Applying Reinforcement Learning To ITS For Adapting Learning Situations

BENNANE A. ; MICHIELS I. ; MANDERICK B. & D'HONDT T.
abennane@vub.ac.be ; isabel.michiels@vub.ac.be ; bernard@arti.vub.ac.be & tjdhondt@vub.ac.be
2, pleinlaan
B-1050 Brussels
Belgium
Vrijie Universiteit Brussel

***Summary**: During the development of the intelligent tutors, several techniques were applied resulting from artificial intelligence and machine learning techniques. The goal is to develop a system, which imitates human behavior in these various cases and in fact in (1) the reasoning process and in (2) the level of interactivity with its environment.*
An intelligent tutor is composed of the following entities: (1) the domain expert (2) the tutor (pedagogical module) (3) the student model and (4) an interface of communication between expert - tutor on the one hand and student from the other.
The expert interacts with the learner according to the situation selected by the tutor. Following the action of the learner, the expert reacts to evaluate the behavior of the learner knowing that the action of this last was successful or not. In both cases, the expert presents an adequate feedback to the learner. The tutor observes the interaction between the learner and the expert and tries to direct the communication according to the objective to be reached and the abilities of the learner.
In this paper, we expose the components of interaction in an intelligent tutor and we discuss how reinforcement learning can be applied to a tutoring system in order to individualize and to adapt the generation of the learning situations.
***Key words:** learning Situation, Tutoring, Reinforcement learning, ITS, Agent, Environment.*

1. Introduction

Using communication and information technology for teaching and training is an idea, which dates at least from the beginning of the last century. The majority of researchers [BRUI97; DEP87] agree that the first step was taken by Pressy, with the machine he built in 1926. The objective of Pressey was to build a device which makes it possible to raise a question, and after the answer of the pupil, to communicate the thruthness of the answer immediately to him. It considers that the true work of the teacher is to concentrate on the activities stimulating the thought, to release it from the repetitive spots and to delegate the training exercises to a machine.
At the beginning of the Sixties, computer-assisted instruction (CAI) came as a true marriage of programming instruction and data processing [ALBE92].
The two most interesting aspect of CAI were (1) the individual character of the program of teaching: the learner, with the programmed machine, has the feeling to be with a system charged to provide him a particular course, adapted to his needs and his level; and (2) the interactive aspect which, if it is agreeably exploited, avoids the learner from sinking in passivity [LESC85]. The applications of CAI, with or without audio-visual dimension, were limited to the organization of information presented at the screen according to a preset sequence [DEP99]. In the Seventies, a limited number of researchers defined new ambitious objectives for CAI. They adopted the human tutor like that of the educational model and applied artificial intelligence techniques to it.
The objective of Intelligent Tutoring systems (ITS) is to engage the students in continuous activities of reasoning and to interact with the student based on a deep comprehension of its behavior [ANDE97].
The possibility of stocking and executing the teaching domain knowledge forms the keystone of intelligent tutorials. The representation of this knowledge makes it possible to not only treat the answers, but to penetrate in the process even of problems solving [DILL93]. One must be continuously aware of the fact that research on intelligent tutor systems is far from the ideal goal which is to create a completely autonomous system in its teaching reasoning [DEP99].
In this paper, we expose the entities of interaction of an intelligent tutor and we discuss how reinforcement learning can be applied to a tutoring system in order to individualize and to adapt the generation of the learning situations.

2. Components of the interaction unit

Recall that the pedagogical module is a set of rules and protocols that manages resources and interactive communication and follows the learner's needs.
The pedagogical module has to fulfill the following tasks:

- · Evaluate the learner's actions and determine the values of the transition parameters, the rewards, and the learning path;
- · Select the learning situations from the database and follow the orders of the evaluation unit and present it to the user;
- · Look for and send the rewards to the learner following the orders of the evaluation unit.

