On Line Darwinist Cognitive Mechanism for an Artificial Organism

R. J. Duro, J. Santos, F. Bellas, A. Lamas

Grupo de Sistemas Autónomos
Universidade da Coruña
Spain
gsa@cdf.udc.es

Abstract

This paper is concerned with the presentation of an on-line cognitive mechanism for autonomous agents. The purpose of the mechanism is to allow autonomous agents, whether virtual or physical, to adapt to their environment and objectives without any external training. This is achieved through the use of their interaction with their surroundings in order to improve the level of satisfaction obtained. The mechanism is based on Darwinist principles and involves a two level concurrent operation of evolutionary processes. The first, or unconscious level, consists of evolutionary processes over models of the environment that are evaluated according to how good they are at predicting the next perceptions. This leads to a current, or conscious model, which is employed for evaluating strategies, using as fitness the level of satisfaction of the motivations of the artificial organism. Once a strategy is selected as the current one, it is carried out by means of the effectors that act on the real environment, returning a fitness measure for the evolution of the models. This cycle is repeated and, as time progresses, the models and strategies the organism works with become better adapted to the fulfillment of its motivations. This type of mechanism presents several advantages over other structures, such as its capability of obtaining original solutions and, at the same time, its capacity to make use of previous experience. In addition, if the environment changes, the mechanism will adapt efficiently.

1. Introduction

The work in the field of Artificial Life has concentrated on the search for simple artificial organisms that are capable of adapting to a changing environment with which they interact and where they must survive.

This search has been carried out from two points of view. On one hand, some researchers have contemplated the emergence of intelligent behavior as the result of the evolution of communities of simple organisms immersed in environments with different degrees of complexity to which they must adapt and where they must survive (Bourgine & Varela, 1991). This is the approach followed by evolutionary robotics (Harvey et al., 1993), when considering the behavior-based alternative (Arkin, 1998), where there is no explicit model of the environment the organism lives in.

From another point of view, perhaps closer to the traditional methods of function implementation, a procedure for the design of an artificial nervous system that controls the behavior of the organism - an insect in the case of Beer (Beer, 1990) - is sought. The design of nervous systems, although conceptually closer to the way in which humans have traditionally done things, presents the drawback that, in many cases, we are predetermining how the being must work, especially in the aspects of use of experience when facing new circumstances and in the generation of original solutions.

The problem could be formulated in the following terms: How can we provide our organisms with an underlying cognitive mechanism which does not predetermine their behavior or learning and which allows them to find creative and original solutions in an autonomous way?

Our approach to solving this question has been to provide an artificial organism with a darwinist cognitive mechanism that allows it to acquire in its life time a model of the environment or environments in which it lives and of the consequences of its actions over its internal state. This model is obtained and continuously updated through an evolutionary process.

There are few examples in the behavior based robotics approach and, in particular, in the evolutionary robotics literature, where explicit models of the environment the robot operates on are used. Watson (Watson, 1994) presents a system that, starting from some pre-trained building blocks and behavior sequences, when released into a real time environment adds its experiences to memory, building new behavior sequences, rules and procedures and deleting unused ones through a genetic algorithm. Nordin et al. (Nordin et al., 1998) employ a memory based genetic programming mechanism in order to obtain a cognitive architecture for a Khepera robot that makes use of previous experience in its interaction with the world. Basically, a planning process can incorporate a GP system that is used to evolve a suitable plan for the optimization of the outcome given the best current world model. In (Steels, 1995), the author introduces a selectionist mechanism, he calls *selectron*, for the evolution of new behavioral competencies of robotic agents. This mechanism may be used on-line and on a real robot as it operates on a changing environment.

These types of mechanisms do impose a certain subdivision of tasks in Minski´s sense, that is, a model of the world is constructed and it is used in order to test possible strategies before actually using them in the real world. Our work is directed in this line. It uses world models as an intrinsic part of the cognitive mechanism. Its advantage over traditional systems is that we provide a way of not predetermining the world models, these are evolved on line as the agent interacts with its environment. The inspiration of this mechanism comes from several proposals made by different authors on Darwinist models for the operation of the brain.

2. Evolutionary Learning Theories

It seems that we are always stringing ideas or representations, and when we reason we usually do it in the form of sequences with an origin and an objective. This sequentiality is also reflected in human language, whose rules are implicitly serial. On the other hand, when we think, according to Calvin (Calvin, 1987a), we project series of actions onto known scenarios. Each action of the series is put through our analyzer of effects on the environment, and as a function of the result it returns we seek the effect of the next action on the string. This process continues until we find a sequence that leads to the desired objective.

From another point of view, we have to consider that at the biological level the brain is made up of millions of neural circuits working in parallel. The problem resides in how do we go from one operating mode to the other, that is, from a massive parallelism to a conscious sequentiality, or, at last, a very limited parallelism, which allows us to interact with our environment in an orderly fashion, and how can we use this step in order to obtain computational power in real time that is as impressive as that of human or animal brains.

