform corrects and gives automatic feedback on errors or successes, it is not possible to observe the individual interaction of students with each skill required in objects, this problem is even worse when considering that for each wrong submission all the skills involved will be affected in order to lower their values. Using as an example, a student who submits a problem that requires high math and programming skills; and that student's skills are high in math and low in programming, with each wrong submission both skills will be diminished. As his programming skill is low, it is normal for several submissions to occur until he is successful in the solution, in which case his mathematical skill will also decrease on a large scale until it can become very low, not consistent with the student's reality.

Thus, in this proposal, we seek to model the existing uncertainty in the importance of each skill in the performance presented by the student in the interaction with the object. It is proposed to identify elements in the teaching-learning process that allow estimating, based on history, heuristics that will constitute strategies for modeling this uncertainty. Different strategies can be adopted. To date, the following tactic has been stipulated: all the abilities of the students involved in the problems are individually observed and the non-determining skill is identified, the difference between the student's skill and that required by the problem is greater in relation to the others and, based on this analysis, these skills are updated according to the skill uncertainty factor estimated by the TrueSkill model.

To estimate the probability of a correct response from the student *i* to the item *j*, the compensatory MIRT equation similar to that used in the M-ERS model (10) is used:

$$P_{ij} = P(Y_{ij} = 1) = \frac{e^{(\sum_{m=1}^{M} \alpha_{jm}(\theta_{im} - \beta_j))}}{1 + e^{(\sum_{m=1}^{M} \alpha_{jm}(\theta_{im} - \beta_j))'}} \tag{10}$$

where α_{jm} is the relevance of the m skill in the j problem, θ_{im} is the m skill of the student i and β_j is the difficulty of the item j.

The parameters of the student's general skill and difficulty of the items are updated in the Elo model (11).

$$\begin{aligned} \hat{\theta}_{i(t)} &= \hat{\theta}_{i(t-1)} + \sigma_j k\{Y_{ij} - P_{ij}\} \\ \hat{\beta}_{j(t)} &= \hat{\beta}_{j(t-1)} + \sigma_j k\{Y_{ji} - P_{ji}\} \end{aligned} \tag{11}$$

where σ_j is the breakdown of the j problem, estimated by the IRT 2-parameter logistic model. k is a constant equal to 0.4 [22], which defines how much the estimate can be affected by the difference between the current and the expected response.

In order to update each skill, it is necessary to identify the determining skill, so that they are updated according to each case:

- Determining Skill (12):

$$\hat{\theta}_{im(t)} = \hat{\theta}_{im(t-1)} + \alpha_{jm} k\{Y_{ij} - P_{ij}\} \tag{12}$$

- Non-determining Skill (13):

$$\hat{\theta}_{im(t)} = \hat{\theta}_{im(t-1)} + \alpha_{jm}\Phi\{Y_{ij} - P_{ij}\}$$
$$\Phi = \sigma^2 \cdot \left[1 - \frac{\sigma^2}{c^2} \cdot \left(\frac{\theta_{im} - \alpha_{jm}}{P_{ij}}\right)\right] \quad (13)$$

As the *k* coefficient indicates the scale for updating the Elo [8], [17] value, the basic principle of the TriMElo model is to update non-determinant skills on a smaller scale by replacing *k* with the coefficient Φ (13) which is variance (σ) of the estimated skill according to the Trueskill model (7).

IV. Model validation

To validate the model, an experiment was carried out with the database made available by URI Online Judge [3], which has a repository of programming problems where users or students submit their solutions in one of the programming languages accepted by the platform and they receive automatic feedback of success or error. Users choose the problems to be solved and can submit several solutions for the same exercise.

The database consisted of 20.000 submissions, organized in chronological order, of 99 problems performed by 195 users of the URI platform. The available data were: date and time of submission, user id, problem id, answer (0 - error or 1 - correct). There was no access to the source codes of the problems solved.

Teachers who teach programming subjects, analyzed the problems and would assign a value between 0.5 and 1.5 regarding the relevance or importance of three skills that can be identified in the source codes. Skills are [2], [7]:

- Data analysis: write a program to solve basic statistical calculations [2]. In source code, this skill can be identified by the use of arithmetic operators [7].
- Data Representation: use data structures [2]. In programming it is understood that this skill is developed through exercises that use identifiers, arrays, pointers or structs [7].
- Abstraction: encapsulate a set of commands that are repeated for the same purpose, use conditionals, loops, recursion, etc. [2]. This skill is developed through the occurrence of logical operators used in conditions, repetition commands and recursive calls [7].

