Virtual Learners Behaviours in Educational Business Games

Nicolas Manin, Sébastien George, Patrick Prévôt

Fichier envoyé pour la publication :
Manin N., George S., Prévôt P., Virtual Learners Behaviours in Educational Business Games, *Lecture Notes in Computer Science*, Innovative Approaches for Learning and Knowledge Sharing, W. Neidj and K. Tochtermann (Eds.), Springer Berlin / Heidelberg, ISBN 978-3-540-45777-0, Vol. 4227, 2006, p. 287-301

Using Virtual Learners' Behaviours to Help the Development of Educational Business Games

Nicolas Manin, Sébastien George, Patrick Prévot

ICTT Research Laboratory, INSA de Lyon
21 avenue Jean Capelle, 69621 Villeurbanne Cedex, France
Nicolas.Manin@laposte.net
{Sebastien.George, Patrick.Prevot}@insa-lyon.fr

Abstract. Educational business games can be of a great help for initial and continuous learning, in the engineering field among others. These games have as major asset the possibility for learners to intervene or act as they wish. Nevertheless, there are numerous possible paths in such games. Designing the appropriate game structure and defining the optimum parameters so as to achieve the pedagogical objectives of business games are not straightforward. Presently, only users feedback allows the adjustments required and the appropriate structure modification needs to be identified. Waiting for feedback may take months and leave users a negative appreciation of the product. In this project, our aim is to conceive a "Simulator of Learners' Behaviour" to ensure the pedagogical quality of educational games. In order to provide realistic players' behaviours, mathematical and economical methods are used.

1 Introduction

During the designing of a computer environment for educational purposes, it is impossible to consider all the possibilities of learners' behaviours. Educational games are so complex that adjusting and finalising them are particularly difficult. It is thus interesting to possess a system that simulates learners' behaviour so as to statistically evaluate the performance of a game without having to wait for an extensive user feedback that could take several months. The objective of this research work is to design a generic simulator of learners' behaviour that can be instantiated for different business games.

In this paper, we shall firstly present educational business game principles and the context of our work. We shall then define virtual learners and a method to simulate real learners' behaviour by using metadata and mathematical methods. At last, we suggest a model named SIMCA which can be adapted to several games, and describe its framework and its functioning.

2 Educational Business Games

Educational business games are pedagogical applications that teach people several aspects of the functioning of a company, of a business or industry. During the past fifteen years, the ICTT laboratory of the *INSA de Lyon* has gathered a lot of experience in the designing of business games [1], [2], [3]. Most of these environments have been developed as group projects by 4th year students (master level) of the Industrial Engineering department under the supervision of research lecturers of the laboratory. These business games have then been used for training in this department and others at INSA. The ICTT laboratory can thus directly gather a large quantity of feedback.

The use of educational business games make up a relevant training method for engineering students. The objective of these games is to establish a virtual environment in which we can artificially recreate on a smaller scale (time, space and actions) the conditions that allow learners to find themselves in a context similar to the situations they will face in their professional life. Business games cover a wide range of subjects (finance, commerce, industrial maintenance, collaborative problem solving…). As a variant of role-playing games, business games confront learner groups, usually associated to different companies, to each other. Endowed with specific powers or skills (tools, competences, protections…), players must take up the challenge and attain goals such as decreasing the stock level, increasing productivity, winning market shares, etc.

2.1 Educational Business Games Models

As researchers, one of our objectives is to situate the development of business games in an organized and structured approach. The goal is to define a common root that make it possible to speed up the development of specific games and at the same time benefit from a common infrastructure that allows designers to create, refine and extend these games without having to start from scratch each time. Our approach is generic and aims at the identification of invariants in order to define a design technology which creates a genuine workshop for configuration and as development help for this type of pedagogical products. The identification of invariants has led us to elaborate the following models: the conceptual, the dynamic, the pedagogical and the computer models.

