Autonomous Virtual Agents Learning a Cognitive Model and Evolving

Toni Conde and Daniel Thalmann

Ecole Polytechnique Fédérale de Lausanne (EPFL), Virtual Reality Lab,
CH-1015 Lausanne, Switzerland
{Toni.Conde, Daniel.Thalmann}@epfl.ch
http://vrlab.epfl.ch

Abstract. In this paper, we propose a new integration approach to simulate an Autonomous Virtual Agent's cognitive learning of a task for interactive Virtual Environment applications. Our research focuses on the behavioural animation of virtual humans capable of acting independently. Our contribution is important because we present a solution for fast learning with evolution. We propose the concept of a Learning Unit Architecture that functions as a control unit of the Autonomous Virtual Agent's brain. Although our technique has proved to be effective in our case study, there is no guarantee that it will work for every imaginable Autonomous Virtual Agent and Virtual Environment. The results are illustrated in a domain that requires effective coordination of behaviours, such as driving a car inside a virtual city.

1 Introduction

The production of believable Autonomous Virtual Agents (AVAs) that are outfitted with learning abilities in a Virtual Environment (VE) is very helpful in many areas. In computer games, the use of AVAs capable of learning a specific task and evolving their skills for that task can greatly improve both the enjoyment and the strategy of the game-play.

An AVA driving a car inside a virtual city is an example of this feature. By adjusting its internal "memory" to match the level of difficulty, the AVA is able to accomplish the task. This process of problem solving can be referred to as task learning. In real life, human learning involves many complex cognitive processes. Realistically, the simulation of AVAs exhibiting behaviours that reflect those of humans demands efficient simulation algorithms. This is especially true for the interactive systems such as computer games.

A number of challenges are raised in developing a system incorporating learning AVAs. From a behavioural animation point of view, there are several areas to consider, such as:

1. The design of a learning control structure,
2. The internal storage of the learning information and
3. The efficient evaluation and calculation of feedbacks and reactions from the environment.

T. Panayiotopoulos et al. (Eds.): IVA 2005, LNCS 3661, pp. 88 – 98, 2005.

Learning involves adaptation and evolution in which modifications made by internal subunits of the adaptive system, like the human brain, mirror external environmental changes. Up to a certain degree of complexity, many Artificial Intelligence (AI) models are able to simulate human learning behaviour [1].

The simulation of human behaviour is achieved through the use of a complex "cognitive map" and the application of a hierarchy of behavioural strategies [2]. The overall cognitive mapping process involves acquisition, coding, storage, recall and decoding [3] of the environmental information. In fact, an individual "cognitive map" will often contain numerous inaccuracies or distortions [4]. Many of these are due to the fact that humans predominantly use a visual perception system and they are unable to process everything they see because of the vast amount of incoming information [5]. Other errors result from the way the information is processed and stored within the "cognitive map" structure itself. Therefore, to simulate human-like behaviour more closely, we separate the AVA from its environment and provide it with perception and effector systems only.

We have developed [6] new methodologies to map all the information coming from the VE and from the virtual sensors of vision, audition and touch in the form of a "cognitive map". They enable the partial re-mapping of the cognitive and semantic information at a behavioural level. For example, when spatial attention is primed with tactile stimulation, the location of the attention spotlight is only partially re-mapped in visual coordinates. With the aid of this framework, we can prepare multi-sensory information for *cognitive learning*.

Unlike mechanical memory that can permanently store information, human memory is imperfect and information can be forgotten. Humans and animals selectively process only the information that is important to them whilst actively searching for new information. Similarly, we can have two types of learning in an intelligent system:

1. *Active learning* where the system selects filters and searches for relevant information.
2. *Passive learning* where the system accepts all incoming data.

In this paper we are presenting research work in the domain of behavioural animation using a high learning approach combined with an active learning approach. This is accomplished through the use of a *cognitive model* defining how the AVA should react to stimuli from its environment. In summary, this paper presents a novel approach that allows an AVA to learn a "cognitive model" by itself.

Document Organisation: Section 2 – State of the Art; Section 3 – Methodology; Section 4 - Realisation and Integration; Section 5 – Experimental and Results; Section 6 – Discussion and Improvement Proposals.

