Construction of a Memory Management System in an On-line Learning Mechanism

F. Bellas, J.A. Becerra and R.J. Duro *

Grupo Integrado de Ingeniería - Universidade da Coruña
Ferrol - Spain

Abstract. This paper is the first of a two paper series that deals with an important problem in on-line learning mechanisms for autonomous agents that must perform non trivial tasks and operate over extended periods of time. The problem has to do with memory, and, in particular, with what is to be stored in what representation and the need for providing a memory management system to control the interplay between different types of memory. To study the problem, a two level memory structure consisting of a short term and a long term memory is introduced in an evolutionary based cognitive mechanism called the Multilevel Darwinist Brain. A management system for their operation and interaction is proposed that benefits from the evolutionary nature of the mechanism. Some results obtained during operation with real robots are presented in the second paper of the series.

1 Introduction

On-line learning in autonomous agents and robots is a very complex problem that has been addressed by many authors from different points of view [1], [2], [3]. Traditional learning architectures have usually been designed for a given set of problems or environments and in most cases only provide mechanisms for the limited modification of parameters (learning) according to a given preset value system. A more promising approach would be to introduce cognitive architectures where every element could be autonomously modified, including the value system. Thus, the challenge becomes how to construct such architecture without introducing too many designer mediated constraints. In this line Weng et al. [4] [5] developed an approach for autonomous mental development (AMD) and provided a description of the role that AMD should play in artificial intelligence. A similar concept is that of cognitive developmental robotics (CDR) [6]. The key aspect of CDR is that the control structure should reflect the robot's own process of understanding through interactions with the environment.

We have addressed the problem from a different perspective, making use of some of the concepts of traditional cognition, but introducing ontogenetic evolutionary processes for the on-line adaptation of the knowledge bearing structures.

The mechanism developed is called the Multilevel Darwinist Brain [7]. The base for the work is the establishment of the main features we would like to see in the agent's on-line operation. That is, the agent must autonomously extract the relevant information for the creation of all the models involved in its cognitive architecture, including its current value system through a satisfaction model, and discard the rest. In a certain way this is a sort of an attention mechanism over its sensorial inputs. In addition, depending on the circumstances it

[*] Work partially funded by MEC of Spain under project VEM2003-20088-C04-01 and Xunta de Galicia through project PGIDIT03DPI099E.

must be able to select the relevant models in order to cope with the current situation. In our case, and following a view of cognition similar to that of Walter Freeman [8], where the agent only perceives what it expects or predicts, the models it chooses for modelling the world, itself or the satisfaction it will achieve are determinant in what can be learnt as they provide the expectations used by the value system and the value system itself.

Furthermore, the agent must be able to transform data into knowledge creating subjective internal representations that are usable and can be accessed in the future. This means that the acquired knowledge must be used to adapt to repeated situations or to facilitate learning processes in new situations. In addition, the agent must be able to induce internal models extracting conclusions from guided behaviours.

From these requirements two elements become very important: on one hand, a mechanism that allows for the on-line learning of all the models and, at the same time provides timely proposed actions when the agent needs them, this is the function of the initial implementation of the Multilevel Darwinist Brain; on the other, there is a need for a dynamic memory management structure so that the information that is learnt can be reused and the learning process adapted to the circumstances and environments the agents find themselves in. This is the purpose of the work presented in this two paper series where we introduce a two level memory system with a Short Term Memory (STM) and a Long Term Memory (LTM) each one with its own type of information and management structure. In addition, we propose a dynamic mutual regulation mechanism so that their operation can be regulated to improve the performance of the agent.

The rest of the paper is structured as follows: Section 2 provides a brief overview of the Multilevel Darwinist Brain. Section 3 deals with the memory representations and management. Finally, section 4 draws some conclusions from this work. The second paper of the series provides a few examples of operation with this mechanism.

2 The Multilevel Darwinist Brain

The Multilevel Darwinist Brain (MDB) is a Cognitive Mechanism [9] that allows a general autonomous agent to decide the actions it must apply in its environment in order to fulfil its motivations. In its development, we have resorted to bio-psychological theories by Changeaux [10], Conrad [11] and Edelman [12] in the field of cognitive science relating the brain and its operation through a Darwinist process.