The Conceptual Model. It is based on three fundamental elements in the game structuring:

- the attractor (i.e. "educational bait"): a challenge that must be taken up by the learner,
- the case: real-life contextual situation (defining the rules, environment …),
- cognitive places: knowledge, concepts, know-how, behaviour, competences… which are objects of the training. The learner is not necessarily conscious of these places but he is compelled to pass these places to take up the challenge proposed to him;

The Dynamic Model. It consists in cutting up the scenario into phases:

- discovery phase,
- analysis and diagnosis phase,
- strategy construction phase,
- decision-making and implementation phase,
- assessment and evaluation phase;

The Pedagogical Model. It integrates the animator role as from the original design of the game:

- briefing sessions, to start the game or phases,
- debriefing and sharing sessions for analysis,
- multimedia sessions that alternate the conceptual and action sections;

The Computer Model. It is based on a multi-agent approach for the following reasons:

- In multi-agent systems, as in games, acts are the structuring elements of the system,
- The cooperative aspects of educational business games dictate a distributed vision,
- Business games must be able to adapt themselves to modifications to the structure or environment brought about by the author.

2.2 Research Issues on Business Game Designing

During the designing of an educational business game, it is impossible to predict all the possibilities in learners' behaviour (navigation, activity choice, answers given to questions, course or route taken…). The richness of this type of educational games makes their realisation difficult (in terms of parameters, game structure, need for interaction…). The only way of creating such a game is to make actual learners test the game and then correct it according to feedback. From experience, we know that this process takes about two years to achieve a finalised version of the game (from a pedagogical point of view). This long testing time causes the lost of product credibility. This happens particularly if the learners having the "appropriate behaviour" (in terms of learning objectives) are not the ones who are rewarded…

Our research objective is thus to suggest models and tools for testing the games prior to their use. To sum up, we aim at designing a simulator of learners' behaviour to test the game's parameters. This simulator can be considered as a test bed sufficiently universal to test thoroughly the game progress with virtual learners (with random actions and events) so as to statistically evaluate how far the pedagogical objectives have been achieved. This solution will be used in conjunction with the user-centered and participatory design approaches. So the final solution will be a sort of hybrid approach that involves both virtual players and working with real users.

The closest work on which we can base ourselves are about virtual players [4], [5] and companions [6], [7], [8]. As in these works, we must define players'/learners' behaviour and a mechanism to retrieve information about what happens all through the game session so that the simulator can act consequently. However, the needs of our learners' behaviour simulator are quite different. In particular, we do not want to create a simulator dedicated to a specific game but to design a simulator for a class of business games.

2.3 Study of Two Business Games

To come up with a model for the simulator, two educational business games have been studied and analysed. The "maintenance game" (in the field of industrial maintenance) and the game called "Garde à Vue" (French for "police custody" – concerning the law field) have been chosen. These two games, both event-based, are nevertheless different enough in their objectives and the knowledge to acquire. This study has contributed to our researches on desired features, the constants and some of the parameters necessary to design a learners' behaviour simulator for educational business games.

These two games are board games with well-defined steps or "squares" and events (with various consequences) that take place at each step. At any moment, in order to play, one must be able to know his or her location (which step he/she is at), his or her game score or the amount of money he or she has, or the available objects or accessories. Hence, the simulator must be aware of the values of variables that indicate the status of the game and that of the player/learner. As for interaction, there are situations where choices are possible (multiple-choice questions, purchases, sales, etc.). These choices of course have an impact on the rest of the game. To be able to make theses choices, the simulator must include a game strategy. That is, he must know the player's objectives and the types of player behaviour to obtain these objectives. For example, in the two games considered, there are means of avoiding risks and decreasing losses or sanctions. There are also actions that have long-term or short-term effects. In the two games, there are both random stages/steps and default ones. In these cases, the simulator passively undergoes what happens to it and is not given any choice. The values of variables and the rest of the game change subsequently.

3 Defining Virtual Learners

The objective is to define virtual learners able to take decisions in business games like human learners. In this part, we will present how to simulate their behaviour, with the combination of two methods: one based on the strategic sense of the game actions, and the other based on the game events and their influence on the objectives.

3.1 Strategies-based Behaviour

We have defined a method to simulate learners' behaviour using game strategies and player profiles. We explain these strategies and profiles below before discussing how behaviours are simulated.

Game Strategies. Strategies are used to annotate actions of the games with behaviour criteria. We have been able to identify the following strategies partly by observing the use of educational business games by actual learners. Some can coexist (but at different intensities or proportions) while others are totally opposed one to another.