One set of theories that have practically been ignored by the artificial intelligence community are those based on evolutionary concepts at different time scales. According to these theories, neural learning and brain development processes strongly depend on an evolutionary base, with mechanisms such as selection and mutation. This Brain Darwinism (Calvin, 1987b) plays the same role in the brain in somatic time as in ecosystems in phylogenetic time.

There are many historical references that postulate more or less clear proposals in this line. Statements by James “To think is to perform selections” (James, 1909) or Spencer “If the doctrine of evolution is true, the inevitable implication is that the Mind can be understood by observing how Mind is evolved” (Spencer, 1986) already pointed in this direction.

Within the field of cognitive science there are four basic theories that relate the brain or neural structure with its operation from a darwinist, or evolution in somatic time, point of view. These theories are: the Theory of Evolutionary Learning Circuits (TELC) (Conrad, 1974, 1976); the Theory of Selective Stabilization of Synapses (TSSS) (Changeux et al., 1973) (Changeux & Danchin, 1976); the Theory of Selective Stabilization of Pre-Representations (TSSP) (Changeux et al., 1984), and the Theory of Neuronal Group Selection (TNGS) or “Neural Darwinism” (Edelman, 1987). For an excellent review see (Weiss, 1994). Most of the work presented here has been inspired on these insights.

In addition to the underlying brain mechanisms, when trying manage the operation of an organism we must consider the problem of motivation. For any organism to adapt to an environment or perform any form of activity, it needs some type of motivation that allows it to establish its objectives, provides it with a reason to spend energy and establishes some order in its representation of the environment.

One of the most interesting paths for the study of motivation presents the set of possible behavioral alternatives as competing for or time-sharing the attention of the organism depending on the levels of drive associated to each one of them (Heiligenberg, 1974). A priority scheme exists whereby some types of stimuli (pain, for example) are prioritary over other types. In addition, the behavioral priorities can be adjusted so that the urgencies of the needs of the organism are balanced with respect to the opportunities of the environment and their quality (Gould & Marler, 1984).

Until recently, the motivations most computational systems had in order to carry out the tasks they where assigned where external motivations. A programmer established what motivated the action of the system in a direct manner, as in traditional programming where the motivation is implicit in the writing of the program, or in an indirect manner by means of a fitness function, as in the case of genetic algorithms. These externally set objectives imply a clear rigidity when trying to achieve adaptability, leading to stereotyped behaviors and standard solutions to the problems. This is assuming that the external fitness function could take into account the inherent variability of dynamic environments. What these types of solutions certainly do not provide is a complete autonomy of the organism.

We believe that for an organism to be autonomous and survive in changing environments it must contain its motivations within itself. In the case of animals, natural evolution and selection has eliminated those strains of the evolutionary tree whose members did not contain the necessary motivations in order to adapt to the environments and circumstances that arose, leaving only those that did contain these internal motivations.

The basic motivations are translated into drives for performing certain actions or behavioral patterns. These drives activate certain “programs”, which may be genetically encoded or learnt and which direct the actions of the individual, this is, they focus its attention on those aspects of the environment or actions that are relevant for completing the programs associated with the drive, and which will lead to placating its motivation. Different drives may conflict in an organism in a given moment of time, and for it to be able to operate there must be a mechanism for solving these conflicts.

Having given a small review of the theories and concepts that inspired our proposal and stated the purpose of the model, that is, to provide our artificial organisms with an underlying cognitive mechanism which does not predetermine their behavior or learning and which allows them to find creative and original solutions in an autonomous way, in the next sections we are going to describe the basic blocks of our model and the way they interact. In section 3 we provide the basic definitions on which our model is based. Sections 4 and 5 are devoted to describing its structure, the different components of the model as well as their interactions. In section 6, we present examples of use of the cognitive architecture in a simulated artificial organism. Finally, we provide some conclusions.

3. Basic Definitions and Structure

Our interpretation of an organism internal cognitive operation and its external behavior is going to be based on two basic concepts: Models and Strategies.

DEFINITION 1. Model: Abstraction that using the sensory inputs in time t, the strategy applied in time t and the state of the system in time t, permits determining the sensory inputs in time t+1 (whenever predictable) and the state of the system in time t+1.

DEFINITION 2. Strategy: Set of motor commands that imply actions of the organism in the environment in a given sequence.

If what we want is an organism whose actions are not simple one to one reactions to its inputs, it is necessary to establish some mechanism that allows it to plan its actions. To plan implies to be able to predict the consequences of some facts or actions occurring in time t on the perceptions in time t+1 (or t+n in general). To predict unavoidably leads to the implicit or explicit existence of a model corresponding to that about which we want to predict. In the case of our organism, it needs to make predictions about its environment and itself.

