



Framework for Learner Assessment in Learning Games

Mathieu Muratet, Amel Yessad, Thibault Carron

► To cite this version:

Mathieu Muratet, Amel Yessad, Thibault Carron. Framework for Learner Assessment in Learning Games. 11th European Conference on Technology Enhanced Learning, Sep 2016, Lyon, France. pp.622-626, 10.1007/978-3-319-45153-4_77 . hal-01359636

HAL Id: hal-01359636

<https://hal.science/hal-01359636>

Submitted on 8 Feb 2022

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.

Framework for Learner Assessment in Learning Games

M. Muratet^{1,2}, A. Yessad¹, and T. Carron^{1,3}

¹ Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris, France

² INS HEA, 58-60 Avenue des Landes, 92150 Suresnes, France

³ Université Savoie Mont Blanc, 73376 Le Bourget du Lac, France

{mathieu.muratet, amel.yessad, thibault.carron}@lip6.fr

Abstract. Learner assessment in learning games (LG) is an interesting research area for both academia and industry. The play traces resulting from the learner’s activity in LGs with large state spaces and a large amount of free interactions, are hard to analyze and to interpret by teachers. In this paper, we present a framework to assist the building of an expert’s solving process that is the base of the algorithm that analyzes player’s traces and generates pedagogical labels about the learner’s behavior.

Keywords: Learning Game, Behavioral Model, Petri Net

1 Introduction and positioning

Learner assessment is considered as a key issue in Technology-Enhanced Learning (TEL). Learners are not all alike, and it is interesting to assess the behavior of each learner who uses the system in order to implement adapted scenarios and provide feedback. Our approach aims (1) to be seamlessly integrated with a LG, rather than presented as a separate artificial assessment disconnected from the nature of the task and (2) to compare a learner’s behavior with experts’ solving model.

Our research focuses on learning games which simulate process (physical, industrial, business, etc.). In this kind of complex systems with a large amount of freedom in interaction, it is hard to model game actions and experts’ solving processes in order to understand and to analyze students’ activity. Thus, our objective is to assist designers in building a model of the experts’ solving and to compare it with the learner’s solving in order to generate a description of learner’s behavior, readable by teachers and designers.

Several research had already considered the issue of automatic assessment of learners by analyzing play traces. Thus, in [1], the authors propose a methodology for extracting conceptual features from student’s log data using a two-dimensional context-free grammar. This contribution is focused on puzzle games like *RumbleBlocks*⁴ or *Refraction*⁵. Other research used Petri nets to describe experts’ solving of “Case study” games and proposed an algorithm to label learners’ actions [2]. However, this algorithm is adapted to unique type of games (case studies) and is not at all suitable for learning games with large state spaces and a large amount of freedom in interaction.

⁴ RumbleBlocks: <http://rumbleblocks.etc.cmu.edu/> accessed April 4, 2016

⁵ Refraction: <http://games.cs.washington.edu/refraction/refraction.html>, accessed April 4, 2016

Our approach shares the same objective with these approaches but aims to propose a scalable and generalizable framework giving more accurate pedagogical information about the learner’s behavior. The pedagogical labels defined are based on the comparison between the learner’s behavior and the expert’s solving of a game level. Fig. 4 depicts the global architecture of the assessment framework. In this paper, we focus on the workflow that assists designers to build the model of experts’ solving.

2 Assistive workflow to build the expert’s solving process

A key point in our methodology is to model the experts’ solving process by a executable model and to assess the learners actions by comparing them with this model. Like [2], we choose to use Petri nets which is a powerful modeling formalism in computer science, system engineering, and many other disciplines (see [3] for details on Petri nets). Petri net combines a well-defined mathematical theory with a graphical representation of the system’s behavior. The theoretical aspect of Petri nets allows precise modeling and analysis of system behavior [4]. However, modeling a complex simulation game with Petri nets is a difficult task both for game designers and experts. The main difficult task is to assure consistency between Petri net modeling and game simulation. In our framework, we propose an assistive workflow to semi-automate the Petri net building.

2.1 Example: The frozen door

We illustrate our contribution with a simple example of a frozen door. Fig. 1 depicts a simple Petri net of a door that the player can open or close (in the initial marking, the door is closed). If the door is connected to other game objects like a key, then this Petri net is extended in order to match with the simulation (cf. Fig. 2). In this second Petri net, the door is locked and the key is required to open it. We also added a boiler into this game level that the user has to turn on in order to solve the level. In the initial marking of this Petri net, the door is closed, the boiler is turned off and the key is in the inventory.

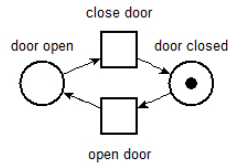


Fig. 1. Petri net of a door that the player can open or close.

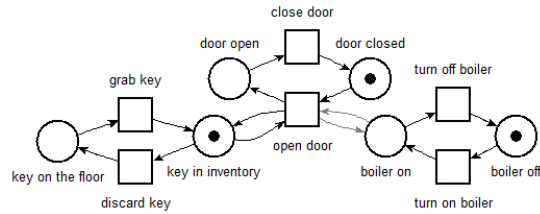


Fig. 2. Full Petri net of a frozen door. Only gray arcs are manually added. The other places, transitions and arcs of this figure are built automatically.