2 State of the Art

A great deal of research has been performed on the control of animated autonomous characters [7-10]. These techniques have produced impressive results, but are limited in two aspects. Firstly, they have no learning ability and are thus limited to explicit pre-specified behaviours. Secondly, they only perform behavioural, not cognitive,

control (where *behavioural* means reactive decision making and *cognitive* means reasoning and planning to accomplish long-term tasks).

On-line behavioural learning has begun to be explored in computer graphics [11] and [12]. A notable example is [13], where a virtual dog can be interactively taught by the user to exhibit a desired behaviour. This technique is based on *reinforcement learning* and has been shown to work well in [14]. However, it has no support in long-term reasoning to accomplish complex tasks. Also, since these learning techniques are all designed to be used on-line, they are, for the sake of interactive speed, limited in terms of how much can be learned.

3 Methodology

In this section we introduce AVA learning in which an AVA automatically learns an unknown *cognitive model*. We have developed a novel technique to achieve AVA learning using a tree search with a *k-nearest neighbours* (k-NN) method.

3.1 Human Adaptability to Learn

In order to simulate the AVA's learning behaviour, a learning model has to be adapted. Learning by experience is one of the most well known principles of human task learning behaviour [15]. Indeed, most of the time we learn by direct experience in performing a task. Learning is an intricate process which involves many aspects of cognitive activities including knowledge acquisition, observing and thinking.

Since each person has his/her own motivations and method of learning, the learning process is affected by the learning pattern. To perform a specific task, a person's

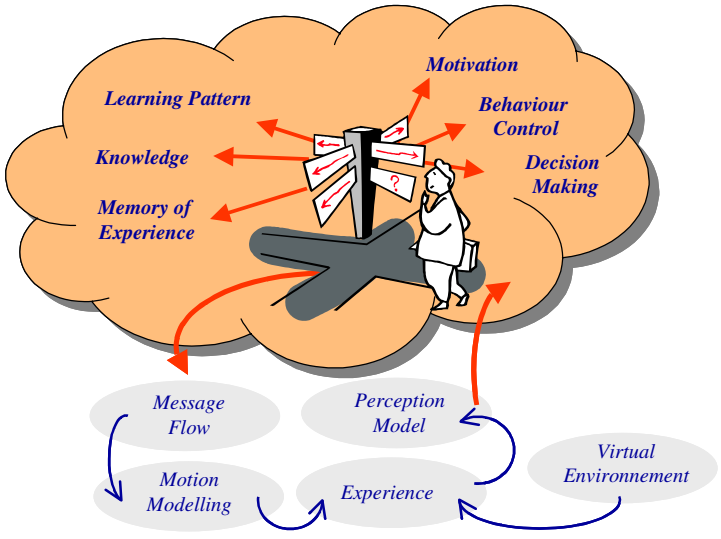

Fig. 1. General behavioural Simulation Model

skills and abilities for the task can be developed during practice [16]. This concept is summarised in Fig. 1 and is used to make up the basis of our learning *cognitive model* for the AVA simulation. Fig. 1 shows the key elements of the human learning process such as: background knowledge of a specific task, motivations to accomplish the task, memory of the past experiences, individual learning pattern and finally trial and error. The learning process is a process of adaptation, evolution and decision making as a whole. Another key issue of learning is the environmental feedback.

3.2 AVA Learning a Cognitive Model and Control Structure

For any given AVA and VE the state space must be continuous. This is because in a stimulating environment where an agent and a human are competing or cooperating intimately, a small difference in state can lead to a large difference in behaviour. A continuous state space can also help achieve a realistic VE. For example, in our car driving case study, a discrete state space would be very unnatural for a car driving simulator. Therefore, our technique uses a continuous internal representation of states and actions.

Most machine-learning algorithms make general and weak assumptions about the nature of the training data. As a result, they typically require large amounts of data to learn accurate classifiers. Normally, the performance improves as the algorithm exploits more information. It generally performs better at recognition than at generalization. This problem can be solved by taking advantage of prior knowledge to eliminate the inconsistent classifiers. Hence, the resulting learning algorithms may be able to learn from very few training examples. To recognise a point, the *k-NN* method implicitly makes a comparative estimate of all the densities of class probabilities appearing in its vicinity and chooses the most probable. In fact, it approximates the Bayesian decision. Finally, a vector of quantification is introduced. The technique consists of replacing a completed combination of points by a limited number of prototypes representative of the training set.

