SIMULATION AND VISUALIZATION OF ANTICIPATORY ALIFE AGENTS

Karel Kohout, Pavel Nahodil

Department of of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Prague, CZECH REPUBLIC

E-mail: {kohoutk,nahodil}@fel.cvut.cz

Abstract: This paper presents our approach in design, simulation and visualization in the field of anticipation and ALife agents. Anticipation occurs in all spheres of life. Nature evolves in a continuous anticipatory fashion targeted at survival. A conscious reaction takes too long to process. Motivation mechanisms in learning, the arts, and all types of research are dominated by the principle that an expected future state controls present action. Under anticipation in ALife domain we understand more that prediction or estimation of the future states. Anticipation takes the information from prediction or estimation as an input in order to perform reasoning, learning and planning. Several anticipatory architectures have been proposed, implemented and tested in the designed simulation environment. A part of visualization is targeted on simulation world itself and the agents that are living in it. Another part is concerned about analysis of parameters and properties of simulation in time.

Keywords: ALife, Simulated environment, Anticipation, Behaviour, Animate, Agent

1. INTRODUCTION

Our research in ALife agent domain began with hybrid agent architectures and evolved lately to anticipation architectures. Hybrid architectures are a trade off between fast but hardly changeable and learning incapable reactive architecture (bottom-up) and the slow but adaptive and planning capable deliberative architecture (top-down). Hybrid deliberative-behavioural architectures have strong roots in biology. Research problem is to find the boundary between both approaches. I will not go into detail about these architectures because they are well described and out of scope of this paper, more about them can be found in (Kadleček, 2001). From the working and designed hybrid architecture we moved towards anticipation. We integrated the anticipatory block into the hybrid architecture. We aim to improve agent's behaviour not only in terms of survival. Nowadays the mere survival is not enough. We want our agent to complete various tasks, to have its own will and own feelings. The description how we are achieving these goals will be described further in this paper.

2. ANTICIPATION

Before introduction of designed agent architectures and also simulation environment we will introduce the term anticipation. We will also determine what we understand under anticipation in terms of ALife. Basic definition of anticipatory systems was published in 1985 by biocyberneticist R. Rosen in his book Anticipatory systems (Rosen, 1985). He defined an anticipatory system as follows: "A system containing a predictive model of itself and/or its environment, which allows it to change state at an instant in accord with the model's predictions pertaining to a latter instant". On the turn of century this definition was revised by D. Dubois (Dubois, 2003). His research showed, that we can observe anticipation even in the systems where the creation of the model is not possible in principle (such as galaxies, electromagnetic systems etc.). Therefore he defined two categories for anticipation strong and weak. Strong anticipation systems are those, which crates the model of the object they are interacting with. Weak anticipation systems do not creates the model, it is a part of their structure. In artificial intelligence, anticipation is the concept of an agent making decisions based on predictions, expectations, or beliefs about the future states. It is widely recognized that anticipation is a vital component of complex natural cognitive systems. The opposite for anticipation behaviour is reactive behaviour. Anticipation seems to be suitable for key role in design and realization of anticipatory behaviour. Our understanding of anticipation is very closely connected to behaviour. The mere estimation or prediction of the future is not enough. We need to take this information and integrate it into the decision making process in order to improve it. This is our goal which is currently under an intensive research.