- favoured *term* (long, medium or short),
- type of *interaction* with the other players (competition, collaboration…),
- need for *adventure* (explore – active , discover – passive…),
- *risk taking* (maximum, average or minimum),
- *enthusiasm* (maximum, average, minimum – "endure" the game),
- *time* spent on the game/*duration* of the game (playing for the longest time or trying to end the game before the other players),
- *desire to win* (winning at all cost or doing everything to lose).

Player Profiles. In business games, some players seem to use similar strategies in the way of solving problems: some are careful while others take more risks; some are more helpful than others, etc. Such behaviours are characteristic of some well-known profiles:

- the *aggressive* player can do anything in order to win the game;
- the *ambitious/calculating* player. Everything he does is motivated by victory;
- the *curious/inquisitive* player wants to learn and to explore the game levels;
- the *careless/thoughtless* player is one who has the necessary knowledge but who does not pay much attention to what he does;
- the *go-ahead/audacious* player does not have a strategy and does not care about risks;
- the *kamikaze* player does all he can to sabotage his chances of winning;
- the *neutral/average/standard* player does not take lots of risks but is not excessively cautious;
- the *nonchalant* player is not really a type of user but is rather a type of behaviour that we want to simulate. It can be that of a person who is not much involved in the game and who will make random choices or that of a program module that needs to test a game's reactions;
- the *hurried* player wants to have the maximum gains as quickly as possible;
- the *cautious* player is the one who takes the minimum risks.

Simulating Behaviour. Each virtual learner has a player's profile, and each game action is annotated with strategic marks. During the game, when a virtual player must choose between different actions, strategic marks of each solution are compared with the strategic preferences of his profile. Then, he can choose the action which is the most similar to his profile. For instance, a cautious-defined virtual player will choose safety actions, while an aggressive-defined opponent will choose instead hazardous actions. This is a method of behaviour simulation that can lead to varied, realistic behaviours. However, it implies that each action of the game has a strategic sense.

The quality of the simulated behaviours depends on the quality of the actions marks. Defining them is a difficult task, because the strategic value of an action can be differently interpreted by people. Therefore, these marks are also the expression of their creator's own judgment, thus the decision of virtual players could not be entirely objective. Another problem is that the strategic value of an action cannot be judged alone, but the whole context and its consequences must be considered. For example, an action is dangerous when it can lead to bad situations. If there is no bad situation, then the same action should no more be considered dangerous. Details on the simulation of behaviours can be found in the article George *et al.* [9].

3.2 Events-based Behaviour

It appears that this strategic-based behaviour is centered on the learner's psychology. Virtual players take decision following their psychological preferences, in a reactive process that isn't directly linked with the winning conditions. Do aggressive-defined

Table 1. This table matches game strategies and player profiles. The parameters on the horizontal axis represent the strategies and the standard profiles are on the vertical axis. The sections marked with a cross specify the strategies corresponding to each learner or player type.

	Player strategies																				
	Term			Interaction			Need for adventure			Risk-taking			Enthusiasm			Time/ Duration			Desire to win		
Player profiles	long	middle	short	competition	collaboration	independent	maximum	average	minimum	maximum	average	minimum	maximum	average	minimum	maximum	average	minimum	maximum	indifferent	none
Aggressive			X	X					X	X			X				X		X		
Ambitious	X			X			X					X	X				X		X		
Curious	X				X		X			X			X			X				X	
Careless			X			X			X		X			X			X			X	
Go-ahead			X	X					X	X			X					X	X		
Kamikaze			X			X			X	X					X		X				X
Neutral		X			X			X			X			X			X			X	
Nonchalant			X			X			X		X				X		X			X	
Hurried			X			X			X	X			X					X	X		
Cautious	X				X		X					X		X			X		X		

virtual players must have a suicidal behaviour in their actions, while sometimes the danger is so evident for human learners? Since the virtual learners are also players, they have to pay attention to game objectives.

Freud used to distinguish instinct and reason in the human psychology. For philosopher Bergson, in 1907, instinct and the intelligence represent two solutions of the same problem. We have thought about reproducing this duality in the decision making, providing virtual players instinctive (strategies-based) and rational (events-based) decision units.

