MOLPP, a New Generic Approach to Design a System for Monitoring Player Progress in Serious Games

Yassine El Borji
LIROSA, Faculty of Sciences, Abdelmalek Essaâdi University, Tétouan, Morocco

Mohamed Khaldi
LIROSA, Faculty of Sciences, Abdelmalek Essaâdi University, Tétouan, Morocco

ABSTRACT

The growing interest in the use of Serious Games in education raises many researches questions and involves some concerns, particularly the ability to track and understand the path of a learner - player as well as his progress. Moreover, their integration into the learning process remains limited because they do not provide effective features for monitoring learner / player progress in order to show them the consequences of their decisions. Thus, we propose a monitoring system called MOLPP (Monitoring Learner – Player Progress in Serious Games) which adopts a new generic approach based mainly on Anderson's ACT-R theory. The MOLPP system implements the Model Tracing diagnostic methodology derived from Anderson's ACT-R theory to analyze learner-player interactions from the Serious Game environment and match these interactions with the performance model, in order to generate a corrective intervention pertaining to the evaluation mode in the form of immediate feedback by complying with all the production rules of the performance model.

Keywords

Serious games, Learner-Player tracking, E-assessment, In-game interactions, Anderson's theory, ACT-R, Production rules, Tracing Model, multi-agents systems.

1. INTRODUCTION

The Problem Based Learning (PBL) approach has been used in the development of a sequence of professional courses for the degree of Computer Engineering at the authors' university [1]. Team-based projects form major components of these courses. All the projects follow the CDIO (Conceive, Design, Implement, Operate) context of engineering education that is being adopted by considerable computer and engineering departments world-wide.

For a long time, actors in the educational field have tended to ignore video games as a source of education. Today, the rapid development of the games industry, their immense effect on the new generation and the existence of certain characteristics within them that may be beneficial for educational purposes have led to an increase in the interest shown by the educational community. Today, serious games play an increasingly important role in the field of education, they have reached a certain maturity to become a possible alternative to traditional methods of learning which is often seen as restrictive and boring by learners. Despite the contribution and potential of serious games in the education, their integration in the learning process still limited, this is due to the lack of tools and features of tracking and monitoring the learner / player progress during a game session except perhaps an overall score, far enough to reflects their progress in the serious game environment.

The present article proposes a generic approach for the monitoring and the assistance of the learners - players during their progression by taking advantage of their interactions with the environment of Serious Game. These interactions can generate a huge amount of valuable information, in case they are available and presented in a consistent way, they can be used with different purposes helping to understand the behavior of the learner - player and render account of his progress.

This approach analyzes the interactions of the learner - player from the Serious Game environment by following a posteriori process based on the modeling of the learner-player's activities in order to detect, explain and characterize the errors in his path. We use Anderson's ACT-R learning theory based on the detection of deviations from an expert problem-solving model while respecting a set of production rules. Once a discrepancy is detected, a corrective intervention is generated in the form of immediate feedback.

The solutions brought by our approach are gathered in a system called MOLPP (Monitoring Learner / Player Progress in Serious Games) intended to be coupled with the Serious Game environment by offering a set of functionalities allowing to monitor the learners - players progress. A system that is mainly based on the implementation of the model tracing methodology derived from Anderson's ACT-R theory [1].

2. IMPORTANCE OF MONITORING LEARNERS / PLAYERS

There is a great need for the current virtual environments used in an educational context to demonstrate evidence of the learner's progress in his training, in order to evaluate his performance as well as to provide the necessary help if there is has a deviation is detected or a problem in solving a problem.

The monitoring of learners is an essential function in any Open Distance Learning (ODL) system. This is an essential factor for the success of such a training, because it allows to follow the learner during his progression which consequently allows to raise the problem of the feeling of isolation of the learner facing the machine [2]. In addition, it allows the identification of its needs in terms of pedagogical support and thus to establish a better relationship between the learner and his training.

In this paper, we are particularly interested in the monitoring of the learners - players in the Serious Games, due to their pedagogical maturities and the role they play in the educational field, which becomes more and more important. This need has been proven in a previous work whose goal was to contribute to the knowledge of Serious Games and promote their use in the educational field. We presented a comparative study which allowed to develop a tool for the evaluation of the quality of the Serious Games intended to be used in the education by examining their technical aspects, playful and pedagogical [3].