To estimate the difficulty and discrimination of the problems, the IRT model of 2 parameters was applied, using the package *mirt* of the RStudio software. For this, the data were tabulated in order to list users, problems and the respective answers (1 - correct or 0 - incorrect).

To apply TriMElo the data were organized in chronological order of submission and each of the skills was initialized with a value of 0.5. In the same database, the M-ERS model was applied in order to compare the performance and the results generated, presented in the next section.

V. Results and Discussions

After applying the models, the results obtained from some submissions were preliminarily analyzed. We chose to analyze only users with more than fifty submissions. In the present work, the data of a user who submitted 74 solutions to 63 different problems, obtained 54 hits and 20 errors was taken as an example. To verify the behavior of the skills updates, two cases were analyzed: one submission whose answer was incorrect and the other with a correct answer.

In submitting with the wrong answer, the problem required three skills: Analyze (relevance 0.9), Data Representation (relevance 0.7) and Abstraction (relevance 0.8).

The values of the user's skills were compared immediately before the update (pre-user) with the values after the update (post-user). In the Table I, the relevance values of the skills in the problem are related to the values of the user skills (pre and post) obtained by the TriMElo model and in the Table II, the values of the M-ERS model.

TABLE I
User skills obtained by the TriMElo model - Wrong Response

	Analyze	Data Representation	Abstraction
Problem	0.6	0.6	0.8
Pre-user	$\cong 0.675$	$\cong 0.709$	$\cong 0.500$
Post-user	$\cong 0.661$	$\cong 0.694$	$\cong 0.170$
Loss	$\cong 0.014$	$\cong 0.014$	$\cong 0.329$

TABLE II
User skills obtained by the M-ERS model - Wrong Response

	Analyze	Data Representation	Abstraction
Problem	0.6	0.6	0.8
Pre-user	$\cong 0.830$	$\cong 0.854$	$\cong 0.770$
Post-user	$\cong 0.738$	$\cong 0.762$	$\cong 0.644$
Loss	$\cong 0.092$	$\cong 0.092$	$\cong 0.125$

Observing the change in the values of the abilities, it is noticed that in both models the abilities suffered a fall, once the user erred the question. Analyzing each skill individually and comparing it with their relevance to the problem, one can assume which skills contributed to the error of the question. The user has the "Analyze" and "Data Representation" skills greater than the relevance required in the problem and the "Abstraction" skill below what is required, so it is believed that the "Analyze" skills and "Data Representation" did not contribute to the error. Observing the results of the two models, it is noted that TriMElo was the one that least "penalized" these skills and the possible cause of the error, the lowest skill in "Abstraction", suffered a greater penalty in the TriMElo model.

In submitting with the correct answer, the problem required both skills: Analyze (relevance 0.6) and Data Representation (relevance 0.7).

Again, the values of the user's skills were compared immediately before the update (pre-user) with the values after the update (post-user). In the Table III, the relevance values of

the skills in the problem are related to the values of the user skills (pre and post) obtained by the TriMElo model and in the Table IV, the values of the M-ERS model .

TABLE III
USER SKILLS OBTAINED BY THE TRIMELO MODEL - CORRECT ANSWER

	Analyze	Data Representation
Problem	0.6	0.7
Pre-user	≅ 0.675	≅ 0.674
Post-user	≅ 0.676	≅ 0.709
Gain	≅ 0.001	≅ 0.034

TABLE IV
USER SKILLS OBTAINED BY THE M-ERS MODEL - CORRECT ANSWER

	Analyze	Data Representation
Problem	0.6	0.7
Pre-user	≅ 0.696	≅ 0.696
Post-user	≅ 0.830	≅ 0.854
Gain	≅ 0.133	≅ 0.158

With the correct answer the user increased all his skills in both models. The user's "Analyze" skill is superior to that required by the problem and "Data Representation", inferior. In this case, the TriMElo model did not increase the value of the "Analyze" skill on a large scale, giving more value to the "Data Representation" skill that the user presented less than required. In the M-ERS model, the "Data Representation" skill also had a gain, but it was in the same proportion as the "Analyze" skill.

In that sense, the TriMElo model values skills in which users have a lower value then the required level to solve the problem, that is, it is expected that if the student can solve a problem even if he does not present the required skill, that skill should be updated in a different scale, accordingly to the required skill of the problem. The difference of values updated for skill is based on the expectation, were a user skill that matches the required skill of a problem is not considered determinant for success, and will receive a lower update value, in the other hand, user's skills that are below the required problem skill, are considered determinants for success, thus should receive a greater value.