Some Theoretical Concepts of Decision. To define a player's behaviour centered on the game events, we have looked at mathematical and economical works, such the Game Theory of John von Neumann and Oskar Morgenstern [10] or the Prospect Theory of Daniel Kanheman and Amos Tversky [11]. We will present here some concepts used in these famous theories that should serve to simulate the virtual player's decision-making process.

Utility Function. In most games, in order to win, the players have to maximize the values of specific resources (e.g. score, money), and/or to minimize other values (e.g. time). While attempting to achieve game objectives, players are attracted by positive events that tend to maximize or minimize their resources. So the evaluation of the variation of a resource in a game event can be expressed in term of attraction or repulsion, like the strategic preferences: this is the utility function. We will use specific utility functions to simulate players' choice between several solutions.

Expected Value. If X is a discrete random variable which can take x_1, x_2… values with p_1, p_2… associated probabilities, the formula of the expected value of X is:

$$E(X) = \sum_i p_i (x_i) . \quad (1)$$

Using this mathematical formula as a utility formula is a very simple method to have objective and evaluated comparisons of game solutions. For example, if a virtual player's objective is to maximize the gain, he calculates the expected value of gain of each solution, and then chooses the solution with the highest value.

However, using the expected value as the utility function is not a sufficient solution. The simulated behaviours are different from that observed on real players – what about the popularity of national lotteries in spite of negative expected gain values? Worst, they can lead to paradoxes, such as the Saint Petersburg's paradox.

The Diminishing Marginal Utility. The mathematician Daniel Bernoulli solved the Saint Petersburg's paradox by introducing psychology parameters. In this theory, the attraction for a gain is not only linked to its value, but also to its utility for the receiver according to his/her initial resources (i.e. the marginal utility). For example, 10 € have more utility for a player who has 10 € than for a player who has 1000 €. In the first case, the 10 € gain represents a 100% gain, and in the second one, only a 1% gain. The more resource someone has, the less s/he feels utility to have more: this is the diminishing marginal utility.

Risk Aversion. When someone plays a double-or-quits game, this player must choose between an assured gain and a lottery with the same expected value of gain. Experience shows that, in the majority of cases, he keeps the guaranteed gain: this is risk-aversion.

Introducing risk aversion to virtual player's profile will be a medium of giving them more realistic behaviour. To simulate the risk aversion, economics use convex functions such as the logarithmic function; however, since it is defined only for positive numbers, we can use another function, like the square root. So we modify the expected value formula to obtain the following utility formula:

$$U(X) = \sum_i p_i \sqrt{(x_i)} . \quad (2)$$

The utility of a 100 € gain X_1 is $U(X_1) = \sqrt{(100)} \approx 10$, whereas the utility of the gain X_2 of the corresponding double-or-quits lottery $L = \{(½; 200); (½; 0)\}$ is $U(X_2) = ½ \sqrt{(200)} + ½ \sqrt{(0)} \approx 7.071$. So the certain gain is more attractive for a risk-averse person.

Loss Aversion. It is a psychological concept which explains that loss-averse people are more sensible to a loss than to a gain, even if they have the same value. Like the risk aversion, adding loss aversion in the mechanism of decision-making is an opportunity to make virtual players' behaviour more realistic. To simulate it in the virtual players' decision, we use the following utility formula that inserts a multiplier coefficient to all losses' values:

$$U(X) = \sum_i p_i (y_i) + \sum_j q_j (\lambda z_j) . \quad (3)$$

The y_i are the positive values (gain) and p_i their probabilities, and the z_j are negative values (loss) and q_j their probabilities. λ is the loss aversion coefficient: $\lambda > 1$ for a loss aversion, $\lambda = 1$ for a loss neutrality, and $0 < \lambda < 1$ for a loss attraction.

Simulating Behaviour. All of these concepts could be used to simulate human players' behaviours. We combine each one of them in an ultimate utility function that will be used to calculate attraction/repulsion of game events. In this function, y_i are the positive values (gain) and p_i their probabilities, the z_j are negative values (loss) and q_j their probabilities, and λ is the loss aversion coefficient.

$$U(X) = \sum_i p_i \sqrt{(y_i)} + \sum_j q_j \sqrt{(\lambda z_j)} \ . \qquad (4)$$

3.3 Coordinating Decisions

Virtual players, with both instinctive and rational decision units, dispose of two distinct solutions for the same problem. Decisions can be compatible, but they can also be opposite. Thus, players need an arbitration process to choose only one solution.

