Student Action Prediction for Automatic Tutoring for Procedural Training in 3D Virtual Environments

Diego Riofrío-Luzcando(✉) and Jaime Ramírez

ETSI Informáticos, UPM, Madrid, Spain
{driofrio,jramirez}@fi.upm.es

Abstract. This paper presents a way to predict student actions, by using student logs generated by a 3D virtual environment for procedural training. Each student log is categorized in a cluster based on the number of errors and the total time spent to complete the entire practice. For each cluster an extended automata is created, which allows us to generate more reliable predictions according to each student type. States of this extended automata represent the effect of a student correct or failed action. The most common behaviors can be predicted considering the sequences of more frequent actions. This is useful to anticipate common student errors, and this can help an Intelligent Tutoring System to generate feedback proactively.

Keywords: Intelligent Tutoring Systems · Educational Data Mining · e-learning · Procedural training · Virtual environments

1 Introduction

Interactive environments have been used as tools to improve the "learning by doing" approach. Virtual worlds are one class of this environments, and are one of the most promising lines of research and development in e-learning.

Two important events in the evolution of educational technology are the emergence of Intelligent Tutoring Systems (ITS), and the rise of Educational Data Mining (EDM) [18] thanks to Learning Management Systems.

As a preamble of the work presented in this paper, a 3D biotechnology virtual lab was developed by our research group [15]. As part of this lab a Reactive Tutor was implemented, which can validate the students' actions and show hints to them, whenever is pedagogically appropriate. After evaluating this virtual lab, we saw the chance to include data mining of the students' logs to improve this tutor.

To provide a better and adaptive tutoring feedback we propose a model that can be used to infer the next most probable action or error of each student. This feedback could prevent students before committing an error, as long as this will be a pedagogically appropriate prevention. This model is intended for any 3D virtual environment (VE) for procedural training, although our starting point

G. Vincenti et al. (Eds.): eLEOT 2015, LNICST 160, pp. 200–207, 2016.
DOI: 10.1007/978-3-319-28883-3_25

was the virtual lab of biotechnology. In addition, we think that this model also may be applied to 2D virtual environments, as long as they serve for procedural training.

Section 2 shows relevant works in the field of educational VEs and EDM. Next, Sect. 3 describes the proposed architecture for the ITS, which leverages the predictive student action model detailed later in Sect. 4. Finally, in Sect. 5 we show the conclusions of this work and future work.

2 Related Work

We have divided this section into two subsections, the first presents an overview of educational virtual environments for procedural training and the second one addresses the use of data mining to improve education.

2.1 Educational Virtual Environments

In the literature exists a wide variety of educational VEs. Some of them show information to students through pictures, videos and interactive objects or help teachers make virtual lectures. Other environments create situations in which students have to perform some tasks. Moreover, there are some VEs that, in addition to simulate situations, supervise the execution of these tasks by employing an ITS and provide clues to students about the actions to be executed.

Among these applications, one of the most well-known is *STEVE* [11], a 3D animated agent that assist one or more students to learn a task that follows a given procedure. A project that takes advantage of STEVE's architecture is MRE (Mission Rehearsal Exercise) [20].

Another recognized project is Lahystotrain [13], developed to train surgeons in laparoscopy and hysteroscopy operations. Students receive proactive and reactive help from an automatic tutor on how to properly perform the surgery.

To the best of our knowledge, these applications were not designed in a generic way, so that some of their key components cannot be reused in new applications easily. However, there exist a few proposals where generic architectures for educational VEs are present. These were designed to provide an automatic tutor that can be adapted to different educational aims. One of these proposals is called MAEVIF [7,10] and consists in a multi-agent system that represents an extension of the classic ITS architecture [19] specifically intended for VEs.

Another generic architecture is defined in MASCARET (Multi-Agent Systems to simulate Collaborative, Adaptive and Realistic Environments for Training) [4], which is an agent based system that integrates an ITS. A recent work by the same authors is called Pegase [3], and proposes a generic and adaptable ITS based on a multi-agent system.

2.2 Educational Data Mining

A large majority of researchers in the field of EDM have been dedicated to study data or logs registered by e-learning systems like Moodle. One of these works is called System for Educational Data Mining (SEDM) [12], and it tries to study the representative behavior of students that had dropped-out an online course before taking the final exam versus the others who did not.

Another work uses Moodle logs to discover a specific behavior student model [2]. They divide these logs in student groups with similar characteristics using a clustering method and then apply process mining to each cluster to create a model (represented by a directed cyclic graph) that shows the most frequent sequences of students' actions.