In order to be able to operate, an organism requires at least two types of models:

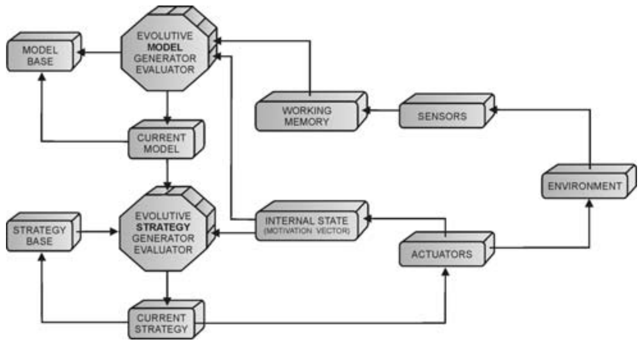

Figure 1: Block diagram of the cognitive mechanism

1- Model of the environment: which allows it to know what it is going to perceive in time t+1 from its actions in time t and the environment in time t. That is, this model permits evaluating the consequences of strategies on the environment and obtains answers to internal questions of the type, what would happen if...? It leads to the possibility of establishing strategies consisting of sequences of several actions. In addition, this model will make it possible for the organism to predict the probable evolution of the dynamic environment in which it operates in order to take advantage of it whenever possible.

2- Model of itself: from its state in time t, the sensory input in time t and the strategy it applies, it allows the organism to evaluate its own state in time t+1. This model of itself provides a means for the organism to establish the probability of satisfying its internal motivations and a prediction of the extent to which they will be satisfied.

In any interaction with the environment, the organism perceives a series of stimuli through its sensors, which determine the model of the environment it uses to predict as a function of its motivations. At the same time, it must be able to observe the consequences of its actions on itself and how close it comes to satisfying its motivations. This is achieved by means of its model of itself. Finally, it needs to be able to generate actions in order to satisfy these motivations. This is achieved by means of effectors, driven by the strategies, which are series of actions the organism will generate.

Thus, as shown in figure 1, the cognitive mechanism of our organism is going to contain three types of mental representations in two different levels. On one hand it is going to contain a representation of possible strategies, on the other a set of models of the environment and itself and finally, a representation of its motivations.

The problem resides in how to combine strategies, models, sensors, effectors, motivations and environment in order to generate a closed set which allows the organism to survive. This combination implies a model of interaction among all of these elements which permits fluid real time operation and which at the same time verifies the requirements we have stated, that is, possibility of learning, creation of original solutions, etc...

Making use of the bio-psychological theories we have commented in section 2, we are going to establish a two level hierarchy. The first level consists of sets of unconscious parallel processes that, after a brief processing stage, lead to a second "conscious" level by selection. In the following sections we analyze the different components of the model and their interactions.

4. Components of the Cognitive Mechanism

The organism perceives the environment by means of its sensors and acts on it through its effectors. In principle, the models, strategies and motivations are isolated from a direct interaction with the environment. The models receive information from the environment by means of the sensors and the strategies produce actions on the environment by means of effectors.

It seems evident that the internal image any organism has of its environment must be given as a function of the sensors it is endowed with. Also, its image of itself, that is, its state is going to be determined by its state sensors. If the organism is very simple and as single internal sensor it has a binary HUNGER (yes/no) sensor, as single external sensor it has a FOOD AHEAD (yes/no) sensor and as effectors it has a WALK FORWARD and a TURN LEFT effectors, its representation of the world will be given as a function of these four terms, it will not appreciate colors, or smells. Taking the analogy to an extreme, its concept of beauty could perhaps be FOOD AHEAD YES, and its concept of happiness could be something like HUNGER NO.

Taking this into account, the models of the environment our organism generates will only contemplate the information it has about it, that is, the information provided by its sensors and in the format (level of integration and pre-processing) in which they provide it. On the other hand, the strategies are going to be constrained to the actions the organism can perform on the environment. These two points are very important as they clearly show that an organism is limited by its sensors and effectors. These are going to determine to a large extent the organism, especially its ability to adapt to the environment, and the way it represents and processes information.

The question now is how we relate models, strategies and motivations. A model must allow us to try the effects of possible strategies on the environment, on the organism and on its motivations without actually carrying them out (one way of thinking). In this sense, the strategies must be "tested" in the models and the models must be capable of predicting their effect on the environment, on the organism and on its motivations.

The case is that simply testing strategies or models is not really very useful. An organism must be able to generate different strategies and as a function of its motivations select the best possible one in order to achieve its objectives. This testing of strategies, depending on their length (how many steps into the future they consider) may make intensive use of the models. These must take data in time t, and possible strategies in order to generate environments in time t+1, which will be used as inputs for the generation of environments in time t+2 with the next step of the strategy, and so on...

