A Model for Student Action Prediction in 3D Virtual Environments for Procedural Training

Diego Riofrío
ETSI Informáticos, UPM
Madrid, Spain
driofrio@fi.upm.es

Jaime Ramírez
ETSI Informáticos, UPM
Madrid, Spain
jramirez@fi.upm.es

ABSTRACT

This paper presents a predictive student action model, which uses student logs generated by a 3D virtual environment for procedural training to elaborate summarized information. This model can predict the most common behaviors by considering the sequences of more frequent actions, which is useful to anticipate common student' errors. These logs are clustered based on the number of errors made by each student and the total time that each student spent to complete the entire practice. Next, for each cluster an extended automata is created, which allows us to generate predictions more reliable to each student type. In turn, the action prediction based on this model helps an intelligent tutoring system to generate students' feedback proactively.

Keywords

Intelligent Tutoring Systems, Educational Data Mining, e-learning, procedural training, virtual environments

1. INTRODUCTION

Interactive simulations or virtual environments (VEs) have been used as tools to improve the learning by facilitating the "learning by doing" approach. Some of them show information to students through pictures, videos, interactive objects or help teachers make virtual lectures. However, there are some educative environments that can also supervise the execution of students' tasks by employing Intelligent Tutoring Systems (ITS), which provide tutoring feedback to students.

As a preamble to this work, a 3D biotechnology virtual lab was developed by our research group [4]. After evaluating this virtual lab, we saw opportunity to include the power of data mining to improve its automatic tutor by taking advantage of student logs.

Despite the work that has already been done about ITS in Educational Data Mining (EDM), the community misses more generic results [5]. Furthermore, it is also remarkable the lack of ITSs that take advantage of models developed by EDM [1].

The work presented in this paper represents a step forward towards the development of an ITS that leverages a predictive model computed by means of EDM to offer a better tutoring feedback. Moreover, this ITS is intended for procedural training in VEs and is domain independent.

Section 2 describes the proposed architecture for the ITS, which leverages the predictive student model (section 3). Finally, in section 4 we show the conclusions of this work.

2. ITS ARCHITECTURE PROPOSAL

The ITS architecture proposal is inspired on MAEVIF architecture [3], which is an extension of the ITS classical architecture for VEs.

Our main contribution resides in the Tutoring Module, which has a Tutoring Coordinator that validates the students' actions and shows error messages or hints. This module also comprises the Student Behavior Predictor (SBP) and within it lies the Predictive Student Model, which is used to find out the next most probable action from the last action made by the student. This information is used to anticipate probable students' errors, which provides a mechanism to avoid them as long as it is pedagogically appropriate.

3. PREDICTIVE STUDENT MODEL

Predictive student model uses historical data from past students and is continually refined (as Romero and Ventura recommend [5]) with actions that students under supervision are doing. In the context of the KDD Process and its adaptation into EDM formulated by Romero and Ventura [5], this model is created in Models/Patterns phase.

The model contains summarized data from historical registries of actions made by past students, and it is used to obtain the next most probable student's action. It consists of several clusters of students where each of them contains an extended automata, detailed in section 3.1. These clusters help to provide automatic tutoring adapted to each type of student. For example, if the student is committing few errors, it is more probable that his/her next action will not be an error. However,it will happen the opposite to a student who has failed more times.

The process of creation of this model is similar to the one proposed by Bogarín et. al. [2], and it is executed at the tutor start-up. Basically, this process consists in taking events from student logs and from them data clusters of students are created based on the number of errors and the time they spent to complete the entire training process. Then, an automata for each cluster is built from the logs of the students using an incremental method. Later, at training time the SBP component updates the model with each new student's action attempt.

3.1 Extended Automata Definition

This automata consists of states (represented by circles) and transitions (represented as arrows) as shown in figure 1. Furthermore, states are grouped into three zones: Correct Flow, Irrelevant Errors and Relevant Errors Zone.