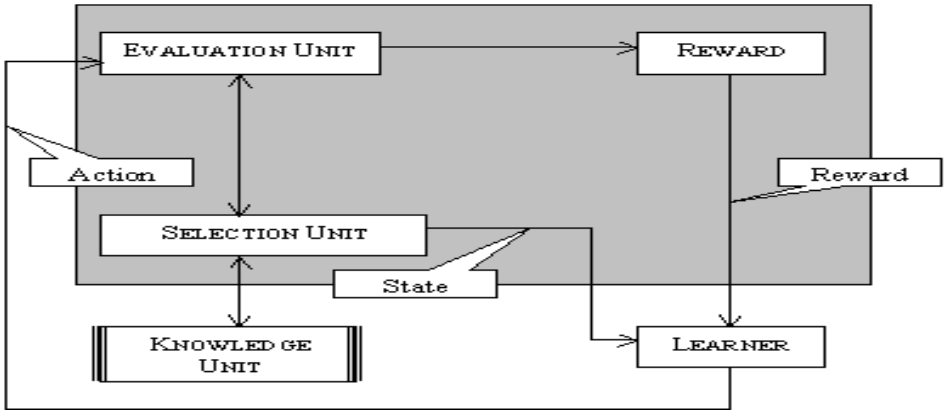

Figure 1: interaction unit components

2.1 Evaluation unit

The evaluation unit is the core of the pedagogical module. The interaction of all units is depicted in Figure 1 and is conceived following the principles of the inferences and the transitions in a training sequence.

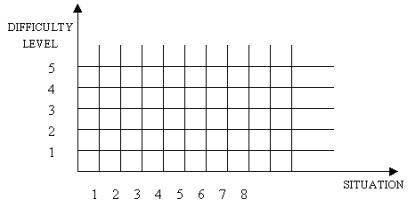

Figure 2: Difficulty and complexity

We considered four principles: success, failure, remediation and high-level principles [BENN2000].

Success principle:

For every situation executed with success, the transition to the following situation does generally from low level toward the right high level, and also horizontally and follow the case of the following situation variable.

Failure principle:

For every situation executed with failure, the transition to the following situation, makes following the cases presented below:

1. One decrements the degree of situation difficulty and one maintain the degree of complexity.
2. If the degree of situation difficulty in progress is inferior to three (3 is the average from difficulty level), and if the situation is executed with success, then we increment the degree of difficulty and we maintain the degree of complexity.

We consider that, all situations having a degree of difficulty lower than the average are conceived in order to be learned, to surmount the difficulties and of reaching the optimal level.

High-level principle:

If the difficulty degree of the current situation reaches the ceiling and if it is executed with success, then the transition to the following situation makes of the following manner: one increases the meter of the measurement situations (complexity degree) and one maintains the maximal rank (partial rank) of the situation difficulties degree.

The remediation principle:

If the difficulties degree in progress situation reaches the lowest level and if the activity of the learner is executed with failure, then the transition to the following situation makes of the following manner: one decreases the meter of the measurement situations (transition to the column precursor) in order to see again the previous situation for to remedied the deficiencies.

2.2 Transition Unit

The transition unit loads on one hand, the selection of the training situations from the database and following the orders of the evaluation unit, on the other, the presentation of situation containing to the user. This unit has two objectives, the one is functional, and concerns search and the recuperation of the information and send it to the presentation function, the other is ergonomic and concerns the choice of the forms associated to the situation in progress and following situation parameters.

2.3 Reward Unit

The object of the reward unit is choosing and sending the adequate feedback to the learner and following his action. The sent message to the learner could be (1) an encouragement following the action executed with success, or (2) some indications in order to complete the instructions to undertake, or (3) a message containing the correct answer. The messages generate since the orders received from the evaluation unit.

3. Tutoring as a Reinforcement Learning Problem

Reinforcement learning is an interesting technique for solving learning problems. It requires a signal of reinforcement that is the only feedback of the environment towards the agent. The agent receives continuously some sensory entrances of the environment called states. It chooses and selects the actions that act on the environment and after every action it receives the environment rewards [SUTT98]. The goal of learning is making the optimal policy that maximizes the rewards and the agent performance.

3.1. Mechanism of training as reinforcement learning

We begin by redefining the mechanism of training like a reinforcement-learning problem. We end with, we will generate a target functions algorithm.

A training sequence is defined like a directed graph as:

- Every situation represents a node (or state) of graph;
- The transition relations between nodes represent the actions and their associate rewards (links).
- Every sequence admits an initial and final situation, respectively the start node (start state) and goal node (goal state).

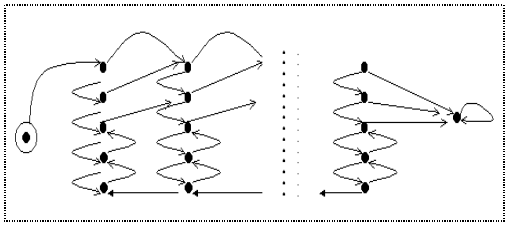

Figure 3: Sequence Graph

3.2. Transition function

One could represent this graph by a matrix line-column, then determine the representation of states, actions, rewards.