However, there is a risk involved in incorporating prior knowledge, since this can add a bias to the learning process. If the knowledge is incorrect, it will then eliminate all the accurate classifiers. As a result, learning algorithms tend to perform fairly well on small training sets but, as the amount of data increases, as in our driving context, their performance suffers because they under fit the data.

In most real-world problems all of these approaches are limited by the very large space of the possible states. These algorithms typically require time that is scaled in terms of the cube of the number of states. Hence, [17] and other researchers have focused on methods to construct computationally manageable approximations of the policy, the value function and the model.

The *k-NN* algorithm was chosen as it provides a local approximation of the target function and can be used automatically without the designer selecting the inputs. It is guaranteed to learn the target function based on the quality of the examples provided and to memorize the decisions made by planning through a *cognitive model*. The decision-making of a *cognitive model* is a very important piece of information. The mapping is likely to be smoother if the information is presented as a separate input to the *k-NN* algorithm.

Generally with the *k-NN* approach, decreasing the number of points reduces the search space and the storage problem. This also leads to a diminution of the computation time. We used a pre-computed phase before the search phase to reorganize the learning space.

The *k-NN* method does not require the separation of the various classes of learning. Instead, we selected a sub-domain of learning points. However, the method necessitates the explicit storage of many examples of the target function. It can also automatically discover the inputs necessary to approximate the target function like in our car driving *cognitive model*. The choice of the *k-NN* metric influences the rate of error and rejection.

Our technique is quite scalable since, if a global approximation is needed, the *cognitive model* can be approximated by several separate machine learners: *k-NN, DSM (Decision Surface Mapping)*, *LVQ (Learning Vector Quantization)* and *SVM (Support Vector Machine)*. Each of them learns a distinct subset of the state to action mapping (see Fig. 2). Decision-making in different regions of the state space may rely on different state information and therefore these machine learners can use different state formulations to reduce the dimensionality.

Our new approach uses a methodology adapted from the data mining domain [18] which computes a locally flexible metric by means of *SVM*. The maximum margin boundary is used to determine the most discriminated direction over the query's neighbourhood. Such direction provides a local weighting scheme for the input features.

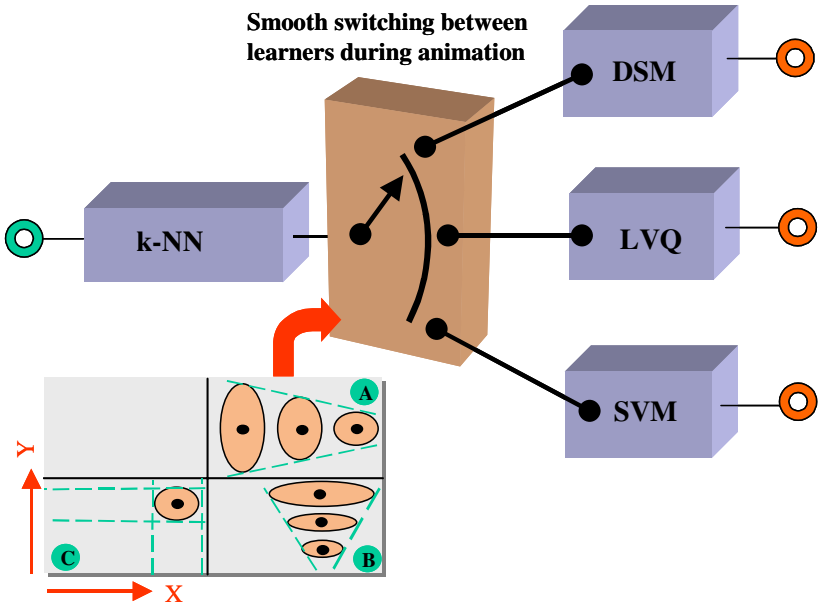

Fig. 2. Cognitive Model with smooth blending. For query A, dimension X is more relevant because a slight move along axis X may change the class label, while for query B, dimension Y is more relevant. For query C, however, both dimensions are equally relevant.

To allow for smooth switching between learners during animation, the actions recommended by each one can be blended for a period of time (see Fig. 2). Traditionally, *cognitive models* are very slow to execute. Performing our smooth blending technique accelerates this cognitive learning process.

3.2 Evolving Process

Each individual's learning pattern and knowledge about the task are represented by predefined motion patterns that may be motion capture data.