The authors found that graphs, models or visual representations are easier to comprehend, and making these representations accessible to teachers and students their results could be very useful for monitoring the learning process and providing feedback.

Other works analyze data collected from students' grade records, for example, Dekker et al. [8] try to predict students drop-out rate at the Eindhoven University of Technology, with the aiming to provide individual attention to students in the "risk group".

Some works try to solve specific problems, like CanadarmTutor Tutoring System [9], which simulates the robotic arm Canadarm2 used in the International Space Station. This ITS provides assistance to users on how to follow a correct solution path with hints of the arm operations. For this, it integrates a cognitive model to assess skills and spatial reasoning, and an expert system that generates solution paths automatically. In addition, it uses data mining techniques to extract a partial task model from past user solutions.

Despite the work that has already been done in this area, the community misses more generic results [17], i.e., results than can be used with other students, or replications of experiments showing that a predictive model can be applied in more than one distinct context. It is also remarkable the lack of intelligent educational systems 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.

3 ITS Arcitecture Proposal

The ITS architecture proposal is inspired on MAEVIF architecture [7,10] and is depicted in Fig. 1. The modules of this architecture are: Communication Module, Student Module, Expert Module, World Module and Tutoring Module.

The Communication Module serves as a link between the graphic engine (OpenSim, Unity 3D, Open Wonderland, etc.) and the tutoring system.

The Student Module (integrated as an agent within the MAEVIF [7,10]) is an adaptive, extensible and reusable student model that infers the student knowledge state using a pedagogic-cognitive diagnosis with non-monotonic reasoning abilities. The purpose of this module is to know the status of each student's learning, that is, what he/she knows of the subject or what he/she does not. This can serve to support a personalized automatic tutoring for each student.

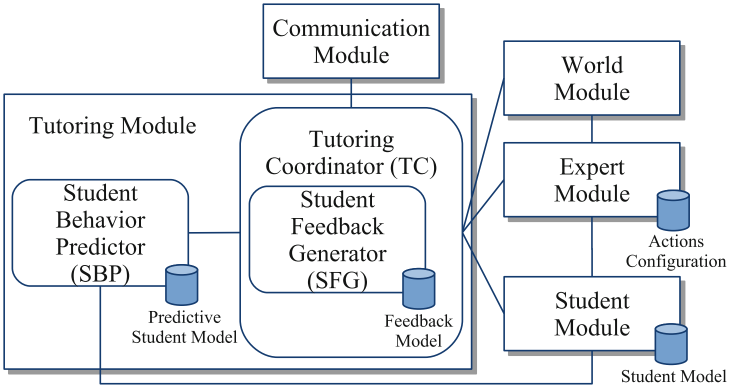

Fig. 1. ITS architecture proposed

One of the disadvantages of the student module is that it needs a lot of prior information on each student and about the knowledge addressed in the actual course to give information with high confidence. This work is detailed in a journal paper and in a Ph.D. dissertation [5,6].

Next, the Expert Module defines the procedure to be learned and the tutoring strategy for each action. This procedure details every step or action that the student should perform and the preconditions (detailed in [15]) that must be met to consider such action as valid in a certain moment of the practice.

The World Module keeps information on the objects implemented inside the VE and their status. This module mainly manages locks of the objects that are being used by students, and therefore they cannot be used by others.

Tutoring Module works as the core of the ITS, because it performs most of the computational logic of the tutoring task. This module is responsible of validating the students' actions by using the information in the Expert Module. In addition, it is also responsible for showing the hints or error messages that more pedagogically appropriate in each moment. The Tutoring Coordinator (TC) and the Student Behavior Predictor (SBP) are the main components of it.

The TC comprises the operation of the reactive tutor, detailed in a paper and in a master thesis [15,16], regarding the validation of students' actions. If the action attempt is correct, it informs this attempt to the SBP and the Student Module, and depending on what they reply the TC generates a feedback using its sub-component called Student Feedback Generator (SFG).

The SBP is the component that concerns us in this paper, because within it lies the Student Predictive Model. This component attempts to predict the next most probable action based on the action records of past students. For this, it relies on the Predictive Student Model that has these action records summarized. Once the most probable action is found, it is delivered to the SFG along with the confidence of this prediction.

4 Predictive Student Model

This model is created using historical data from past students and is continually refined with the actions from students under supervision. Taking into account the Knowledge Discovery in Databases Process and its adaptation into EDM formulated by Romero and Ventura [17], our model can be considered a result of the Models/Patterns phase of this well-known process.