3. A MONITORING APPROACH BASED ON ANDERSON'S ACT-R THEORY

3.1 Essentials on the Anderson ACT-R Model

Anderson's model is built around a theory of cognition based on assumptions about the organization and acquisition of complex cognitive skills [4]. A theory that focuses on the representation of knowledge in a learner's memory and how this knowledge allows him to produce behaviors adapted to the structure of his environment.

We distinguish two types of knowledge in this theory: declarative knowledge and procedural knowledge [5]:

- Declarative knowledge concerning knowledge formalized in the form of a network comprising nodes (concepts) and arcs (associations between concepts), also called semantic knowledge, elementary units of information storage represent them.
- The procedural knowledge that concerns the "know-how" is stored in a specific memory, translate the knowledge associated with the action. The specificity of the productions is that, contrary to the declarative memory knowledge that can be added, the procedural components can be integrated in the procedural memory only by the action, ie by "doing". According to Mayers, a procedural knowledge is acquired in three stages: first, the acquisition of this knowledge in declarative form, then its interpretation to produce the behavior and finally the transformation of this interpretation into a compiled form [Mayers, A. , 1997] ;

In its original version, the ACT-R architecture offers a relatively complete set of principles for reporting on the construction and organization of the cognitive system. The construction of a knowledge processing and analysis system based on Anderson's ACT-R theory must be guided by the set of principles described below [1], [6]:

- Principle 1: Represent the model (or behavior) of the learner as a set of production rules <If condition Then action> ;
- Principle 2: Highlight the hierarchical goal structure underlying problem solving;
- Principle 3: Teach in a problem solving context (skills are acquired through action) ;
- Principle 4: Foster an abstract level of knowledge representation for problem solving. that is, the production rules must be abstract and can therefore be applied to situations that are sometimes very different from the original situation;
- Principle 5: Minimize the working memory load (to avoid errors due to loss of working memory) ;
- Principle 6: Respond immediately to errors ;
- Principle 7: Adapt the teaching to the student's expertise ;
- Principle 8: Facilitate resolution by successive approximations ;

In summary, Anderson's basic model assumptions are:

- Cognitive functions can be represented by production rules;
- The learning mechanism is reduced to the transformation of the declarative knowledge taught into a procedural knowledge through the execution of the tasks;
- The methods of intervention in case of error;

3.2 Production systems

Generally, production systems are the basis of most existing cognitive modeling in problem solving. These systems have been well adapted to simulate problem solving strategies [7]. They make it possible to manage the set of knowledge expressed in the form of rules of production thanks to an inference engine outside knowledge. The transition from one production rule to another occurs when the actions produced by the triggering of a production create the necessary conditions for the activation of another production. Therefore, these systems also called rule-based system, use sensors to perceive information from the environment and / or the internal state of the virtual entity; these perceptions are taken into account to select the next action to perform. In these systems, a behavior corresponds to a rule in the form conditions => action, so the action to be performed will be chosen from the set of actions whose conditions are satisfied by the perceptions coming from the sensors.

Production systems are mainly characterized by automaticity, the learning of which consists of a modification of the memory of rules, adding, deleting or modifying rules. The benefits of representing knowledge in the form of rules are multiple:

- Knowledge is organized in a modular way, making it easy to add and delete rules;
- The given solutions are explained by the list of applied rules, the solutions are then verifiable and false rules can be detected;
- Probabilistic reasoning is possible by adding a probability coefficient to each rule;

The architecture of the production systems takes into account two different types of long-term memory, a declarative memory and a procedural memory. These two different types of memory interact with each other through a working memory, as shown in the figure below:

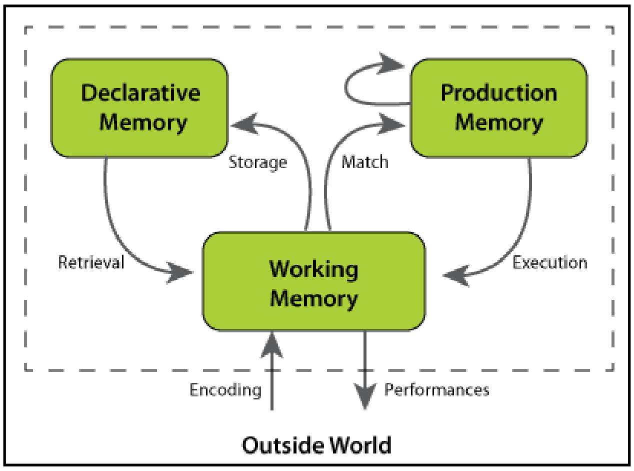


Fig 1 : Anderson ACT-R theory architecture [1]

There are two other important concepts in production systems: Pattern matching is the process that determines whether one of the production conditions is met by the current state of working memory. The second concept is conflict resolution; it is the process that determines the output that should be applied if several production rules are applicable.