Taking the example of high jump, an AVA is assumed to have previously acquired the knowledge of how to jump by making full body movement. However, the AVA has to improve its performance in order to achieve a target. A high jump athlete may have to make several attempts before he/she can jump over a horizontal bar. Similarly, simulating this kind of task requires an evolution model that approximates the evolving learning process during which the ability of the virtual athlete evolves as it improves.

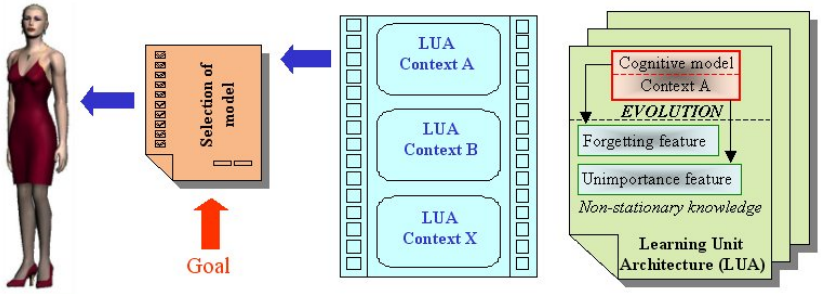

Fig. 3. AVA High Learning (AVAhighLEARN) with our Learning Unit Architecture (LUA)

Our proposal of an approach for the evolving process involves "behaviour capture" and a *Learning Unit Architecture* (*LUA*). Supplying a different cognitive model for each context is a simple method of learning context-sensitive policies. These policies are then placed in the AVA's brain and the selection of the suitable *k-NN* to use is determined by the AVA's current internal state (see Fig. 3).

For the evolving process, we introduce features such as forgetting and unimportance. If a state to action case was recorded long ago and/or is very similar to a new one being added, it is likely to be removed. Thus the AVA has the ability to "forget", which is very important in learning something as dynamic as a human behaviour.

4 Realisation and Integration

The realisation and integration of our AVA High Learning (*AVAhighLEARN*) methodology which combines different machine-learning techniques with several novel improvements could be more useful to the computer graphics community than techniques based purely on machine-learning approaches (see Fig. 4).

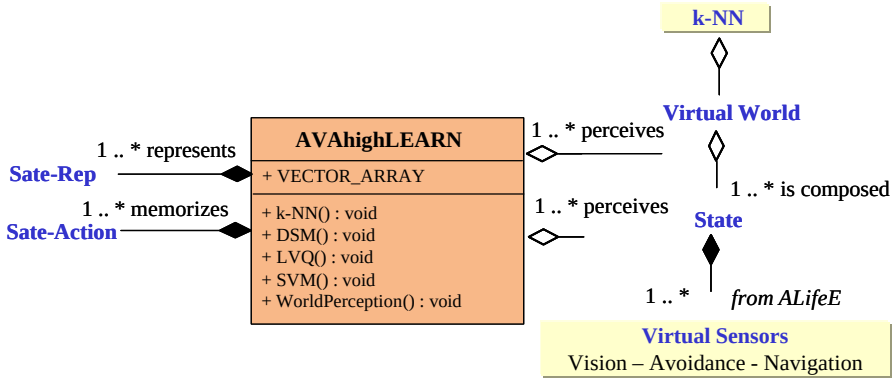

Fig. 4. Comprehensive UML design of *AVAhighLEARN* including virtual sensors from our *AlifeE framework*

An AVA is fitted with sensors to inform it of the state of its external and internal VE. An AVA also possesses effectors to exert an influence on the VE and a control architecture to coordinate its perceptions and actions. The AVA's behaviour is adaptive as long as the control architecture allows it to maintain its variables in their viability zone. All of these characteristics are integrated in our *ALifeE framework* (see Fig. 4) developed for our research. It is based on an original approach inspired by neuroscience and equips an AVA with the main virtual sensors in the form of a small nervous system [6]. The acquisition steps of signals, filtering, selection and simplification intervening before proprioception, active and predictive perception are integrated into virtual sensors and a virtual environment.

5 Experimental and Results

With our approach it is not necessary to program an explicit *cognitive model*. Studying how a task is accomplished is usually necessary before an explicit AI model can be programmed. Thus, in this experiment, our technique for AVA learning relieved us of this burden and therefore reduced the animation workload.