The graphs in Fig. 2 show the evolution of this user's skills in both models. According to the graphs, it is easy to observe in both models the variation of skills according to the error or correctness of the problem. However, in the TriMElo model it is noticed that some skills have not changed in certain submissions. This was because these skills were not required in the solved problems. The M-ERS model also showed variations in skills, depending on whether it was right or wrong, but all skills were updated in all submissions.

The graph in Fig. 3 shows the evolution, in both models, of the "Analyze" skill that is present in all problems submitted by the user. It is noticed that there is a discrepancy in the values between the two models. This difference is related to the way in which each model adjusts its values; the TriMElo model updates the skill value on a smaller scale, considering

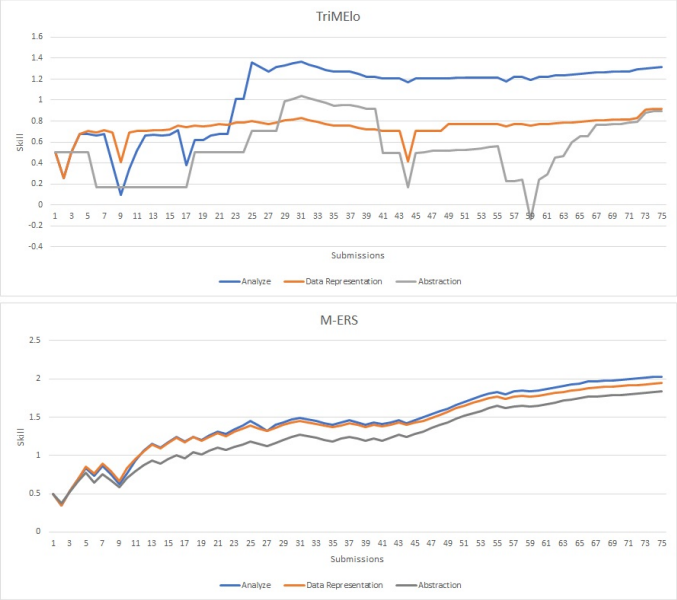

Fig. 2. Evolution of Skills in the TriMElo and M-ERS models.

the user's skill, the relevance of the skill to the problem and the uncertainty in relation to the user's skills.

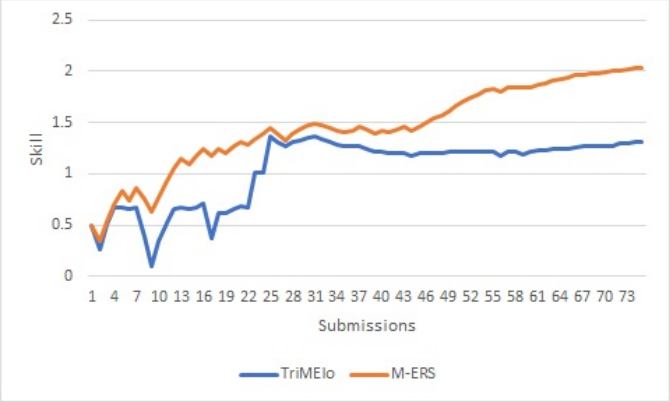

Fig. 3. Evolution of the "Analyze" skill in the TriMElo and M-ERS models.

From the point analysis with a user, Pearson's correlation coefficient was calculated for all submissions. With the result close to 0.858 it suggests that there is a strong correlation in the skills of the two models. Such correlation is justified by the way in which the models carry out the estimates, both increase the skills with the successes and decrease with the errors. What differentiates them is the way in which each one deals with the variation of skills.

VI. FINAL CONSIDERATIONS

In the present work, the TriMElo model was proposed, based on the M-ERS model, with the objective of dynamically estimating the progress of the users' skills in massive online programming system. The proposed model aims to update the skills considering their influence on the response in order

to adjust their values appropriately. The model considers the determining skills, which are the lowest skills of the user in relation to the relevance of the skill in the problem, where it can be assumed that, in case of error in the solution, these determining skills may have contributed to the failure. Thus, a correct answer should increase the value of the determinant skill, just as an incorrect answer should not over-penalize non-determinant skills.

Both models were applied to the same database in order to analyze and compare the behavior of the evolution of a user's skills. Differences in the values of the estimated skills were identified, especially in situations where the user's skill is superior to that required in the problem, in which case the TriMElo model updated to a lesser extent.