To avoid this problem, each decision unit won't provide only the most attractive solution, but the list of all solutions and their probabilities of choice, depending on their attraction or repulsion for the player. Then, an arbitration unit combines the two probabilities of each solution in a unique probability. An advantage of this probabilistic method is to define varied behaviours.

4 The SIMCA Model

We have defined a method to simulate virtual learners' behaviours for business games. Now, in this part, we describe the global functioning of the simulator. The designed model is named SIMCA ("*SIMulateur du Comportement d'Apprenants*" which means "simulator of learners' behaviour").

4.1 Global Functioning of a Simulation

The simulator is composed of an inference engine that contains functioning rules and parameters that describe behaviour characteristics of virtual players. Business games on which we experiment the simulator have a game engine separated from the graphical interface. In this way, the simulator does not use Human Computer Interface of business games but intercepts the messages coming from the game engine and sends some commands and actions directly. The global functioning of the SIMCA model is depicted in figure 1.

To use the simulator on a particular business game, the game administrator has to list all the important data of the game that will be used by the virtual learners (variables representing choices, game states, scores and players progress). This list is then given (1) to the simulator administrator who associates the parameters of the simulator with the game variables (2).

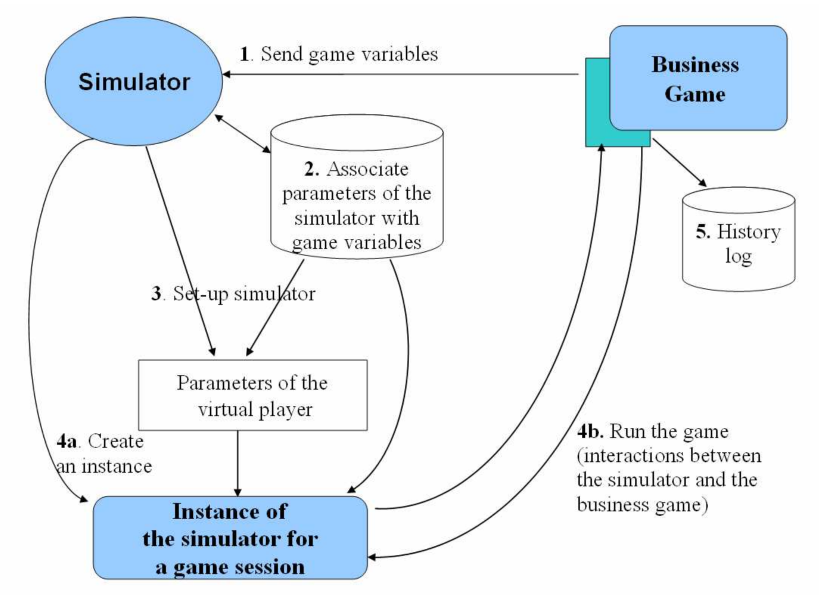

Fig. 1. The SIMCA Model.

Before launching the simulator, the administrator chooses the parameters of the virtual player for a game session (3). He can choose among a set of existing players' profiles, those defined in the table 1, or specify a particular player profile by configuring the different strategies (interaction, need for adventure, risk-taking, etc…). He also defines the player's objectives policy.

Finally, an instance of the simulator is created from the game information and for a virtual player profile (4a). A game session could then be run. Some messages are then exchanged (4b) between the game (choices, events …) and the simulator (decisions, actions …). During the session, the simulator engine will interact with the virtual players when they will have to take decisions in order to reach the set goals and to follow their strategies.

Exchanged events are recorded in a history log (5) that aims to help the game designer to identify necessary modifications before the use of the game in real situation with learners. It should be noticed that the actual model does not take the interaction between learners into account.

4.2 Virtual Learners in the SIMCA Model

Virtual players are an essential component of the SIMCA model. We have previously defined their theoretical decision-making process. In this part, we describe how to integrate them and their decision process within the SIMCA architecture.