The idea of building this model arose from our experiences evaluating the 3D virtual lab of biotechnology [15]. During the model design, it was necessary to observe the behavior of students in the virtual world following the ethnographic method. Subsequently, as recommended by Mostow et al. [14], the logs generated by the reactive tutor were analyzed by hand to identify interesting phenomena.

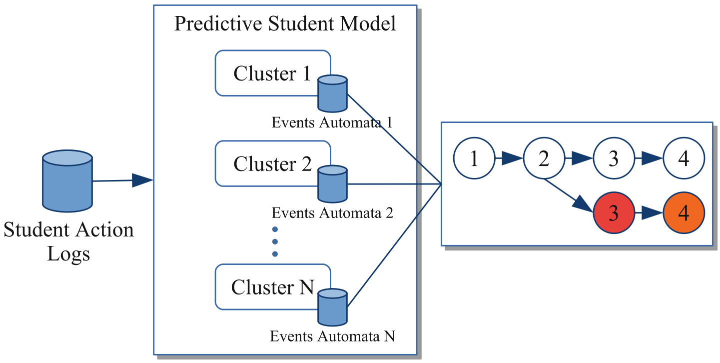

Fig. 2. Predictive Model

The model (see Fig. 2) consists of several clusters of students where each of them contains an extended automata (detailed in Sect. 4.1). These clusters help to provide automatic tutoring adapted to each type of student.

The creation process of this model is similar to the one proposed by Bogarín et al. [2], and it is executed at the tutor start-up. In fact, it consists in taking events from student logs stored in the Student Model. First, students in this logs are clustered 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 student logs included in each cluster. Later, at training time the SBP component updates the model with each new student action informed by the TC component.

4.1 Extended Automata Definition

This automata consists of states (represented by circles) and transitions (represented as arrows) as shown in Fig. 3. Furthermore, the states are grouped into three zones: Correct Flow, Irrelevant Errors and Relevant Errors Zone.

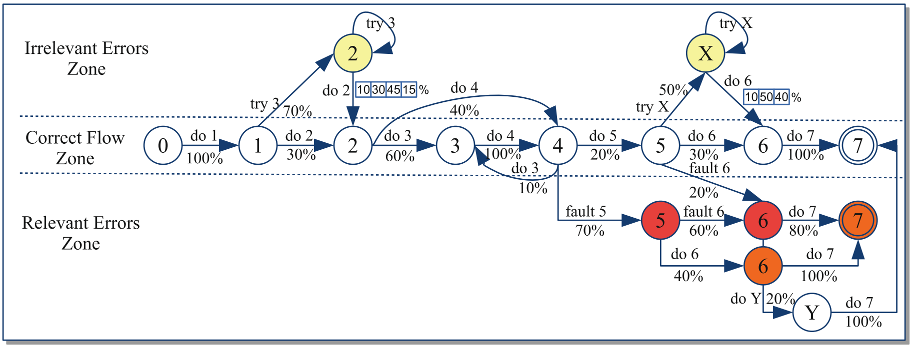

Fig. 3. Example of an extended automata

Transitions denote events provoked by students 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.

It can be seen in Fig. 3 how an event associated with a right action, registered in student logs, produces a correct state. For example, the event of "doing action 1" conducts to state 1 and the "doing action 3" event conducts to state 3. Each state and each transition contain the number of students whose logged sequences of events have passed through, which permit to calculate event probabilities between states.

In the case of states with loops, event frequencies to next states are reflected in a vector. In this way, the probability of a student leaving the loop at each iteration can be calculated.

Correct Flow Zone. In this area are the events that represent the valid sequence of actions for an exercise, which logically ends up with a final satisfactory state. These states are graphically represented by white circles (see Fig. 3).

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 tutor (blocking errors [15]). These are graphically represented by a yellow circle.

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 [15]). 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.

5 Conclusions

This proposal achieves an automatic tutoring 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, i.e., messages that can prevent students from making possible future errors. For this, we have designed an ITS, presented above, which leverages the predictive model to provide that kind of tutoring.

We saw that data mining results from student logs can allow ITSs to discover which tutoring actions help more students in each situation. As a direct result, the ITS can help students in a more personalized way and in real time (as also pointed out by Mostow et al. in [14]).

We agree with Mostow et al. [14] in which 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 needs 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, this person 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 a teacher to improve his/her own teaching.

Another future line of work is to improve the world module to use an ontology that defines the virtual world objects. With this and some other extensions, the ITS could offer a more flexible tutoring feedback using objects salience or the calculation of the objects paths.