2. SIMULATION ENVIRONMENT

For the various simulations we use to prove correctness of designed architectures we designed our own simulation environment. Its design started two years ago and has been continuously worked on recently. Several agent architectures have been tested in this environment. We named this environment World of Artificial Life (WAL). It defines, what objects can agent meet, what changes are possible, provides the agent with information about the virtual world and it defines the body of an agent. WAL architecture was mainly focused on environment. It used the agent architecture from previous research, mainly (Kadleček, 2001). We modified this agent architecture to comply with the environment. We intended to use it then to compare behaviour with other architectures. The proposed abstract architecture consists of several parts. Each one of them is focused to one specific area. It recommends the selected solution in order to maintain interconnectivity and modularity. Application engine is main program part. It synchronizes time steps for whole application, contains interfaces to all other parts (modules) of application and also contains and maintains all part of simulation like agents and environmental layers (see below). Interface between engine and its program surrounding is data representation. Well-defined standard of XML language is supported, but has proved as highly inefficient to transfer along network to other modules of distributed application. For this purpose binary data representation was designed. This binary representation is conversion of XML tree structure to byte effective format. Environmental layers are the main components of virtual world of agents. They enable to dissemble complex world into simpler part which are easy to implement. They can overlap each other and together they will create more complex world. The layer is a logically detachable part of world which is capable to act individually and which creates the virtual world along with other layers. All described above is just an algorithm with no human interface. Visualization of the designed world can be both attractive and useful tool. For this purpose an external visualization module or internal (default) can be used. The internal visualization is mainly meant to debug and observe simulation by creator. The external module can be exact opposite. It I supposed to be used to present this simulation to the wider audience than science community. The on-line or off-line tools for analysis of the parameters flow in time are also supported. The proposed environment is compatible with the 3D visualization tool called VAT (Řehoř, 2003). It can be used to observe any of agent's parameters at any time of simulation. Running simulation can also be stopped at any moment and even traced back to certain point of the history. It can also be used to run again the simulation to observe if any change of behaviour will occur while starting from exactly same situation. Also change of simulation parameters should be available while simulation is running. Agent is any object in simulation either virtually alive (creature, predator) or virtually non-living (trees, food, water, rocks). Sensors and effectors of agent are his interfaces with virtual world therefore they are part of the environment and its layers. Besides that agent mind and control are not part of environment. They are separated and can even run remotely. Here described architecture was implemented and currently is being used in few diploma theses for testing agent control mechanisms. The effort of creators of this application is to present it as standard for Artificial Life domain specific simulations. The work on this subject follows up on the former research of a MRG group on FEE, CTU. It concludes the design of new environment for Artificial Life domain simulation. They are used in both the visualization of an environment in which agents live and the visualization of agent parameters over time. This serves for better analysis and better behavioural pattern recognition. While designing this architecture special care was applied to modularity of the whole application.

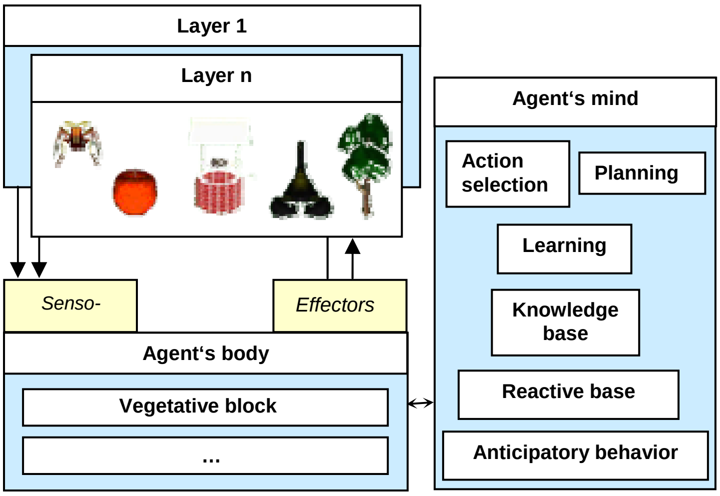

Fig. 1. Block scheme of WAL abstract architecture

3. ANTICIPATORY ARCHITECTURES

The above described architecture gave us a solid ground for further research of behaviour. Further works started with anticipation research. It is an interesting yet challenging topic. There are several successes on the field of anticipatory behaviour. We would like to introduce two already designed and implemented anticipatory architectures with different approaches described and implemented.