States representation:

S is a matrix of 5 lines and N columns representing the situations of a training sequence.

Actions representation:

Normally the actions achieved by the student are:

- Choose an answer (Closed question: MCQ[1], T-F[2], MAQ[3]);
- Type an answer (Open question);
- Unroll a demonstrative situation.

The agent could execute some other actions, like asking for help or giving hints, make some counts using a utilitarian and come back to the situation, etc.

But our interest here is the answer to the following question: Are the actions executed by success or by failure? Knowing that { state + action} gives as a result the following state, then our goal is to determine the outcome of every pair state-action. We can summarize this by giving the actions set: $\{ A^{success}, A^{Failure} \}$

Rewards representation:

Recall that we have five situations s1, s2, s3, s4, s5. Each one of them, is executed with some probability: α, β, λ, μ, ρ.

The reward $R^{a}_{ss'}$ from the current state (s) to next state (s') by executing the action (a) is: $R^{success}$, $R^{failure}$.

3.3. Integration of the success and failure functions

In a concrete situation of learning, the student chooses an answer (closed question) or types an answer (open question) ended by validation. The feedback of the answer validation gives like outcome the veracity of the answer: success or failure. In below, we address the algorithm for these two functions.

What data do we need, if we want to make the connection with a reinforcement-learning problem?

We think that we need four things, a set of states, a set of actions, a reward function and a transition rule or policy. Our interest is to develop the transition rule algorithm, denoted as T.

Then, what is the algorithm of this mapping?

In the first, we signal that the set of actions will be {0, 1} such that the element 0 represent the failure action and the element 1 represent the success action.

```
T(s, i, j, a)                    // 1 ≤ i ≤ 5; 1 ≤ j ≤ N; a ∈{0, 1}.
{
case a = 0                       //failure
{
        load s(i,j) // current state;
        if (i >1)
        then s'← s(i -1,j);
        if (i = 1)
        then
        {
          (1)  load previous situation s(.,j-1);  // Previous column.
          (2)  s'← s(partial  min, j-1); // 1 ≤ partial  min ≤ 3.
          }
    }
case a = 1                       //success
   {
        load s(i,j); // current state;
        if (i >= AVG) //AVG: Average..
        then
        {
        load next situation s(.,j+1); // next column.
        partial  max ← max (s(.,j+1)); // 3 ≤ partial  max ≤ 5.
        if (i < partial  max)
        then s'← s(i+1,j+1)
        else s'← s(partial  max, j+1);
        }
        if (i < AVG)
        then s'← s(i+1, j);
   }
}
```

4. Conclusion

In this paper we concentrated on the modelisation of tutoring systems and the interaction between its components. We demonstrated that the use of modern reinforcement learning techniques can give innovative results for enhancing tutoring systems. We believe that this combination was possible because we could reformulate question like a reinforcement learning problem by determining the whole of the states and their corresponding actions, the rewards and the functions of transitions and the algorithm correspondents. We believe that the multiplication of efforts in concrete and real-life experiments can lead to a frequent use of these reinforcement learning techniques for the design process of intelligent tutoring systems.

References

[ALBE92]: ALBERTINI J. M. , La pédagogie n'est plus ce qu'elle sera. Le Seuil, Presses du CNRS, 1982.

[ANDE97]: ANDERSON J. R., CORBETT A. T. & KOEDINGER K. R.: Intelligent tutoring systems. Handbook of human-computer interaction, Second completely revised Edition, ELSEVIER, 1997.

[BENN2000]: BENNANE A., Processus de conception des tuteurs intelligents et l'apprentissage renforcé ; VUB Bruxelles 1999.

[BRUI97]: BRUILLARD E., Les machines à enseigner. Editions Hermes, Paris, 1997.

[DEP87]: DEPOVER C. , L'ordinateur média d'enseignement, DeBoeck, Bruxelles, 1987.

[DEP99]: DEPOVER C. , GIARDINA M. & MARTON P. , Les environnements d'apprentissage multimédia ; L'Harmattan, Paris, 1999.

[DILL93]: DILLENBOURG P., Evolution épistémologique en EIAO. Ingénierie Educative. Sciences et Techniques Educatives. 1 (1), 39-52, 1993.

[LESC85]: LESCORT A. , Intelligence Artificielle et systèmes experts ; CEDIC NATHAN, Paris, 1985.

[SUTT98]: SUTTON, R. & BARTO A. G.; Reinforcement Learning, A Introduction; A Bradford Book, 1998.

[1] Multiple Choices Question.

[2] True – False.

[3] Multiple Answers Question.