Metadata and Graphs of Actions. In the Game Theory, game trees are graphs in which nodes represent the situations, and edges the different moves. Decision takers use them, starting from the present situation (root), to choose the actions that can lead them to the best future situation. We want to use similar structures to represent the game metadata used by the virtual players.

We define graphs of actions as structures that contain both the strategy-related and event-related marks. Nodes and edges can be of different nature. There are four different kinds of nodes:

- *Decision nodes* (D-nodes) represent a decision situation, and signify a choice for the player. Their out edges represent all different possible actions. Each one is annotated with strategic marks related to the players' list of strategies (cf. part 3.1).
- *Events nodes* (E-nodes) contain a list of events in relation to objectives, in term of gain or loss (e.g. score: -50, time: +10). According to his objectives, the paths leading to some E-nodes will attract or repulse the player. They have no or only one out edge.
- *Random nodes* (R-nodes) represent a randomly-determined fork, for example a lottery. Their out edges correspond to the universe of possibilities, and each one has a probability.
- *Tests nodes* (T-nodes) represent a game or player's condition-determined fork. Their out edges correspond to the different possibilities of satisfaction of the condition, and each one has a list of conditions.

These graphs have an entrance node, always a decision node, since it represents the initial decision situation. The exits of graph are the events-nodes, as they are the final consequences of the chosen actions, but not all of them are exits.

One advantage of using graphs of actions is that virtual players take decisions with the metadata they contain. So a simulation can be run without the finished game, just with the data files of its graphs. That can be useful to test the game parameters before or during the implementation. Another advantage is that the game structure can be easily changed, just by replacing a graph file by another.

Virtual Learners' Decision. The decision-making process is the key of the simulation because it reflects the quality of the simulated behaviour. Each virtual learner has:

- a strategic profile: aggressive, ambitious, curious, careless…, or cautious.
- a list of objectives, in the form of a list of weighted resources/objectives, indicating their priority. Virtual players seek to maximize resources with positive weight (e.g. score), according to their value, and to minimize resources with negative weight (e.g. time). They are neutral to resources with null weight; this can be interesting to test only specific objectives of the game.

The virtual player evaluates the solutions of a problem starting from the D-node root of the corresponding graph of actions. Attraction( ) is the function that calculates the attraction of an action, node or event. The repulsion corresponds to a negative value of attraction. We say that the attraction of an action (a D-node's out edge) is equal to the attraction of the successor node. There is now two distinct evaluations of the solution, leading to two distinct decision attractions.

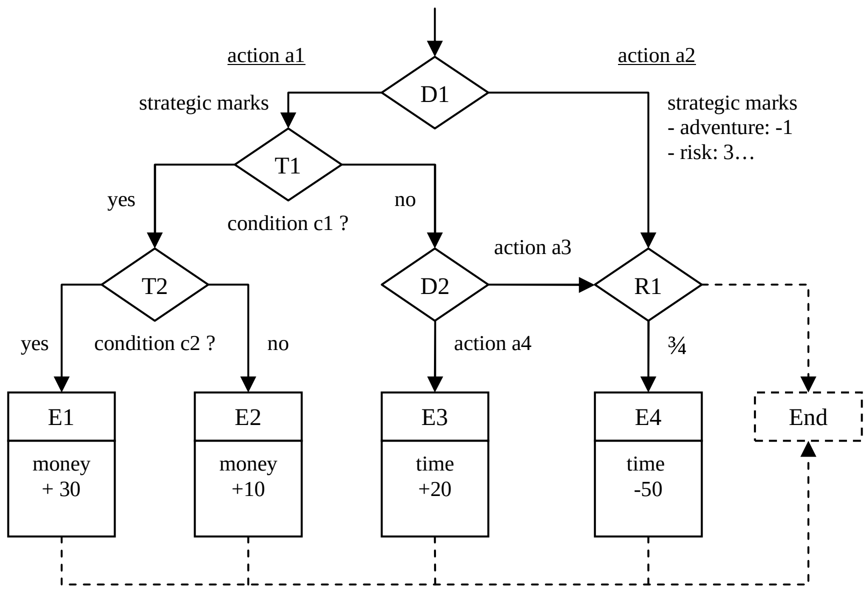

Fig. 2. An example of graph of actions. The virtual player who uses this graph is in a decision situation represented by the D1 decision node, where he can choose between the "a1" and the "a2" action. The first action leads to the T1 node and the other one will lead to the A1 node. D2 is another decision node, with "c" and "d" possible actions. T1 and T2 are test nodes representing respectively conditions "c1" and "c2". R1 is a random node; ¾ is an example of probability leading to E4 node. E1, E2, E3 and E4 are events nodes, with examples of events.