3.1 Architecture Lemming

First architecture was named Lemming (Foltýn, 2005). This architecture fully leveraged from the WAL environment. It uses the algorithms known from Artificial Intelligence for agent's learning and offers an alternative to genetic programming and the artificial neural networks commonly used in ALife domain. This model ensures agent's survival in unknown dynamically changing environment. Learning abilities of architecture Lemming were proven in experiments in which agents acted. Agent learned which objects are necessary for his survival. This architecture introduced Classification Information Block (CIB) for information storage about agents' usabilities. Usability is an attribute of agent saying what it can do with an object and it is derived in trial-error iteration. The structure used for CIB is decision trees implemented by Top-Down Induction of Decision Trees (TDIDT) algorithm. To find the most significant attribute and recursively continue until there are no attributes or training examples left measuring the amount of information by entropy is used. Generalization using LGG (least general generalization) algorithm was incorporated into architecture Lemming and used for agents. During the process of reasoning, each agent/object in the surrounding environment is assigned some attraction based on agent's knowledge, his current needs and long-term goals. An attraction is a basis for agent's movement using concept of the virtual potential field generation known from mobile robotic. Several experiments were presented. Lemming's long-term ability to survive in given environment proves justification for chosen and implemented behaviour primitives and proper tuning of used settings. In Predator-Prey Simulation, an application of architecture Lemming on classical problem from ethology and biology is used. The simulation also presents simple application of the knowledge transfer from parents to descendants. Setting experiment constants leaves space for user's interaction with the simulator.