We implemented our *cognitive model* learning approach to the driving simulation of a car inside a virtual city (see Fig. 5a and b). The AVA is a pilot driving a car inside the virtual city. The pilot and his/her co-pilot, have dual control over the acceleration and the wheel of the car (see Fig. 6). The controls are real-value (e.g. the action space is continuous) and the car can move to any location or take any orientation. The continuous action is then quantified to achieve real-time performance. Consequently, the possible actions of the pilot and the virtual instructor become limited.

The experiment is performed with an approximate cognitive model with the *ALifeE framework* [6] but with pseudo-perception features. The characteristics of pseudo-perception are used to compare the performances obtained with case study including our *ALifeE framework*. Indeed, in most of the AVA's simulation environments, sensorial modalities and perception are not integrated in a way faithful to reality. In this experiment visual pseudo-perception is provided by the AVA pilot's field of view

which is in the shape of a circular zone (see Fig. 7a and b). Dynamic (e.g. cars) and static (e.g. road signals, traffic lights) objects are represented by rectangular graphic symbols. To test the recognition of the road signals and traffic lights, we integrated this visual pseudo-perception method so that it could determine which object is the closest to a given ray "r" of the circular zone (see Fig. 7b).

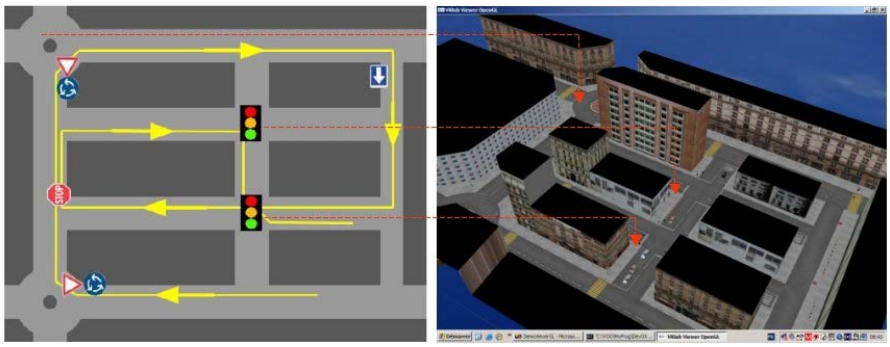

Fig. 5a and b. Car driving simulation inside a virtual city. Semantic information such as road signals and traffic lights are included.

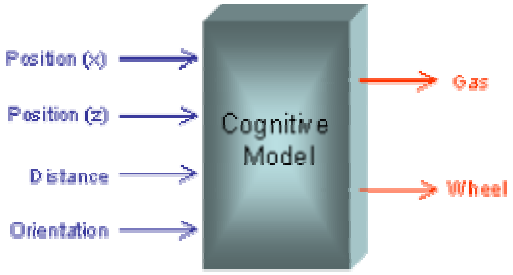

Fig. 6. Explicit cognitive model with inputs and outputs

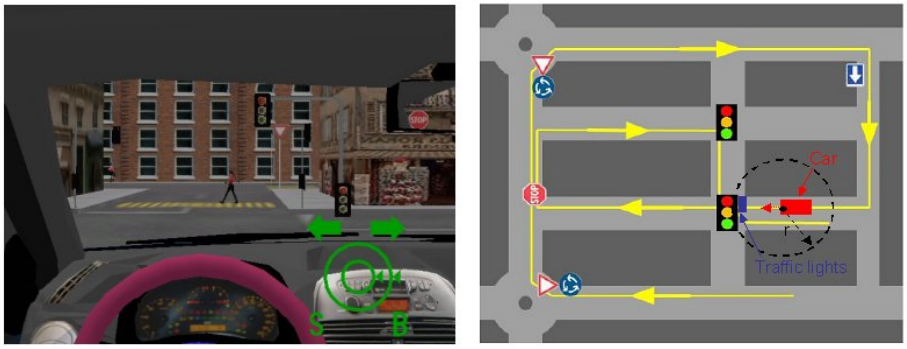

Fig. 7a and b. An AVA learning to drive a car inside a virtual city with visual pseudo-perception. The car "sees" the traffic lights inside a circular zone.

The k-NN algorithm was trained to approximate a single policy. It is only useful for one cognitive model and one goal at a time (see Fig. 3). Fig. 8a and b show the training of the approximate cognitive model for driving and the path of a car inside the virtual city, respectively. Subsequently, the information is used to simulate the behaviour of the AVA pilot.