In general, the two models behave very similarly, which is justified by the fact that the TriMElo model uses M-ERS as a base. The main difference between them is in the different treatment of skills: TriMElo, before the updates, considers the user's skills and the relevance of these skills to the problem and infer the uncertainty factor as a skill update scale.

It is believed that the use of TriMElo in programming exercise recommendation systems may allow for a better recommendation when analyzing each user's skill and the skills involved in the exercises. Thus, as a future work, the aim is to apply the TriMElo model, through a case study, recommending programming exercises for students beginning in computing courses.

ACKNOWLEDGEMENT

Universidade Federal do Rio Grande (FURG) and Instituto Federal de Educação Ciência e Tecnologia Sul-rio-grandense (IFSUL).

REFERENCES

[1] F. B. Baker. *The basics of item response theory*. ERIC, 2001.

[2] V. Barr and C. Stephenson. Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community? *Acm Inroads*, 2(1):48–54, 2011.

[3] J. L. Bez, N. A. Tonin, and P. R. Rodegheri. URI Online Judge Academic: A tool for algorithms and programming classes. In *2014 9th International Conference on Computer Science Education*, pages 149–152, 2014.

[4] P. Dangauthier, R. Herbrich, T. Minka, and T. Graepel. Trueskill through time: Revisiting the history of chess. In *Advances in neural information processing systems*, pages 337–344, 2008.

[5] D. F. de Andrade, H. R. Tavares, and R. da Cunha Valle. Teoria da Resposta ao Item: conceitos e aplicações. *ABE, Sao Paulo*, 2000.

[6] D. F. de Andrade, H. R. Tavares, and R. da Cunha Valle. *Teoria da Resposta ao Item: conceitos e aplicações*. ABE - Associação Brasileira de Estatística, São Paulo, 2000.

[7] R. L. de Souza, F. Z. Ferreira, and S. S. da Costa Botelho. Proposta para avaliação de códigos fonte com tf-idf. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 112–121. SBC, 2020.

[8] A. E. Elo. *The rating of chessplayers, past and present*. Arco Pub., 1978.

[9] R. Herbrich, T. Minka, and T. Graepel. Trueskill™: A bayesian skill rating system. In *Advances in Neural Information Processing Systems*, pages 569–576. MIT Press, 2007.

[10] Y. Lee. Estimating student ability and problem difficulty using item response theory (irt) and trueskill. *Information Discovery and Delivery*, 2019.

[11] T. Minka, R. Cleven, and Y. Zaykov. Trueskill 2: An improved bayesian skill rating system. *Tech. Rep.*, 2018.

[12] T. Minka, Y. Zaykov, and J. Tims. Trueskill™ ranking system, 2005.

[13] G. L. Moreira, W. Holanda, J. C. d. S. Coutinho, and F. S. Chagas. Desafios na aprendizagem de programação introdutória em cursos de TI da ufersa, campus Pau dos Ferros: um estudo exploratório. *Anais do Encontro de Computação do Oeste Potiguar ECOP/UFERSA (ISSN 2526-7574)*, (2), 2018.

[14] R. T. Nojosa. Teoria da Resposta ao Item (TRI): modelos multidimensionais. *Estudos em Avaliação Educacional*, (25):123–166, 2002.

[15] J. Y. Park, F. Cornillie, H. L. van der Maas, and W. Van Den Noortgate. A multidimensional IRT approach for dynamically monitoring ability growth in computerized practice environments. *Frontiers in psychology*, 10, 2019.

[16] L. Pasquali. *TRI–Teoria de Resposta ao Item: Teoria, procedimentos e aplicações*. Editora Appris, 2018.

[17] R. Pelánek. Applications of the elo rating system in adaptive educational systems. *Computers & Education*, 98:169–179, 2016.

[18] A. Prisco, R. Santos, S. Botelho, N. Tonin, and J. Bez. A multidimensional Elo model for matching learning objects. In *2018 IEEE Frontiers in Education Conference (FIE)*. 2018.

[19] A. Prisco, R. Santos, N. T. S. Botelho, and J. Bez. Using information technology for personalizing the computer science teaching. In *2017 IEEE Frontiers in Education Conference (FIE)*. 2017.

[20] M. D. Reckase. Multidimensional Item Response Theory. *Handbook of statistics*, 26:607–642, 2006.

[21] A. Robins. Learning edge momentum: A new account of outcomes in cs1. *Computer Science Education*, 20(1):37–71, 2010.

[22] K. Wauters, P. Desmet, and W. Van Den Noortgate. Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers & Education*, 58(4):1183–1193, 2012.