To implement the MDB, an utilitarian cognitive model [7] was adopted which starts from the premise that to carry out any task, a motivation (defined as the need or desire that makes an agent act) must exist that guides the behaviour as a function of its degree of satisfaction. From this basic idea, the concepts of action, world and internal models (W and I), satisfaction model (S) and action-perception pairs (set of values made up by the sensorial inputs and the satisfaction obtained after the execution of an action in the real world) are used to construct a cognitive mechanism. Its functional structure is shown in Figure 1.

Two processes must take place in a real non preconditioned cognitive mechanism: models *W, I* and *S* must be obtained as the agent interacts with the world, and the best possible actions must be selected through some sort of internal optimization process using the models available at that time. The main operation can be summarized by considering that the selected action is applied to the *environment* through the actuators obtaining new *sensing* values. These acting and sensing values provide a new *action-perception pair* that is stored in the action-perception memory (*Short-Term Memory* from this point forward). Then, the

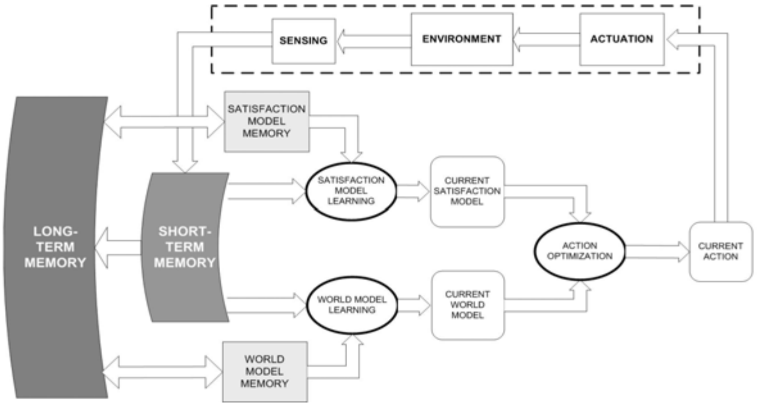

Fig. 1: Block diagram of the Multilevel Darwinist Brain

model learning processes start (for *world, internal* and *satisfaction models*) trying to find functions that generalize the real samples (*action-perception pairs*) stored in the *Short-Term Memory*. The best models in a given instant of time are taken as *current world* and *internal models* and *current satisfaction model* and are used in the process of optimizing the *action* with regards to the predicted satisfaction. When an action is needed, the best one obtained according to the models is applied again to the *environment* through the actuators obtaining new *sensing* values.

These steps constitute the basic operation cycle of the MDB, and we will call it an *iteration* of the mechanism. As more iterations take place, the MDB acquires more information from the real environment (new *action-perception pairs*). The models obtained become more accurate and, consequently, the action chosen using these models is more appropriate.

The model search process in the MDB is not an optimization process but a learning process (we seek the best generalization in time, not minimizing an error function in a given instant t). Consequently, the search techniques must allow for gradual application, as the information is known progressively and in real time. To satisfy these requirements we have selected *Artificial Neural Networks* (there is no restriction on the type of ANN) as the mathematical representation for the models and *Evolutionary Algorithms* as the most appropriate search technique. The algorithm we have used most often is the PBGA [13] although successful examples have been obtained with other Genetic and/or evolutionary strategies. Evolutionary techniques permit a gradual learning process by controlling the number of generations of evolution for a given content of the short-term memory (usually no more than four). The learning process takes place through input/output pairs using as fitness function the error between the predicted values provided by the models and the real values for each action-perception pair in the STM.

3 Memory in an on-line learning mechanism

Any cognitive architecture that provides learning capabilities needs to have some kind of memory structure in order to store past information that could be used in the learning process. In the MDB we have included a memory management system that considers two elements, the Short Term Memory (STM) and the Long Term Memory (LTM), and an interplay mechanism that connects their behaviour.