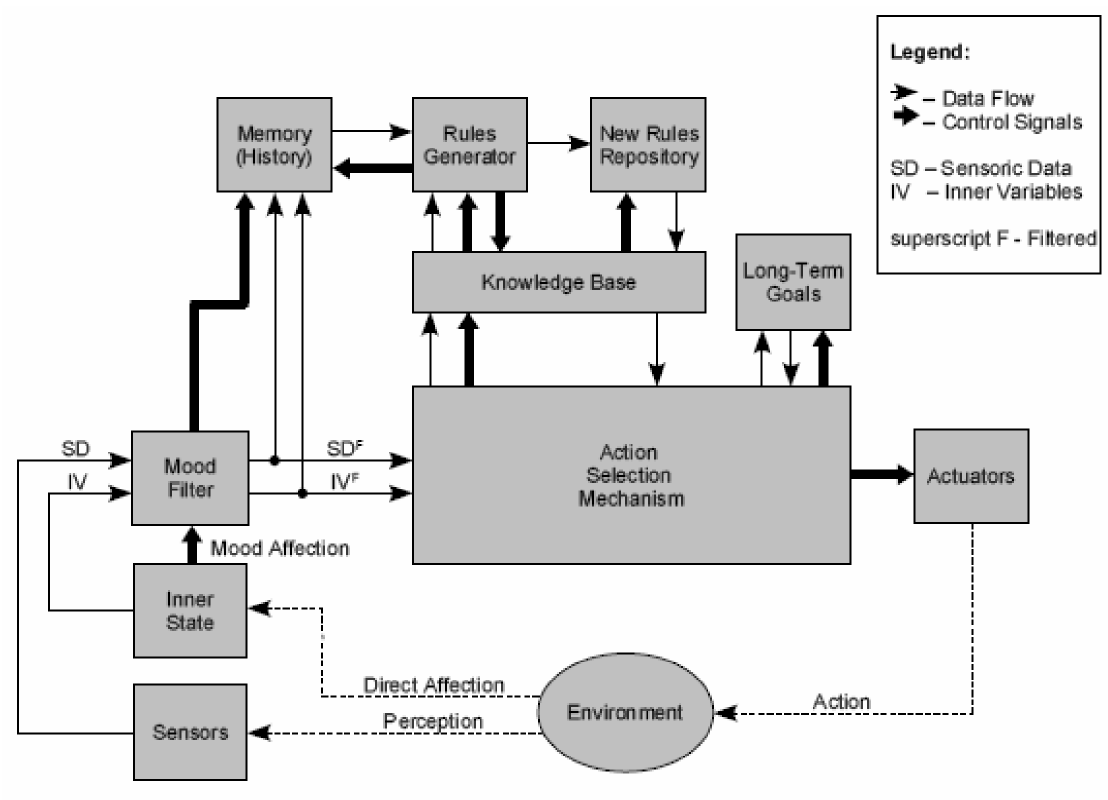

Fig. 2. Block scheme of Lemming agent architecture

3.2 Anticipatory Learning Classifier System

The other architecture designed focuses on learning problem in artificial life domain. The Anticipatory Learning Classifier System (ACS) was defined and proven as promising approach. It combines reinforcement learning, evolutionary computing and other heuristics to produce adaptive learning system (Mach, 2005). Learning and adaptation can be in ACS divided into several topics. They are knowledge structure, decision making, new rule generation, generalization and learning from environment with delayed reward. Created rules are in the form of “IF condition THEN do action AND EXPECT next state” format. Each rule in ACS has in addition two values, the reward for performing the action and the quality of prediction (how precise the estimation of the next state is). ACS models the environment as hidden Markov model. The aim is to discover the states of the model which are initially unknown.

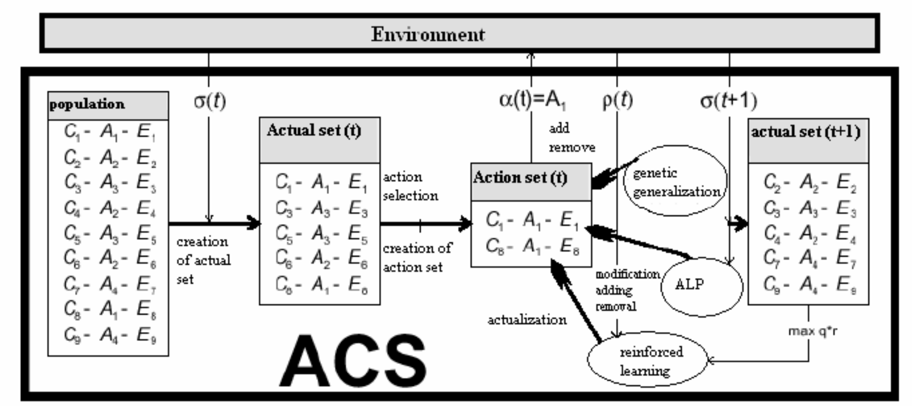

Fig. 3. Block scheme of ACS agent architecture

4. SIMULATIONS

Several simulation scenarios was tested to prove capabilities of each designed architecture. All of mentioned agent architectures are capable or running in WAL. Even the works created before WAL were modified to be able to run in this environment. Architecture Lemming was designed in WAL directly. ACS was primarily implemented in Matlab but the interface to communicate with WAL environment was proposed. The graphic capabilities of the designed architectures are not sufficient in case of displaying more parameters. In WAL simulations we used the graphical tool named VAT for detailed analysis. This tool is helpful not only in the simulation evaluation phase but also in the phase of debugging the architecture. It can be used to detect errors in the implementation. I will demonstrate this on use case. The simulation scenario for a single agent was intended to move an object between two places. The agent had enough food and water to satisfy its needs. Fig. 4 shows the visualized data from this simulation. This picture shows the paralel coordinates mesh. Each variable has its own axis and these are layed in paralel and the actual values are connected to form a curve. The third dimension is time. Even a very first look at the 3D mesh could advise that there is obviously something wrong with the simulation. Almost all of parameters are zero (the mesh is flat). The agent is not hungry or thirsty. It is neither tired nor sleepy. But we implemented and designed all these. The reason for this could be a data export failure, a mistake in implementation of the inner agent vegetative block or bad initial configuration of an agent. Because we run the simulation previously and exported the data and the vegetative block was working properly, there is no problem with implementation itself. Brief check of the configuration showed that there was enormously high value set to the time function of the chemicals. This cause their very little change even in a long time sumulation.