The production systems in our approach operate by comparing the contents of the working memory and the "IF" part of a production rule of the procedural memory by consulting the declarative memory, which will be represented by an ontology on the domain addressed by the Serious Game in order to recover the information concerning the triggered production.

C. Structuring of production rules

As mentioned earlier, production systems, also known as rule-based systems, are used to manage a set of knowledge expressed in the form of production rules. This technique of representation of knowledge consists of defining a set of production rules that model the knowledge under consideration [8]. These rules have four characteristics:

- They are modular;
- They have an abstract character;
- They are structured in goals;
    - They have an asymmetry between conditions and actions;

For our approach, the behavior of a learner - player in a Serious Game is a rule in the form of: "If .... So ... ", which link a set of conditions to a set of actions. The first part of the "condition" rule is expressed by a logical predicate that must be true for the action to be triggered; the second part "action" is an executable part of the rule that indicates changes to add to the knowledge base. In short, the major objective of representing the activities of learners - players in the form of a set of production rules is to symbolize the characteristic features of their behavior.

To implement the diagnostic methodology called model tracing for a Serious Game that aims to educate children about road safety and sustainable mobility, it is first necessary to implement the performance model.

The Serious Game in question, called “Safety and Mobility for All”, is an extension of Renault's road safety program, launched in 2000 - aimed at achieving the UN's goal of halving road accidents in Europe by 2020. Based on two young characters, Lea and Jules, the game focuses on the pair of organizing different trips through role-playing exercises, during the game the kids are introduced to real-world considerations such as road signs, traffic directions, weather conditions.

We want to create a production rule whose purpose is to teach children that they must always cross the road through the pedestrian crossing. This rule will be triggered in a scenario where the learner-player chooses a place away from the pedestrian crossing to cross the road.

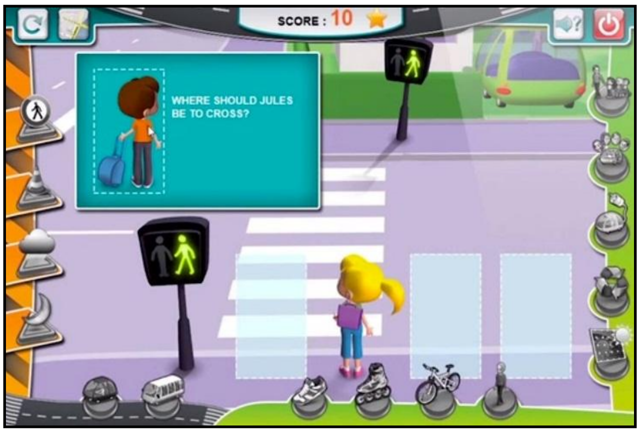
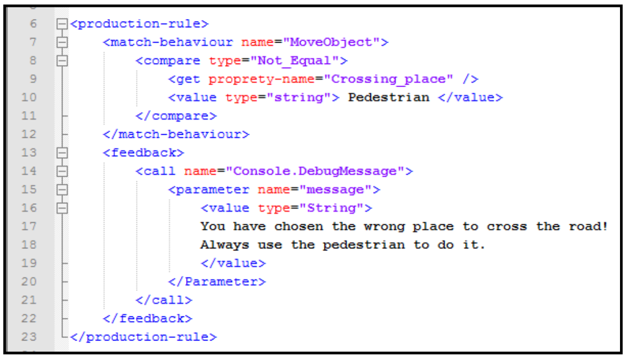


Fig 2 : Screenshot of the Serious Game "Safety and Mobility for All"