3.1 Short term memory

The Short Term Memory is a memory element that stores data obtained from the real time interaction of the agent with its environment. The internal models the agent creates during the learning process should predict and generalize all the data stored in the STM. Thus, what is learnt and how it is learnt depends on the contents of the STM during time. Obviously, it is not realistic to store all the samples acquired throughout an agent's lifetime. The STM must be limited in size and, consequently, a replacement strategy is required in order to store the information the agent considers more relevant in a given instant of time. The replacement strategy should be dynamic and adaptable to the needs of the agent and, therefore, it must be subject to external regulation. For this reason, we have designed a replacement strategy that labels the samples using four basic features related to saliency and temporal relevance:
The point in time a sample is stored (T): it favours the elimination of the oldest samples, maximizing the learning of the most current information acquired.
The distance between samples (D): measured as the Euclidean distance between the action-perception pair vectors, this parameter favours the storage of samples from all over the feature space in order to achieve a general modelling.
The complexity of a sample to be learnt (C): this parameter favours the storage of the hardest samples to be learnt. To calculate it, we use the error provided by the current models when predicting a sample.
The relevance of a sample (R): this parameter favours the storage of the most particular samples, that is, those that, even though they may be learnt by the models very well, initially presented large errors.

Thus, each sample is stored in the STM has a label (L) that is calculated every iteration as a linear combination of these four basic terms:

$$L = K_t \cdot T + K_d \cdot D + K_c \cdot C + K_r \cdot R$$

where the constants K_i control the relevance of each term. Thus, the operation of the STM can be regulated through the modification of these four parameters. Depending on the value of the constants K_i different storage policies can be obtained. The regulation of the parameters can be carried out by the cognitive mechanism or by other parts of the memory system so as to improve the learning and generalization properties. Here we will concentrate on memory interactions.

3.2 Long term memory

The Long Term Memory is a higher level memory element, because it stores information obtained after the analysis of the real data stored in the STM. From a psychological point of view, the LTM stores the knowledge acquired by the agent during its lifetime. This knowledge is represented in the MDB as models (world, internal and satisfaction models) and their context, so, the LTM stores the models that were classified by the agent as relevant in certain situations (context).

In an initial approach we have considered that a model must be stored in the LTM if it predicts the contents of the STM with high accuracy during an extended period of time (iterations in the MDB). If this happens, such model could be considered as acquired knowledge. We don’t want to store models obtained over equivalent Short Term Memories, that is, equivalent contexts. Thus, every time a new model is a candidate for inclusion in the LTM (it has been stable in its prediction of the STM), it is phenotypically compared with the rest of models in the LTM. This is done through their crossed predictionof their associated contexts (STM contents when they accessed the LTM) in order to decide if it is a new model or similar to an existing one. If they predict each other´s context well, they are taken as models of the same phenomena.

From a practical point of view, the addition of the LTM in the MDB, avoids the need of re-learning the models in a problem with a real agent in a dynamic situation every time the agent changes into different states (different environments or different operation schemas). The models stored in the LTM in a given instant of time are introduced in the evolving populations of the models of the MDB as seeds so that if the agent returns to a previously learnt situation, the model will be present in the population and the prediction will be accurate soon. If the new situation is similar to one the agent has learnt before, the fact of seeding the evolving population with the LTM models will allow the evolutionary process to reach a solution very fast.

3.3 Memory interplay

We have developed a mutual regulation system to control the interaction between these memories in the MDB. There are two main undesirable effects in the learning process that can be avoided with a correct management system.

First of all, as we mentioned before, the replacement strategy of the STM favours the storage of relevant samples. But what is considered relevant could change in time (change of motivation or environment), and consequently the information stored in the STM should also change so that the new models generated correspond to the new situation. If no regulation is introduced, when situations change, the STM memory will be polluted by information from previous situations (there is a mixture of information) and, consequently, the models that are generated do not correspond to any one of them.

These intermediate situations can be detected by the replacement strategy of the LTM as it is continuously testing the models to be stored in the LTM. Thus, if it detects a model that suddenly and repeatedly fails in the predictions of the samples stored in the STM, it is possible to assume that a change of context has occurred. This detection will produce a regulation of the parameters controlling the replacement in the STM so that it will purge the older context. It can even become a completely temporal strategy for a while. This purge will allow new data to fill the STM and thus the models can be correctly generated. It is a clear case of LTM monitoring affecting the operation of the STM and thus the process by which the models are generated.