Acknowledgements. I would like to acknowledge Secretariat of Higher Education, Science, Technology and Innovation from Ecuador (SENESCYT) for their economic support.

References

1. Baker, R.S.: Educational data mining: an advance for intelligent systems in education. IEEE Intell. Syst. **29**(3), 78–82 (2014)
2. Bogarín, A., Romero, C., Cerezo, R., Sánchez-Santillán, M.: Clustering for improving educational process mining. In: Proceedings of the Fourth International Conference on Learning Analytics And Knowledge, pp. 11–15. ACM, New York (2014)

3. Buche, C., Querrec, R.: An expert system manipulating knowledge to help human learners into virtual environment. Expert Syst. Appl. **38**(7), 8446–8457 (2011)
4. Buche, C., Querrec, R., De Loor, P.: MASCARET: pedagogical multi-agents systems for virtual environment for training. Cyberworlds (2003)
5. Clemente, J.: Una Propuesta de Modelado del Estudiante Basada en Ontologías y Diagnóstico Pedagógico-Cognitivo no Monótono. Ph.D. thesis (2011)
6. Clemente, J., Ramírez, J., de Antonio, A.: A proposal for student modeling based on ontologies and diagnosis rules. Expert Syst. Appl. **38**(7), 8066–8078 (2011)
7. de Antonio, A., Imbert, P.R., Méndez, G., Ramírez, J.: Intelligent virtual environments for training: an agent-based approach. In: Pěchouček, M., Petta, P., Varga, L.Z. (eds.) CEEMAS 2005. LNCS (LNAI), vol. 3690, pp. 82–91. Springer, Heidelberg (2005)
8. Dekker, G.W., Pechenizkiy, M., Vleeshouwers, J.M.: Predicting students drop out: a case study. In: International Working Group on Educational Data Mining, ERIC (2009)
9. Fournier-Viger, P., Nkambou, R., Nguifo, E.M., Mayers, A., Faghihi, U.: A multi-paradigm intelligent tutoring system for robotic arm training. IEEE Trans. Learn. Technol. **6**(4), 364–377 (2013)
10. Imbert, R., Sánchez, L., de Antonio, A., Méndez, G., Ramírez, J.: A multiagent extension for virtual reality based intelligent tutoring systems. In: Seventh IEEE International Conference on Advanced Learning Technologies, 2007, ICALT 2007, pp. 82–84 (2007)
11. Johnson, W.L., Rickel, J.: Steve: an animated pedagogical agent for procedural training in virtual environments. SIGART Bull. **8**(1–4), 16–21 (1997)
12. Lara, J.A., Lizcano, D., Martínez, M.A., Pazos, J., Riera, T.: A system for knowledge discovery in e-learning environments within the European Higher Education Area Application to student data from Open University of Madrid, UDIMA. Comput. Educ. **72**, 23–36 (2014)
13. Los Arcos, J.L., Muller, W., Fuente, O., Orúe, L., Arroyo, E., Leaznibarrutia, I., Santander, J.: LAHYSTOTRAIN: integration of virtual environments and ITS for surgery training. In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) ITS 2000. LNCS, vol. 1839, pp. 43–52. Springer, Heidelberg (2000)
14. Mostow, J., Beck, J.: Some useful tactics to modify, map and mine data from intelligent tutors. Nat. Lang. Eng. **12**(2), 195–208 (2006)
15. Rico, M., Ramírez, J., Riofrío, D., Berrocal-Lobo, M., De Antonio, A.: An architecture for virtual labs in engineering education. In: Global Engineering Education Conference (EDUCON) 2012, pp. 1–5, IEEE (2012)
16. Riofrío, D.: Diseño e Implementación de un Laboratorio Virtual de Biotecnología. Master's thesis, ETS de Ingenieros Informáticos, Universidad Politécnica de Madrid (2012)
17. Romero, C., Ventura, S.: Data mining in education. Wiley Interdisc. Rev. Data Min. Knowl. Disc. **3**(1), 12–27 (2013)
18. Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.: Handbook of Educational Data Mining. CRC Press, Boca Raton (2010)
19. Sleeman, D., Brown, J.S.: Intelligent Tutoring Systems. Academic Press, New York (1982)
20. Swartout, W., Hill, R., Gratch, J., Johnson, W.L., Kyriakakis, C., LaBore, C., Lindheim, R., Marsella, S., Miraglia, D., Moore, B.: Toward the holodeck, integrating graphics, sound, character and story. Technical report (2006)