The fact that the learner-player chooses a place away from the pedestrian crossing will trigger the proper production rule for this action by generating immediate feedback in the form of a corrective intervention to keep it on track. Rule number 96, for example, can be described as follows:



```
<production-rule>
    <match-behaviour name="MoveObject">
        <compare type="Not_Equal">
            <get proprety-name="Crossing_place" />
            <value type="string"> Pedestrian </value>
        </compare>
    </match-behaviour>
    <feedback>
        <call name="Console.DebugMessage">
            <parameter name="message">
                <value type="String">
                You have chosen the wrong place to cross the road!
                Always use the pedestrian to do it.
                </value>
            </Parameter>
        </call>
    </feedback>
</production-rule>
```

Fig 3: An example of a production rule to teach students that they must always cross the road through the pedestrian crossing

4. ARCHITECTURE AND IMPLEMENTATION OF THE “MOLPP” SYSTEM

4.1 General architecture of MOLPP system

Our system is composed of several interacting units. Each of these units supports a specific task, the realization as well as the coordination of all these tasks allows to obtain a monitoring environment with a capacity of evaluation of the behavior of the learners – players.

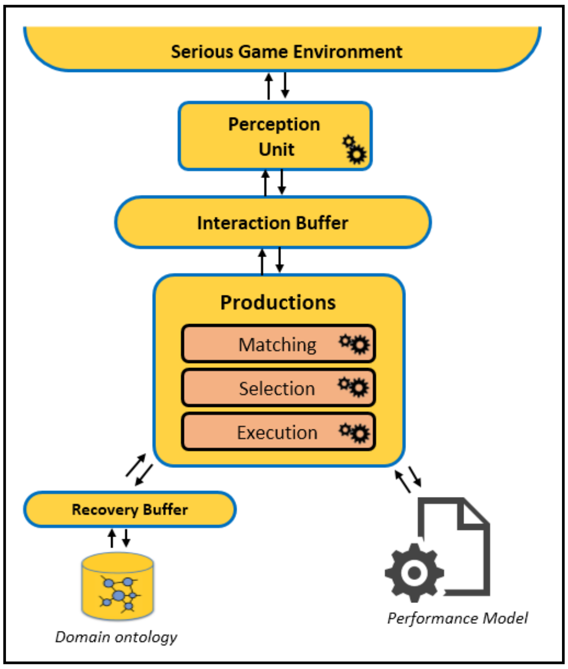


Fig 4 : General architecture of the "MOLPP" system

The architecture of our system incorporates a perceptual unit that serves as a listener on the learner-player's interactions with the serious environment in order to gather the information needed to analyze and diagnose his errors and thus generate an appropriate corrective action. This unit will be associated with a buffer by allowing it to communicate with other units via the information available in the respective buffers associated with them. The MOLPP architecture integrates other mechanisms such as the declarative memory unit which will be represented in our approach by an ontology on the domain tackled by the Serious Game, it plays the role of a knowledge base in which the knowledge will be stored as a network of nodes (concepts) and arcs (associations between concepts). An ontology whose main purpose is to represent the knowledge base of a Serious Game from the semantic web.

Our approach also incorporates a procedural unit (also called Productions) associated with a performance model that incorporates action / condition production rules, where the condition specifies what must be true for rule enforcement, and where action specifies the set of actions to execute if the rule is applied [9]. Indeed, the main objective of this procedural unit is the interpretation and analysis of the learner / player's activities in order to detect the gap in his / her career compared to the performance model.

4.2 Implementation of the MOLPP system

In order to meet the conditions allowing the Anderson model to integrate into a Serious Game, we opted for a multi-agent architecture. In what follows we highlight the organization of the multi-agent architecture of the MOLPP system. An architecture that implements the Model Tracing method derived from the ACT-R theory proposed by Anderson. The architecture of the various agents of the MOLPP system can be grouped into two distinct categories of agents that put forward their functions: perception agents and analysis agents:

✓ **Perception agents**

The perceptual unit serves as a listener on the interactions of the learner - player with the Serious Game environment. The « Perception_Agent » analyzes the action of the learner-player from the Serious Game environment and builds the «ActionEvent», which represents an instance of an Action. It is composed of a class of actions, the target object and possibly a parameter. The interaction buffer passes the constructed ActionEvent to the "Matching_Agent" of the production unit.