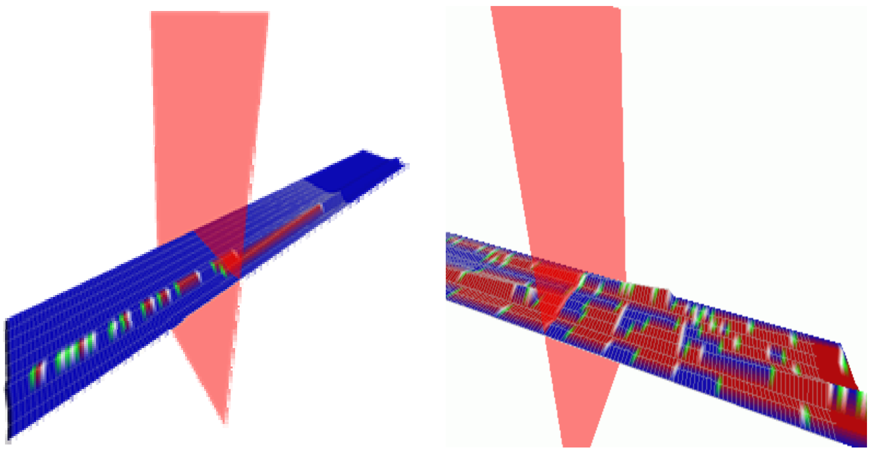

Fig. 4. Use case – visual tool usage for debugging

Another experiment named Rescue Mission was performed on architecture Lemming. This experiment shows learning abilities of architecture Lemming and applied generalization. The experiment had two parts. Firstly the agent has to discover which tool belongs to which create and then successfully use it to open it. In the scenarion there were four different tools each for opening a different create. The agent is motivated by a reward locked in the last create. All experience is gained by trial-error learning plus there is a generalization used for proper tool usage discovery. After each successful utilization of a tool, knowledge in General Rules Block is updated. Two successful examples of opening different kinds of crate are enough for lgg algorithm to create general rule connecting crate's property, used tool and the final state of the crate. This knowledge is then used for setting new long term goals to find proper tools which agent doesn't posses and which will be needed for perceived crates opening. We have to emphasize that Classification Information Block doesn't contain any information about tools for which agent deduced he will need (using general rule) and these tools would be treated as collectible objects.

5. CONCLUSION

In this article we showed the recent advances in the field of artificial life specifically focused on the anticipatory behavior in our reserach group. Anticipation as we see it is not plain prediction or estimation of the future. This can be done using several statistical approaches. Anticipation in ALife means more then just prediction. It is for us utilizing the obtained or derived information about the future for the higher cognitive processes such as planning and decision making. It has been shown that anticipation cannot be the only control mechanism. The reaction base, and other features, ensuring learning, and evolution of an agent are necessary as well. The information about future can be taken in account while decision about next action is made. Anticipation is still the subject of intensive research.

REFERENCES

Rosen, R. Anticipatory Systems - Philosophical, Mathematical and Methodological Foundations, Pergamon Press, 1985, ISBN 0-08-031158-X.

Dubois, D. M. Mathematical Foundation of Discrete and Functional Systems with Strong and Weak Anticipation. In Anticipatory Behavior in Adaptive Learning Systems. Lecture Notes in Computer Science 2684. Heidelberg: Springer, 2003.

Kadleček, D., Nahodil, P. New Hybrid Architecture in Artificial Life Simulation. In Lecture Notes in Artificial Intelligence No. 2159, Berlin: Springer Verlag, 2001. pp. 143-146. ISBN 3-540-42567-5.

Řehoř, D., Kadleček, D., Slavik, P., Nahodil, P. VAT - A New Approach for Multi-Agent Visualization. In 3rd IASTED International Conference on Visualization, Imaging and Image Processing, 8.-10. 9. 2003, Benalmadena, Spain. Spain: ACME Press, pp. 849 – 854. ISBN 0-88986-382-2.

Foltýn, L. Realization of Intelligent Agents Architecture for Artificial Life Domain. Diploma thesis. Prague: Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Cybernetics, 2005.

Mach, M. Data mining knowledge mechanism of environment based on behavior and functionality of it's partial objects. Diploma thesis. Prague: Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Cybernetics, 2005.