There can be more than one model for any given goal so that greater variety and/or robustness can be achieved. It is also possible to use the k-NN algorithm with different explicit cognitive models of the same AVA's "brain" (see Fig. 3).

In this experiment we improved the planning of our cognitive model taking advantage of the pseudo-perception features.

We tested our methodology with a *LUA* concept, mainly to encourage evolution of push the behaviours of the pedestrians to evolve and to verify the car pilot's ability to "forget", which is essential in learning dynamic human behaviours.

The final result was good driving behaviour, since the pilot could plan far enough ahead to adequately manoeuvre the car inside the virtual city. We achieved our best results by performing low-level learning method for 40 iterations (see Table 10).

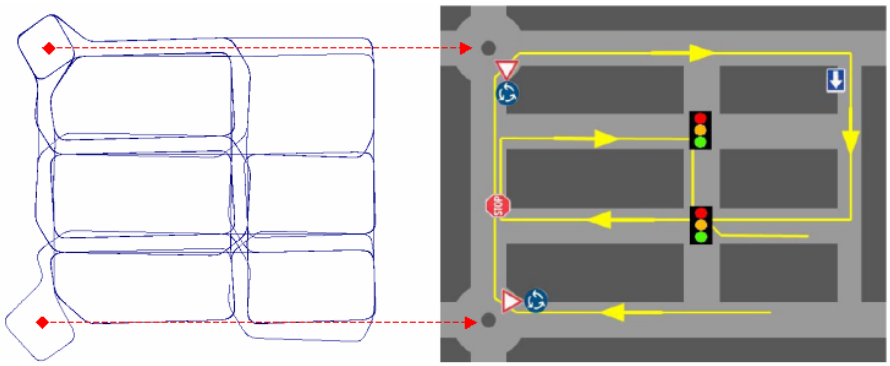

Fig. 8a and b. Snapshot of the car path with approximate cognitive model

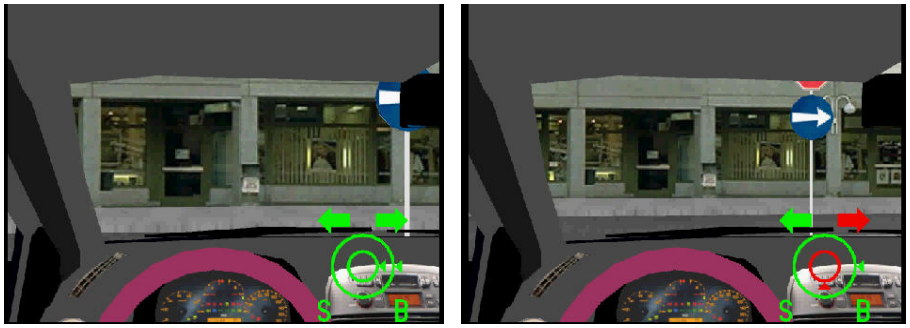

Fig. 9a and b. A pilot wishes turns left and the co-pilot's indicators inform him to turn right based on his learning knowledge – road signals. The panel informs the driver that he/she must turn right (Arrow in red at bottom right corner of fig. 9b). The co-pilot, steering wheel is indicated by a red circle.

Table 10. Results of our cognitive model with AVAhighLEARN method, using Learning Unit Architecture (LUA). All animations were rendered in real time using OpenGL on a 3.0 GHz PC with an nVIDIA GeForce FX Go5350 video card.

	k-NN	**DSM**	**LVQ**	**SVM**
Execution time	15 µs	9 µs	8 µs	6 µs
Storage	1.4 MB	1.4MB	1.2 MB	24 KB

6 Discussion and Improvement Proposals

In this paper we presented a novel approach to simulate an AVA's task learning behaviour for interactive VE applications. Our contribution is to propose the concept of a *Learning Unit Architecture* (LUA) that works as a control unit of the AVA's brain. The *LUA* model is based on a human learning model. It is not a true simulation of the real human brain's learning activities, but rather a simulation system that models its numerous aspects. This *LUA* can also be extended to represent different types of learning behaviours.

Through this general and reusable technique, an AVA automatically learns to mimic the intelligent decision making process of a human. This is carried out by a human animator who has interactive control over the actions and decisions of the AVA. The designer constructs the *cognitive model* in an intuitive manner thus making this process simpler and quicker.