Following the ideas of the models presented in section 2, we are going to employ a darwinist strategy in order to endow our organism with the necessary cognitive mechanisms for it to benefit from a massively parallel processing model and, at the same time, maintain a conscious sequentiality in its interactions.

The prototype of darwinist strategy in computation are genetic algorithms and evolutionary techniques (Holland, 1975). These methods are based on the generation of a number of representations or encodings of possible solutions to a problem (random at the beginning). These are evaluated using a fitness function that arranges them according to their fitness for solving the problem. Out of this set of solutions, the parents of the next generation are selected from the most fit and they mate, that is, they generate offspring by means of some type of combination of the genotype of the parents. These offspring undergo some random mutations and the process is repeated until an optimum solution is achieved.

When defining a GA we must be capable of evaluating that which we are going to evolve, and we must define a time scale or speed at which we want it to evolve. In our case, the two elements of the mechanism that must undergo an evolutionary process are the strategies and the models.

The treatment given to these two elements is going to be different. On one hand, as we have already pointed out, a model is evaluated by how good it was in predicting the inputs in t+1. This can be easily carried out by establishing some type of error function that relates predicted and real inputs for a given environment model and the predicted and real state in the case of the state, as we mentioned in previous section. This function can be the difference between them weighted by their relevance. That is, in the case of the models, the interaction with the environment, through the sensory inputs, will evaluate the set (population) of models in the organism. The fitness of a model is calculated by determining how good it was at predicting the inputs the agent has just obtained from the information it had before they were obtained. Between interactions with the environment we carry out some GA generational steps (selection, procreation and mutation) as a function of the evaluation, that is, using as a target the previous inputs. In each moment of the interaction we will select the model with the highest fitness value as the current model. Observe that by using this mechanism there is a massively parallel processing of many combinations of models and a selection process that leads to the current model, that is, to the model the organism is conscious of and which it uses for the evaluation of strategies, as indicated in figure 1.

In the case of strategies, the mechanism is similar, but now the strategies are evaluated using the current model and the fitness criterion is the predicted satisfaction of the motivational index. This index will be some type of combination of the elements of the motivation vector. Consequently, the fitness of a strategy is the resulting predicted satisfaction of motivations after applying the strategy, previous state and previous sensorial inputs to the current model. The set of strategies (the population) will undergo some GA generational steps and the strategy providing the greatest satisfaction will be the one selected for its execution. Its commands will be sent to the effectors.

Summarizing, the models are evaluated by the environment through the sensory inputs and the strategies by the current model through the predicted satisfaction of the motivations. This basic cycle will be repeated indefinitely and establishes the process for performing a parallel processing scheme leading to sequential solutions with very few constraints regarding the type of solutions that may result. It also provides the opportunity for generating new unprogrammed solutions and the capacity for predicting and planning.

If we consider the possibility of storing solutions that have worked satisfactorily, both in the case of strategies and models, and use them in new GA processes as seeds in the populations, we establish a mechanism that permits combining old solutions (seeds) with new elements (mutation and randomly generated solutions) so that original solutions and experience based solutions can be obtained without the problems or circumstances having to be identical. A combination of solutions is simply selected if the model says that its quality is going to be the best, independently of the problem. If the storage space is limited, there will have to be some type of forgetting mechanism, but that is the topic of another paper.

5. Notes on Implementation

Once the general operation of the cognitive mechanism has been presented, it is necessary to comment the base for the models, strategies and motivations. In the cognition and learning theories we have considered, the authors talk about selection of neural circuits, both during the formation of the brain and during learning and the selection of outputs in the adult brain. Continuing with the analogy, the "chromosomes" of the GA models can well correspond to artificial neural networks. Anyway, and despite the fact that biologically this is the most plausible representation, and perhaps from some computational points of view something very appropriate, especially because of their local learning capabilities, in principle there is nothing against them being any other type of information structure. The only requirement is that whatever the structure, it must be able to take some inputs and produce some outputs. Consequently, they could well be, in some cases, functions or production systems. In fact, there is no reason, except for complexity, why they cannot be different formats at the same time. In the examples presented we have made use of rule systems and neural networks for the models.

With respect to the strategies, they are sequences of commands to the effectors. Therefore they can be simple command lists interpreted by the effectors (which is the case here), or any other type of structure, with perhaps the only requirement that they must be chainable so that new strategies can be formed from parts of older ones.

In the case of motivations, they may be taken as one more sensor, which perceives a value as a function of hunger, thirst, pain (it can be structured as a pain matrix specifying different areas of the body) or whatever we want to include. These motivations are like containers which when the level is below a certain threshold implicitly or explicitly send messages of alarm that become stronger as the containers become emptier. These container levels can be simple numerical values sent to the internal models or to previous processing units which combine, select or relate them in some way to the drive they induce, that is, to their relevance in the fitness functions.