Instinctive Attractions. To express instinctive attractions, the virtual players use a simple comparison algorithm of the strategic marks. The marks of the starting D-node's out edges (i.e. possible actions) are all compared with the player's strategic preferences. If the strategy of the mark matches with the player profile, then the attraction will be positive; otherwise, it will be negative. All marks are compared one by one, and the sum of the attraction gives a global attraction value.

Rational Attractions. To express rational attractions, virtual players use a path graph algorithm. The attraction of an event is calculated using a utility function as previously defined. If the graph contains cycles, i.e. a situation leading to a previous one, then the algorithm could run in an infinite time. So it is important to define a limited number of covered nodes. We define the following recursive algorithm:

- attraction (E-node) = $\sum_i$ utility(event$_i$) + attraction (succ), where event$_i$ are the E-node's events.
- attraction (R-node) = $\sum_i$ probability(succ$_i$) * attraction (succ$_i$), where succ$_i$ are the R-node's successors.
- attraction (T-node) = attraction (succ$_i$), where succ$_i$ is the T-node's successor who completes all conditions of the node's out edge.
- attraction (D-node) = max(attraction (succ$_i$)), where succ$_i$ are the D-node's successors.

For example, from the node D1 in figure 2, the two possible solutions are the actions “a1” and “a2”. If the conditions “c1” and “c2” are both completed, then the attraction of “a1” is : attraction (T1) = attraction (T2) = attraction (E1) = utility (money, +30); and the attraction of “a2” is : attraction (A1) = ¾ attraction (E4) = ¾ utility (time, -50).

The Choice of a Solution. After 1) the strategic marks comparison and 2) the graph exploring, each algorithm gives to virtual players an evaluation of each solution, such as this example list (independent of the fig. 2):

Solution “A”: attraction = 0; solution “B”: attraction = 15; solution “C”: attraction = -5; solution “D”: attraction = 25; solution “E”: attraction = 10

What does the attraction mean in this example? The B, D and E solution, with positive attractions values are attractable for the players; the A solution, with null value is neutral; the C solution, with negative value, is repulsive. In this case, the choice of the solution will concern the first three ones. If there are no positive values, virtual players will choose between the neutral solutions. And if there are no null values, they will choose between the repulsive solutions.

The easiest way to select a solution is to choose the one with the highest attraction value, because it is the “best” (or at least the worst) one. But there are opportunities to build a more varied behaviour. For example, we can add a probability to each solution depending on its attraction value. So in the precedent example, the “D”, “B” and “E” solutions would have respectively 50%, 30% and 20% chance of being chosen, according to their attraction of 25, 15 and 10. The other solutions would have a 0% chance.

When the virtual player selects a solution, the simulator informs the game of the chosen action, the information is saved in the game log file, and the game session continues.

4.3 Implementation and Uses.

We have developed a prototype of simulator using java language, and XML data files, one for each graph of actions of the game. They are represented in XML, because it is an easy way to describe the graph structure. The game administrator uses an editor to create the graphs easily. The simulator administrator has then to reference the files, so that the simulator engine can find the graph corresponding to each decision situation. For example, he associates a graph to a board case of the game.

There is also one data file that contains all player profiles. The simulator administrator uses it to initialize the virtual players before the start of the game session.

Creating Graphs of Actions. Below is an example of an XML file that represents a graph of actions. The XML syntax is adapted to our process algorithm.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<graph>
  <Dnode name="D1">
    <outEdge name="Action1">
      <strategy name="risk">2</strategy>
      <!-- other strategic marks -->
```

```
        <succ type="Tnode ">T1</succ>
      </outEdge>
      <!—- other out edges -->
    </Dnode>

    <Rnode name="R1">
      <outEdge>
        <proba num="2" den="3"/>
        <succ type="Enode">E1</succ>
      </outEdge>
      <!—- other edges -->
    </Rnode>

    <Enode name="E1">
      <event name="money">-15</event>
      <!-- other events -->
      <outEdge>
        <succ type="END">END</succ>
      </outEdge>
    </Enode>


  <Tnode name="T1">
      <outEdge>
        <test type="=">
          <object>time</object>
          <value>100</value>
        </test>
        <!—- other tests -->
        <succ type="Rnode">R1</succ>
      </outEdge>
      < !-- other out edges -->
    </Tnode>

  <!-- other nodes -->

  </graph>
```

List of Player's Profiles. Example of the XML file that contains the list of player's profiles.