✓ **Agents of the Productions Unit**

The production unit consists of three Agents allowing the interpretation and analysis of the actions of the learner-player in order to detect the gap in his career automatically and in real time.

The "Matching_Agent" agent is associated with a performance model that includes action / condition type production rules, where the condition specifies what must be true for the application of the rule and the action part specifies the set of actions to execute if the rule is applied. Based on the "Pattern Matching" concept, the "Matching_Agent" agent makes it possible to compare and project the ActionEvent on the production rules of the performance model and to determine if one of the production conditions is respected.
After the "Pattern Matching" that uses the first part of the production rule called "Match-Behavior", it is the agent "PR_SelectAgent" that takes care of selecting the production rule to trigger that corresponds to the behavior of the learner and satisfies perceptions from the perception unit in “ActionEvent”.

The production unit also deals with the execution of the triggered production rule. The agent "PR_ExecuteAgent" uses the second part of the production rule called "Feedback" in order to apply it and generate the immediate feedback - Messages or Actions - that conforms to the action of the learner / player. The agent "PR_ExecuteAgent" communicates via the recovery buffer with the ontology on the domain addressed by the Serious. An ontology that represent the Knowledge Base (Game Knowledge Base) of Serious Game from the semantic web, in order to retrieve the information concerning the production triggered to help generate comprehensive and punctuated feedback with explanations to help the learner / player understand the causes of his gap.

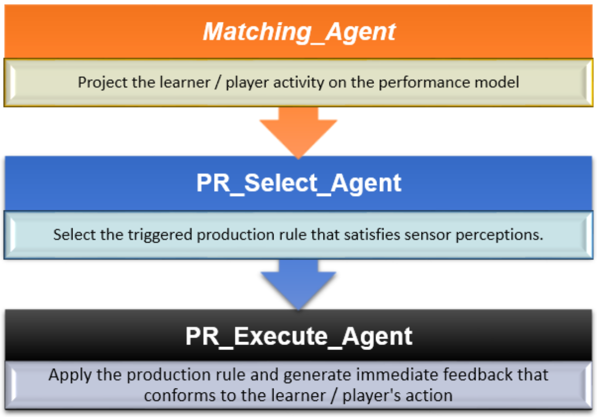

Matching_Agent
Project the learner / player activity on the performance model
PR_Select_Agent
Select the triggered production rule that satisfies sensor perceptions.
PR_Execute_Agent
Apply the production rule and generate immediate feedback that conforms to the learner / player's action

Fig 5 : Analysis process in the production unit

✓ **Domain ontology**

The domain ontology that we propose is derived from a complete ontology called Ludo allowing the modeling, the creation and the representation of a Serious Games and its elements to know the graphic part, the rules of the game, the base of the knowledge, the profile of the players and the context of the game [10].

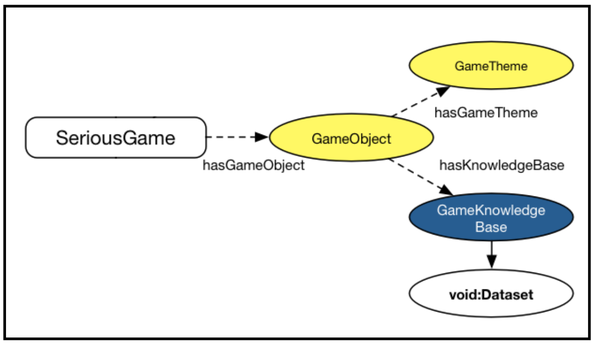


Fig 6 : Domain Ontology

We are particularly interested in the representation of the Game Knowledge Base of a Serious Game from the Semantic Web. A representation that can model Serious Games that must communicate with one or more sets of data from the Web, mainly to:

- Retrieve useful information that can be represented in the game through virtual objects;
- Store and access the game settings;
- Access to the profiles of the players;
- Generate additional content or resource recommendations for the player;

In the domain otology above, the “GameKnowledgeBase” class is defined as a subclass of “void:Dataset” and property “hasKnowledgeBase” is defined with class “GameObject” as domain and “GameKnowledgeBase” as range [10].

✓ **Multi-Agents architecture Multi-Agent Architecture of the MOLPP System**

The figure below illustrates the general operation of MOLP agents as a whole. First, we have the learner-player who interacts with the Serious Game environment. These interactions generate an enormous amount of valuable information that can be used with different purposes to help understand the behavior of the learner - player and report on his progress. Agents communicate and collaborate with each other to analyze these interactions by implementing the Model Tracing diagnostic methodology derived from Anderson's ACT-R theory.