6. Application Examples

In this section we are going to present a simple illustrative application of the cognitive mechanism discussed in previous sections. From this point on we will talk about *world models*, including in this description both the models of the environment and the internal models. A *world model* is a function that takes as inputs the environment as perceived by the agent, its internal state and the action taken in instant t and produces as outputs the predicted perception and internal state in t+1 as described above.

We will use two different approaches to develop our example. In the first one, the world models will be encoded as simple rule lists, direct representations of the real world. In the second case, the models will be implemented as neural networks whose weights will undergo the evolutionary

External perception (t)	Internal perception (t)	Strategy	External perception (t+1)	Internal perception (t+1)
2 bits	3 bits	3 bits	2 bits	3 bits

Figure 2: Binary encoding of the rules used in the model of world. The external perception has 4 possible values (00, 01, 10, 11) as a function of the height sensor. The internal perception is just the position of the three legs (up or down, 1 or 0) and the strategy in this case is a simple action (movement of the legs) so each bit is the position of each leg (1 or 0).

process.

We have considered a three-legged robot in a very simple environment. The robot must learn by itself to stand up on its three legs starting from any configuration, much like a baby when it is learning to stand. The example is quite simple from an application point of view, but it is useful in order to illustrate how the mechanism works and how it really adapts to changing environmental conditions. More complex examples would hide the operation of the mechanism.

6.1. Rule Based World Models

In the beginning of the process the robot has no clue on what its environment is. It has a sensor that indicates the height of its body and three actuators that move the legs. In this first case the legs have just two possible positions, up and down, which have been encoded using a binary representation. Obviously, before the robot can do anything about obtaining the highest possible satisfaction level (getting up) it must establish the relationships between actions, perceptions and internal state so that an informed action can be taken in order to achieve complete satisfaction. It must obtain a good model it can use in order to evaluate possible strategies.

We have encoded each model as a set of rules that describe the world. In figure 2 we display one of these rules and its encoding. Each rule represents the conditions to be matched in each particular situation (the particular inputs and action) and the consequences of the rules that encode the new predicted readings in the external and internal sensors. The initially random model base has a population of 2000 individuals, each one made up of 832 genes. They are distributed in 64 rules of 13 elements (see figure 2). In the case of the strategy base, we encoded each strategy as a sequence of three actions.

The first step in the mechanism is the evolution of the model base. After a fixed number of generations we select a world model and use it to test the strategies in the evolution of the strategy base. After the evolution of the strategies we select one and it is executed in the real world.

As shown in figure 3, when the robot starts to operate in its world, as all the models are random, its information about the world is basically nil. Consequently, any strategy it decides to carry out based on these models will be no more than a random motion of its legs. Through repeated random actions, the robot can start obtaining associations of actions and their consequences on its internal and external perceptions and thus use these pairs in order to evolve the model population. As this evolution progresses, the models become increasingly better and thus, the current model represents the real world in a more accurate manner. Consequently, the actions (strategies) selected as good using this model become more appropriate as time progresses. Finally, a moment comes when the representation of the world provided by the current model is good enough for the robot to be able to select the optimal strategy in order to achieve its objective (in this case, stand at its maximum height). During the evolution, the tripod acquires wrong leg positions, as shown in figure 3, until it reaches the desired one with the three legs up (we use this configuration as the starting point in figure 5a). Each iteration of the mechanism implies 20 evolutionary generations for the models and 4 generations for the evolution of strategies.

Figure 4 represents the difference between the representation provided by the current model each instant of time and the real world in each iteration of the evolutionary process. This figure is included only for the purpose of explaining the evolution. In the on-line learning of the model, the genetic algorithm is not aware of the complete real world, only of the difference between the predictions made by each model in the population and the real inputs once an action has been carried out. To accelerate evolution, we have introduced a short span of memory containing a few prediction-real value pairs obtained in previous instants of time. Thus, the fitness function for the models includes the last four results (the last four rules) obtained from the mechanism. This makes evolution smoother as the fitness function only changes slightly between two iterations. For each model, its fitness is calculated as the sum of the known fitness of each one of its rules. Usually, as the robots operate in unknown environments and are only aware of the results of the actions they have carried out, only a few of the rules have been evaluated. In addition, the persistence of the evaluated values of rules is rather short due to the fact that as evolution progresses they undergo crossover and

Figure 3: Sequence of five consecutive actions for the tripod robot while the models are random. The position of the legs will change continuously until the models become correct and the desired configuration (three legs up) is achieved.

mutation, which destroy this information. This is not necessarily bad, as this destruction allows the system to generate new alternatives in a guided manner, thus permitting the exploration of new possibilities for the models and strategies. Having perfect persistence of the evaluation of rules would lead to a static representation of the world, with no possible improvement.