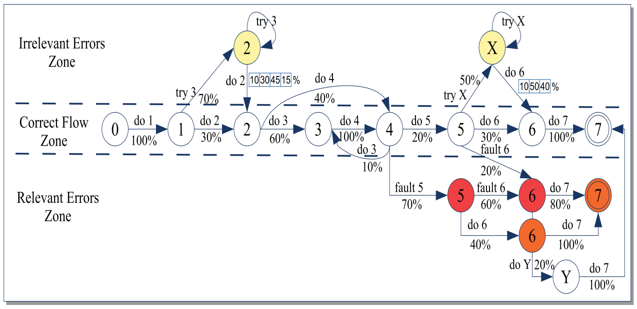

Figure 1: Example of an extended automata

Transitions denote events across an exercise such as actions or action attempts that past students have performed so far and new students may repeat in the future. An event may be a valid action of an exercise or an error detected by the tutor at the time of validating an action attempt. Accordingly, states represent the different situations derived from the events provoked by students.

Each state, and each transition, contains the number of students whose logged sequences of events have passed through, which becomes into event probabilities between states. In the case of states with loops, event frequencies to next state are reflected in a vector. In this way, the probability that a student leaves the loop on each iteration can be represented.

3.1.1 Correct Flow Zone

In this area, events represent the valid sequence of actions for an exercise, which ends up with a final satisfactory state. These states are represented by white circles.

3.1.2 Irrelevant Errors Zone

This zone groups states derived from error events that do not influence in the final result. These error events are associated with action attempts blocked by the automatic tutor (blocking errors [4]). These are graphically represented by a yellow circle.

3.1.3 Relevant Errors Zone

This area encompasses states derived from error events that actually influence in the final result, i.e. if an event of this type occurs the final result will be wrong unless a repairing action is done (non-blocking errors [4]). In this case there will be an error propagation to the subsequent states, because it does not matter what the student does later (except for some repairing action), the subsequent states will be considered also erroneous. The states derived directly from these errors are graphically represented by red circles and the subsequent correct states by orange circles.

In addition, repairing actions can be found in this area. These actions fix errors occurred earlier and redirect to one state in the correct flow.

4. CONCLUSIONS

Our proposal achieves an automatic tutoring in procedural training more adapted to each type of student by applying methods of extraction and analysis of data, which can anticipate possible errors depending on its configuration.

The principal application of the presented predictive model is to help students with preventing messages. For this, we have designed an ITS, presented above, which leverages the predictive model to provide that kind of tutoring.

We consider that the advice of an expert educator or teacher of the subject is essential at design time, despite this ITS may become very independent once its tutoring strategy is configured. This is because the resulting predictive model need to be analyzed for refining the tutoring strategy. In order to facilitate this task, it will be necessary to develop an application that displays the model to the expert or professor. In this way, he/she could visualize where students make more mistakes or where the practice is easier for them, and with this information he/she could decide where and what tutoring feedback is needed. Additionally, this could also help teacher to improve his/her own teaching.

5. ACKNOWLEDGEMENTS

Riofrío thanks Secretariat of Higher Education, Science, Technology and Innovation from Ecuador (SENESCYT).

6. REFERENCES

[1] R. S. Baker. Educational data mining: An advance for intelligent systems in education. *Intelligent Systems, IEEE*, 29(3):78–82, 2014.

[2] A. Bogarín, C. o. b. Romero, R. Cerezo, and M. S a nchez-Santill a n. Clustering for improving educational process mining. In *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*, pages 11–15. ACM, 2014.

[3] R. Imbert, L. Sánchez, A. de Antonio, G. Méndez, and J. Ramírez. A multiagent extension for virtual reality based intelligent tutoring systems. *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*, pages 82–84, 2007.

[4] M. Rico, J. Ramirez, D. Riofrío Luzcando, M. Berrocal-Lobo, A. De Antonio, and D. Riofrio. An architecture for virtual labs in engineering education. In *Global Engineering Education Conference (EDUCON), 2012 IEEE*, pages 1–5, 2012.

[5] C. Romero and S. Ventura. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1):12–27, 2013.