The other undesirable effect we must avoid is the continuous storage of models in the LTM. This happens because the data stored in the STM are not general enough and the models seem to be different although they model the same situation. The replacement strategy of the LTM can detect if the agent’s situation has changed or not and, consequently, after a change of situation it can detect if the number of models attempting to enter the LTM is high. In such case, the parameters of the replacement strategy of the STM are regulated so that we favour information that is more general by empowering parameters such as distance, relevance or complexity and the reduction of the influence of time.

Using these two strategies in the interplay between memories together with the management mechanisms for each one of them individually, a dynamic memory structure arises that improves the efficiency in the use of memory resources, minimizing the number of models stored in LTM without affecting performance and allowing these models to be as general as possible. This last fact is quite important because, as models are used as seeds in the evolution processes, the more general they are the better they will adapt to new situations.

4 Conclusions

In this work we have presented a memory management system for the MDB that shows the importance of the interplay between memories in an on-line learning system. The data stored in the Short Term Memory are modeled and stored in the LTM as knowledge the agent can use in future learning situations. The replacement mechanisms of these memories are very interdependent as the parameters of the STM replacement strategy influence the type of models obtained in a stable manner and, consequently define candidates for LTM, and the models present in the LTM and their result over values in the STM regulate the way the STM acquires data.

With this operation schema applied to the MDB, we can obtain a behavior in the agent that shows two main features: the agent learns autonomously, paying attention to the relevant information of each situation and the agent is able to transform data into knowledge creating subjective internal representations that can be reused in the future. In the second paper of this two paper series we present a set of experiments of the application of this structure.

References

[1] Drescher, G. L., Learning from Experience Without Prior Knowledge in a Complicated World, *Proceedings of the AAAI Symposium on Parallel Models*, AAAI Press, (1988)

[2] Hayes-Roth, B., An architecture for adaptive intelligent systems, *Artif. Intel., 72* pp 329-365, (1995)

[3] J.S. Albus, Outline for a Theory of Intelligence, IEEE Transactions on Systems, Man and Cybernetics, Vol. 21, No. 3, (1991)

[4] J. Weng, Developmental Robotics: Theory and Experiments, *International Journal of Humanoid Robotics, vol. 1, no. 2*, 199–236, (2004)

[5] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur and E. Thelen, Autonomous Mental Development by Robots and Animals, *Science, vol. 291, no. 5504*, pp. 599 - 600, (2000)

[6] Asada, M., MacDorman, K. F., Ishiguro, H., Juniyoshi, Y., Cognitive Developmental Robotics as a New Paradigm for the Design of Humanoid Robots, *Rob. and Auton. Syst., V. 37*, pp. 185-193 (2001)

[7] F. Bellas, R.J. Duro, Multilevel Darwinist Brain in Robots: Initial Implementation, *ICINCO2004 Proceedings Book (vol. 2)*, pp 25-32 (2004).

[8] W. Freeman, How Brains Make Up their Minds, *Weidenfeld & Nicolson*, (1999)

[9] R. J. Duro, J. Santos, F. Bellas, A. Lamas, On Line Darwinist Cognitive Mechanism for an Artificial Organism, *Proceedings supplement book SAB2000*, pp 215-224, (2000)

[10] J. Changeux, P. Courrege, A. Danchin, A Theory of the Epigenesis of Neural Networks by Selective Stabilization of Synapses, *Proc.Nat. Acad. Sci. USA* 70, pp 2974-2978 (1973).

[11] M. Conrad, Evolutionary Learning Circuits. *Theor. Biol. 46*, pp 167-188 (1974).

[12] G. Edelman, Neural Darwinism. The Theory of Neuronal Group Selection. *Basic Books* (1987).

[13] F. Bellas, R. J. Duro,. Statistically neutral promoter based GA for evolution with dynamic fitness functions. *Proceedings of IASTED2002*, pp 335-340 (2002)