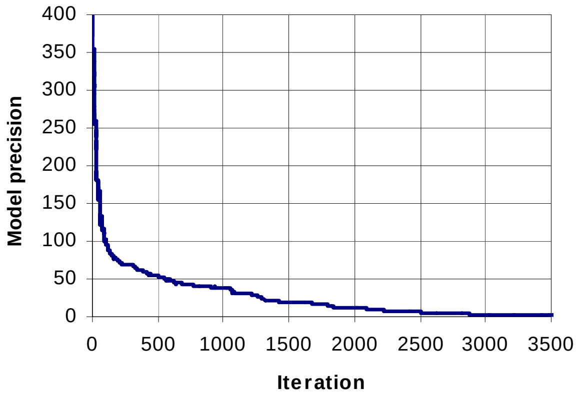

Figure 4: Difference between the model representation and the real world

This process has allowed the robot, as shown in figure 5, to obtain a creative solution to its problem. It had no clue on how to achieve its optimal state, and through interaction with the world it has obtained a model of this world that is good enough to evaluate strategies and thus obtain the optimal one.

We have also wanted to test the ability of the mechanism to make use of previous experience in order to generate new solutions. Previous experience is stored in the model base and in the strategy base. The models in the base have evolved to be a good representation of worlds the robot has been inserted in, and the strategy base includes strategies that have been more successful in previous instants of time. Consequently, if we change some parameter in the world or the objectives of the robot, it will start the evolution of new models or strategies from those in the base, thus obtaining a good solution for the slightly modified new world in a much shorter time, as shown in the sequences of figure 5. As we mention before, the first goal for the tripod was to stand on its three legs. This was obtained in 2500 iterations considering that if in 10 consecutive iterations (200 generations of model base evolution and 40 of strategy base evolution) the leg positions did not change, this position was stable. In figure 5, we show 3 different examples in which the goal was changed. The first five images (figure 5a) show the tripod starting from the previous goal (three legs up) and how it reaches a new one (left and center legs down, right leg up) in just 2 iterations. The tripod maintains this new goal until