```
  <?xml version="1.0" encoding="ISO-8859-1"?>
  <profiles>
    <profile>
      <name>Aggressive</name>
      <strategies>
        <strategy id="risk">3</strategy>
        <!—- other strategies -->
      </strategies>
    </profile>
    <!—- other profiles -->
  </profiles>
```

5. Conclusion

The development of business games is delicate and limits very often their use. We have proposed in this article the base of a simulation model of learners' behaviour aiming at favouring the testing of business games in compressed time. This model, named SIMCA, is based on the definition of players' profile types and game metadata (graphs of actions), which ensures tests as comprehensive as possible. Technologies used allow an easy set-up of the simulator despite the existing differences in the development of business games. The SIMCA model leads to the implementation of a prototype that will be tested soon on two business games.

There are many futures directions of this work. First, in an incremental development approach, the actual model could be improved so as to refine interactions between learners. In the same direction, we aim at simulating more realistic behaviours by adding for example the change of strategies during a game session. Furthermore, the simulator runs currently in compressed time, which allows fast virtual testing. A real time mode could be added in order to simulate virtual players that can be able to play with or against human players.

Finally, this work is part of a global study on an integrated environment for the design of business games. Beyond the testing environment, future work will concern a complete software suite for the development and configuration of educational business games.

References

1. Akkouche I., & Prévôt P.: Cooperative distance learning-sessions management: A transportation model application to Role Distribution. IEPM'97: International Conference on Industrial Engineering and Production Management, Lyon, France (1997) 166-174
2. David B., Gibaud O., Roland J.-P., Tarpin-Bernard F., & Vial C.: Cooperative learning Systems: Extension of the micro-world concept. Telecommunications for Education and Training, Prague (1997) 48-54
3. Babari M., Mahi A., & Prévôt P. : Approche générique pour la conception et l'ingénierie de jeux d'entreprise multimédias coopératifs - Cas du jeu de la maintenance multimédia. Colloque international TICE 2000, Troyes, France (2000) 377-384
4. Bouzouane A., Dionne C., Stiharu-Alexe I., & Gagné D. : Jeu de rôle virtuel à base d'agents intelligents. 6ème journées Francophones d'Intelligence Artificielle et Systèmes Multi-Agents (JFIADSMA 98), Pont-à-Mousson, France (1998) Hermès, 147-161
5. Roge O., & Labat J.M.: Integration of Virtual Players into a Pedagogical Simulator. Computer Aided Learning In Engineering Education (CALIE 2004), Grenoble, France (2004) 227-232
6. Gazzaniga G., Morrone G., Ovcin E., & Scarafiotti A.: The "virtual" companion: a "real" help for educational hypermedia. International Conference on Educational Use of Technologies (ICEUT 2000), Beijing, China (2000) 363-366
7. Zapata-Rivera J.D., & Greer J.: Exploring Various Guidance Mechanisms to Support Interaction with Inspectable Learner Models. Sixth International Conference on Intelligent Tutoring Systems (ITS 2002), Biarritz, France, (2002) 442-452.
8. Burleson W.: Affective Learning Companions. Doctoral consortium in conjunction with the 7th International Conference on Intelligent Tutoring Systems (ITS 2004), Maceio, Brasil (2004)

9. George, S., Titon, D., Prévôt, P. : Simulateur de comportements d'apprenants dans le cadre de jeux d'entreprise. Conference Environnement Informatique pour l'Apprentissage Humain, Montpellier, France (2005) 389-394
10. Von Neumann, J., & Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press (1944)
11. Kahneman, D., & Tversky, A.: Prospect Theory: An Analysis of Decision under Risk. Econometrica, 47 (1979) 263-291