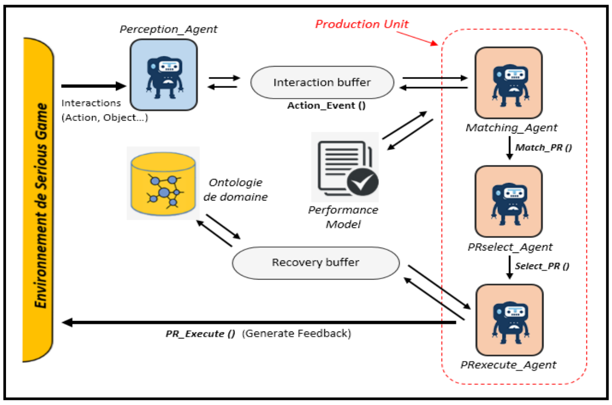


Fig 7 : Multi-Agent Architecture of the MOLPP System

5. CONCLUSION

In this paper, we presented a monitoring system called Monitoring Learner (MOLPP) intentedr to be coupled with the Serious Game environment by offering a set of functionalities to monitor learners - players and progress. A system that implements the diagnostic methodology called Model Tracing derived from Anderson's ACT-R theory.

The implementation of the "model tracing" method will allow the MOLPP system to support the detection of deviations from an expert model - performance model - of solving the problem by respecting a set of production rules. A corrective intervention is triggered immediately after the detection of a discrepancy in the form of immediate feedback - message or action - that conforms to the action of the learner - player.

However, beyond what has been presented in this paper, and in order to look at the effectiveness of the proposed approach, the Model Tracing diagnostic methodology implemented as part of this work needs to be Experienced and tested with Serious Games that adopts real-life teaching situations.

In the near future, we want to complete the process of implementing features that highlight the main contributions of the MOLPP monitoring system with Serious Games. We see that it is wise to implement modules of the system under the JADE platform, since it allows the creation of cognitive agents and allows the implementation of distributed systems.

In a first implementation phase, which concerns the implementation of the agents of the procedural unit, the communication via the buffers between the agents of the production unit has been established and experimented either between the agents or outside the production unit by consulting the performance model.

6. REFERENCES

[1] Anderson, J. R. (1993). Rules of the Mind. LawrenceErlbaumAssociates, Hillsdale, New Jersey

[2] Glikman, V. (2002, April). La" E-formation" entre globalisation des produits et pluralité des services. In Colloque international 2001 Bogues-Globalisme et Pluralisme. Université de Montréal-GRICIS.

[3] El Borji, Y., & Khaldi, M. (2014). Comparative study to develop a tool for the quality assessment of serious games intended to be used in education. International Journal of Emerging Technologies in Learning (iJET), 9(9), 50-55.

[4] Mayers, A. (1997). Miace: une architecture théorique et computationnelle de la cognition humaine pour étudier l'apprentissage (Doctoral dissertation, Université de Montréal).

[5] Grégoire, J. (1999). Que peut apporter la psychologie cognitive il l'évaluation fonnative et il l'évaluation diagnostique? In: L'évaluation des compétences et des processus cognitifs, B. Noël & Ch. Depover ed., Bruxelles, De Boeck, 1999, 17-33.

[6] Anderson, J.R., Corbett, A.T, Koedinger, K.R. & Pelletier, R. (1995). Cognitive tutors: Lessons learned. Journal of the Leaming Sciences, 4,167-207.

[7] Klahr, D., & Robinson, M. (1981). Formal assessment of problem-solving and planning processes in preschool children. Cognitive Psychology, 13(1), 113-148.

[8] Kass, R. (1987). The Role of User Modelling in Intelligent Tutoring System. Rapport technique, Moore School, Université de Pennsylvanie.

[9] Anderson, J. R., & Lebiere, C. J. (2014). The atomic components of thought. Psychology Press.

[10] Rocha, O. R., & Faron-Zucker, C. (2015, October). Ludo: An Ontology to Create Linked Data Driven Serious Games. In ISWC 2015-Workshop on LINKed EDucation, LINKED 2015.