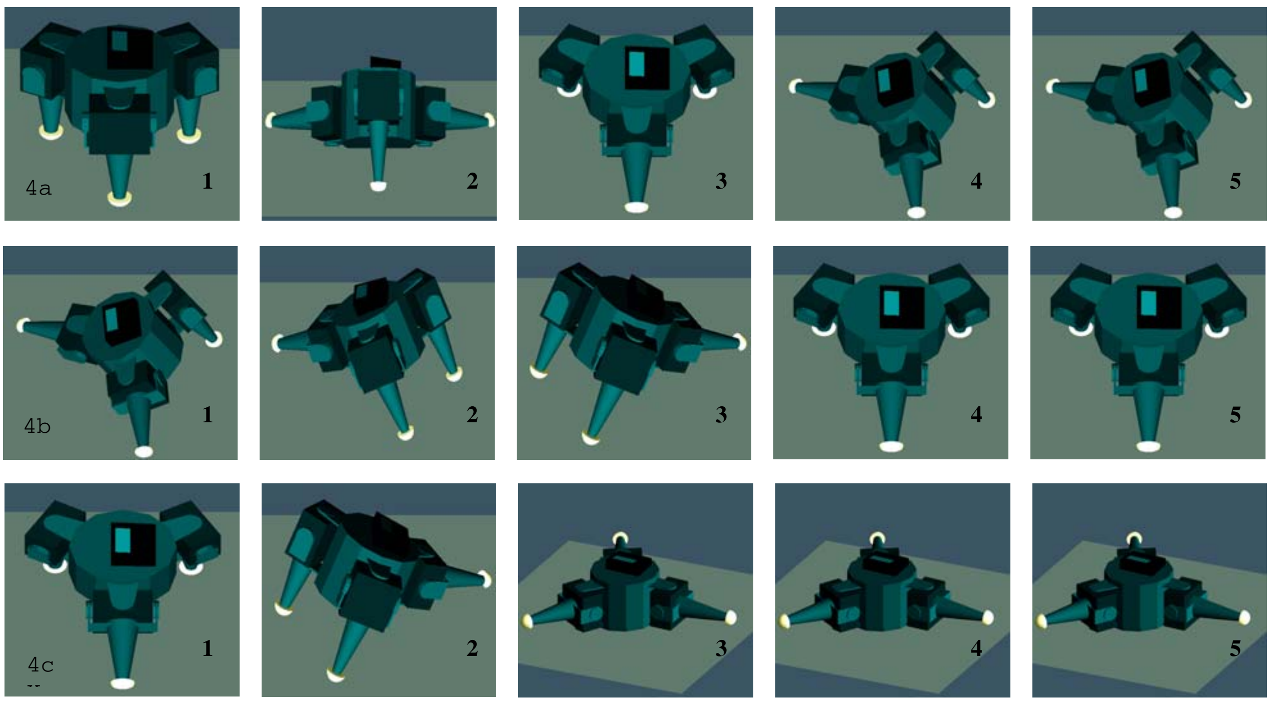

Figure 5: Three different sequences of five consecutive actions for the tripod robot, starting from the previous position. In figure 5a the starting position is: *left-leg* **up**, *center-leg* **up**, *right-leg* **up** and the goal position is: **down-down-up** reached after two intermediate iterations. In figure 5b the starting position is: **down-down-up** and the goal is**: up-down-up** reached again in two iterations. Finally, in figure 5c the starting position is: **up-down-up** and the goal is **down-down-down** reached in just one intermediate iteration

Internal perception (t)			Strategy			Internal perception (t+1)			"Satisfaction"
1.0 - 10.0	1.0 - 10.0	1.0 - 10.0	-1.0 – 1.0	-1.0 – 1.0	-1.0 – 1.0	1.0 - 10.0	1.0 - 10.0	1.0 - 10.0	-3.0 – 3.0

Figure 6: Continuous encoding of the values used to obtain the world model. The internal perception takes values from 1.0 to 10.0 as a function of the leg height (1.0 means down and 10.0 high) and the strategy is an incremental action (movement of the legs) from the previous position (given by the internal perception). The first value of the strategy is related to the first value of the internal perception and so on.

we change it 10 iterations later. The next five images (figure 5b) show the tripod starting again from the previous position and how it reaches the goal through two wrong intermediate positions. The last sequence of five images (figure 5c) show a better result needing just one iteration to achieve the desired position.

6.2. World model using neural networks

Instead of using rule based world models, in this example we have developed a model base using neural networks. Now, as indicated in figure 6, the legs perform relative motions (encoded continuously from –1 to 1) from its previous position (internal perception). Consequently, the robot can move up or down by small amounts. We have introduced a new field in t+1and it's a value of satisfaction for the robot (how close it is to the goal).

The world models are neural networks with 6 inputs (internal perceptions and strategy in t) and 4 outputs (predicted internal perceptions and satisfaction for t+1). We evolve the weights of the neural networks using as fitness function the similarity, in the form of an Euclidean distance, between the values predicted by the network and those that were obtained in reality. Thus, the network now must obtain the function that relates values in t with values in t+1.

The strategies base is similar to the previous example because we evolve the strategies directly (as action strings). The difference is that in this case the values are continuous from –1 to 1.

The mechanism works as follows: starting from a real (but random) set of values we evolve the models of world (weights of the network) using this set of real values in the fitness function. We take the first 6 values of the set and introduce them in each neural network. The difference between the output given by the network and the last 4 values of the set is the fitness value for each individual. After a fixed number of generations, we select the best network (the best world model) and we use it to evaluate the strategies during the evolution of the strategy base. Each strategy (each individual in this second evolution) is applied, together with the values for the internal perception in t, to the 6 inputs of the selected network, and in this case the fitness of that strategy is the satisfaction given by the output of the network. Again, after a fixed number of generations we select one strategy, the one that is predicted to produce the highest satisfaction, and execute it in the real world. This will result in a new perception in t+1 and a new value of satisfaction, so we have a new set of values (real behavior) to use in the fitness function corresponding to the evolution of the world models in the next iteration of the whole mechanism.

One must be careful when implementing this mechanism because the fitness function for the world models is changed each iteration with the world (each moment of time the agent works on local objectives and perceptions) and this may cause the evolutions of the models to oscillate. To address this question, we must prevent the model base from evolving too much between two interactions with the real world. If the model base evolves too much, when we change the fitness function the best individual (model) will probably change and the solution oscillates. On the other hand, if we do not evolve the models enough between interactions with the world, there will be no convergence. We expect about 2 or 4 generations of evolution to be enough with each set of values (fitness function) to obtain a smooth convergence.

The goal of this example is the same as before: the robot must learn to stand up on its three legs starting from any position. The main difference is that now the number of possible values for the internal perceptions and actions are infinite (continuous) and the search space is much bigger.

To obtain these results we have used an ANN with 6 input neurons, two hidden layers (each one with 8 neurons) and 4 output neurons. The model base had a population of 1000 individuals and the strategies base had a population of 100 individuals. Each iteration implied 2 generations of evolution in both cases.

In figure 7 we present some results corresponding to the position of the robot. A position of 30 implies the robot standing straight on its three legs and position 3 implies the robot body is touching the ground and all three legs are up. Before being able to obtain a good result, the robot must explore the world. This was achieved by changing its objective often for a while until it had a chance of being presented with different starting positions and actions. On the figure this is what takes place up to interaction point 1100. Obviously, during this first stage, the motion of the robot is quite erratic as it is basically exploring the consequences of its own actions on its environment (perceptions) and satisfaction.