In order to implement the automatic learner assessment, we construct two Petri nets semi-automatically. The first Petri net is called “Full Petri net” (FullPn) and includes all

actions that learners can perform in the game. A FullPn models game simulation and its marking depicts the state of the simulation. The second Petri net, called “Filtered Petri net” (FilteredPn), is a part of the FullPn and includes only actions used by experts to solve the current level. It embeds the expert’s action sequences allowing to solve the game level.

The building of the FullPn is a challenging task due to the high number of actions that the learner can perform in each state of the game. This building process has to be automatic or at least semi-automatic. In our work, the semi-automatic building of Petri nets is based on the definition of game objects and their properties. Each game object is described by the actions the user can perform on. For instance, in the role playing game we used to test our framework, the object “the door” can be opened or closed, the object “the key” can be grabbed or discarded and the object “the boiler” can be turned on or turned off. The objects and their properties are described in a user friendly editor called Tiled⁶. We have implemented a complex XSLT transformation to build a Petri net from Tiled game object descriptions (for instance, Fig. 2 is the result of this transformation for a simple level, only the gray arcs were added manually). We can summarize the benefits of this transformation process by the following points: (1) the transformation process is weak-dependent on the game level because once the game objects are described in Tiled, the transformation is not changed and the game object can be reused in several levels; (2) the effort of developing the transformation is performed once, while we can use it many times, at each game level; (3) the transformation generates less errors than the manual building of Petri nets; and (4) the Petri net building has to be validated/completed by LG designers, but the validation task is less time-consuming and less complicated than building a Petri net from scratch.

Once we have built the FullPn, we filter it by removing transitions that are not used by experts, in order to build the FilteredPn. In the example of the frozen door (cf. Fig. 2), the objective is that the player opens the frozen door. The expert’s solving consists in turning on the boiler and opening the door with the key. Formally, it corresponds to fire, in sequence, the transition “turn on boiler” and then the transition “open door”. Fig. 3 represents the FilteredPn that results from the filtering of the FullPn of Fig. 2. Once the FilteredPn is built, we compute its reachability graph that serves us to analyze the learners’ actions.

2.2 Workflow overview

As depicted in Fig. 4, the designers start by using a user-friendly graphical tool to build a game level. From this high level game description, we use an XSLT transformation to build two files: (1) a low level game description that is compatible with the game engine and (2) a Petri net that describes the game simulation (the FullPn). An expert can play this new level (several times if several solutions are available) and the game engine traces the expert’s actions. These traces are used to filter the FullPn and build the FilteredPn. We notice that a non-expert’s trace could be used to filter the FullPn, for example an original and correct solving made by a learner positively assessed by teacher can be added to the expert’s traces to enlarge the FilteredPn.

⁶ Tiled: <http://www.mapeditor.org/>, accessed April 4, 2016

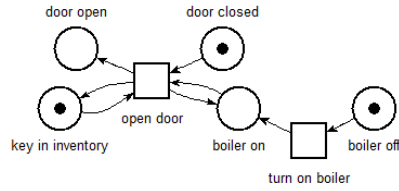


Fig. 3. Filtered Petri net of the frozen door. Only transitions (actions) used by the expert are kept from the Full Petri net, here, turning on the boiler and opening the door.

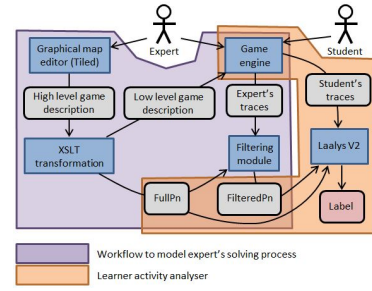


Fig. 4. Global architecture of the assessment framework.

Once the FullPn and the FilteredPn are generated, they can be validated or completed by expert/designer manually in order to include constraints not configurable with the graphical editing tool. Then, this validated FilteredPn is used by the labeling algorithm to label learners' actions pedagogically.

3 Conclusion

The work presented in this paper deals with the assessment of learners' behavior in learning games. This paper focuses on a workflow to help the designers to model expert's solving with Petri nets. We illustrated the methodology with the simple and pedagogical example. This framework was used to design 18 levels of the LG "Les Cristaux d'Ehere" and produced full and filtered Petri nets of these levels automatically. On average, the Petri nets produced by this way are composed of 22 places, 19 transitions and 59 arcs. The most complex Petri net produces more than 127 000 game states.

References

1. Erik Harpstead, Christopher J. MacLellan, Kenneth R. Koedinger, Vincent Aleven, Steven P. Dow, and Brad A. Myers. Investigating the solution space of an open-ended educational game using conceptual feature extraction. In *Proceedings of the 6th International Conference on Educational Data Mining, Memphis, Tennessee, USA, July 6-9, 2013*, pages 51–58, 2013.
2. Amel Yessad, Pradeepa Thomas, Bruno Capdevila Ibáñez, and Jean-Marc Labat. Using the petri nets for the learner assessment in serious games. In *ICWL*, pages 339–348, 2010.
3. J. L. Peterson. *Petri Net Theory and Modeling of Systems*. Prentice Hall, Reading, Mass, 1981.
4. Jiacun Wang. *Chapman & Hall/CRC Computer & Information Science Series*, chapter Petri Nets for Dynamic Event-Driven System Modeling. 2007.