Future work should continue to improve the current simulation system in order to simulate more complex human learning behaviours. The challenges that need to be addressed concern the efficiency, the realism and the control of the simulation.

Through this *AVAhighLEARN* method, an AVA can independently and automatically learn a *cognitive model*. For the animator, this alleviates the workload of designing an explicit model. It also permits the creation of tasks for which it would be difficult, or virtually impossible, to develop an explicit model.

However, there are some weaknesses in our approach. For instance, when performing on-line AVA learning, it can be hard to design the expected behaviour of the *cognitive model* with exactitude.

Simulating automatically learning behaviours is a not an easy and appealing task. Our approach could take interactive computer graphics to a completely new level, especially in the entertainment market. It would also be very useful if an animator could interactively train an AVA for *cognitive learning*.

The approach presented here is part of a more complex model that is the object of our research. The goal is to realize a Virtual Life environment for an AVA including different interfaces and sensorial modalities coupled with different evolving learning methodologies.

Acknowledgments. This research has been partially funded by the Swiss National Science Foundation.

References

1. Flake, G. W.: *The Computational Beauty of Nature*. The MIT Press, 2000.
2. Larkin, P.: Achieving human style navigation for synthetic characters. A survey. In *Proceedings of Neural Networks and Computational Intelligence*, 2003.
3. Downs, R., Stea, D.: Cognitive maps and spatial behavior. *Image and Environment*. R. Downs and D. Stea, Eds. Chicago: Adline Publishing, 8-26, 1973.
4. Griffin, D.: Topographical Orientation. *Image and Environment*. R. Downs and D. Stea, Eds. Chicago: Adline Publishing, 296-299, 1973.
5. Kamwisher, N., Downing P.: Separating the wheat from the chaff. *Science*, vol. 282, 57-58, 1998.
6. Conde, T., Thalmann, D.: An Artificial Life Environment for Autonomous Virtual Agents with multi-sensorial and multi-perceptive features. *Computer Animation and Virtual Worlds*, 15(3-4), 311-318, John Wiley, 2004.
7. Reynolds, C.: Flocks, herds, and schools: A distributed behavioural model. In *Proceedings of ACM SIGGRAPH*, 25-34, 1987.
8. Tu, X., Terzopoulos, D.: Artificial fishes: Physics, locomotion, perception, behaviour. In *Proceedings of ACM SIGGRAPH*, 43-50, 1994.
9. Blumberg, B., Galyean, T.: Multi-level direction of autonomous creatures for real-time virtual environments. In *Proceedings of ACM SIGGRAPH*, 47-54, 1996.
10. Perlin, K., Golberg, A.: A improv: a system for scripting interactive actors in virtual worlds. In *Proceedings of ACM SIGGRAPH*, 205-216, 1996.
11. Burke, R., Isla, D., Downie, M., Ivanov, Y., Blumberg, B.: Creature smarts: The art and architecture of a virtual brain. In *Proceedings of the Computer Game Developers Conference*, 2001.
12. Tomlinson, B., Blumberg, B.: Alphawolf: Social learning, emotion and development in autonomous virtual agents. In *Proceedings of First GSFC/JPL Workshop on Radical Agent Concepts*, 2002.
13. Blumberg, B., Downie, M., Ivanov, Y., Berlin, M., Johnson, M., Tomlinson, B.: Integrated learning for interactive synthetic characters. In *Proceedings of ACM* SIGGRAPH, 417-426, 2002.
14. Conde, T., Tambellini, W., Thalmann, D.: Behavioral Animation of Autonomous Virtual Agents helped by Reinforcement Learning. In *Lecture Notes in Computer Science*, vol. 272, Springer-Verlag: Berlin, 175-180, 2003.
15. Jordan, M.I., Rumelhart, D.E.: Supervised Learning with a distal Teacher, In *Cognitive Science*, 16, 307-354, 1992.
16. Bransford J.D., Brown A.L.: Cocking R.R. Brain, Mind, Experience, and School, National Academy Press, Washington, 1999.
17. Mitchell, T.: *Machine Learning,* McGraw Hill, 1997.
18. Domeniconi, C., Gunopulos, D.: Adaptive Nearest Neighbor Classification using SupportVector Machines. In *Advances in Neural Information Processing Systems 14*, MIT Press, 223-229, 2001.