The second section (section B) of figure 7, corresponds to the operation of the robot when its objective has been set to get up on all three legs as high as possible. That is, satisfaction increases with height. After a few doubtful interactions with the world, the robot learns this relationship between satisfaction and actions and goes straight into this state.

To test the adaptability of the mechanism, we now change the objective of the robot so that maximum satisfaction is achieved when it is flat on the ground (height 3). This implies a new mapping between actions, previous state and satisfaction although, if it has learnt the model of the world correctly, it does not imply a new mapping between actions and perceptions. Consequently, the time required to obtain strategies that are appropriate for this objective should be reduced. This is clearly seen in part C of figure 7. Finally, just to show that this adaptability not only works for the extremes, we allowed the robot to obtain maximum satisfaction when it reached a height of 21 (part D).

7. Conclusions

In this paper we tried to provide a first indication of how a cognitive operation model for an artificial organism based on a darwinist mechanism could work so as not to predetermine de behaviors of the organism. This model is based on a two level hierarchy of processes. On one hand we have an unconscious level, which generates models of environment and internal state using genetic algorithm processing techniques. Through an evaluation of the models using the sensory information we choose a current model that is employed, in a conscious level, for the evaluation of the strategies, which are selected as a function of how well they satisfy the motivations of the organism. Once the strategy has been selected it is applied and the whole process starts again. This provides an "evolutionary learning" mechanism that may be employed on-line in dynamic environments, where the final model is a summary of the correlation of the information the organism has considered relevant for the required behaviors, unlike traditional simulation in robotics, where the designer determines the previous simulations of environment and robot.

For the learning that takes place not to be lost, it is necessary to store the models and strategies that were successful and use them as seeds in new evolution processes. This seeds, when appropriately used, are a way of introducing experience into the reasoning process, and the random genetic mutation and the generation of random elements of the populations of models and strategies are a way of generating variety and new models and strategies or parts of them. These two processes give way to original solutions and experience based reasoning.

This very simple model can be extended by using evaluation functions with memory, that is, evaluation

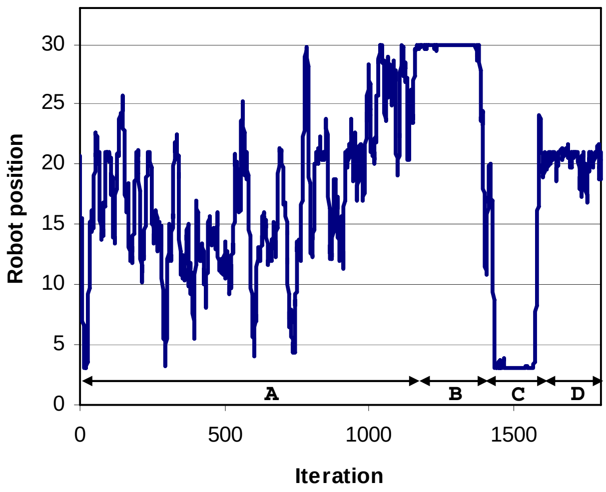

Figure 7: Height of the tripod during 1800 steps in the interaction with the world. In area A the robot was provided with different objectives so as to explore the environment. Area B corresponds to an objective of 30 (maximum satisfaction for position 30. For area C the objective was changed to 3 and finally in area D the objective was 21.

functions for the models that take into account series of inputs or evaluation of strategies using several models and selecting as a function of combined fitness. The evaluation functions for the models can be weighed by the motivations so that a given motivation in a given degree leads to models that are more selective towards certain perceptions. We can even provide more complex hierarchical structures with models and submodules, the former combinations of the later .The same can be applied to strategies and substrategies.

Acknowledgements

This work was funded by Xunta de Galicia under project PGIDT99PXI10503A.

References

Arkin, R.C. (1998). *Behavior Based Robotics*. MIT Press, Cambridge, MA.

Beer, R.D. (1990). *Intelligence as Adaptive Behavior. An Experiment in Computational Neuroethology*. Academic Press.

Bourgine, P. and Varela, F.J. (1991). Towards a Practice of Autonomous Systems. *Proceedings of the First European Conference on Artificial Life*, MIT Press.

Calvin, W.H. (1987a). *Whole Earth Review 55*, pp. 22-28.

Calvin, W.H. (1987b). The Brain as a Darwin Machine. *Nature*. Vol. 330.

Changeux, J.P., Courrege, P., and Danchin, A. (1973). A Theory of the Epigenesis of Neural Networks by Selective Stabilization of Sinapsis, *Proc. Natl. Acad. Sci*. USA 70, pp. 2974-2978.

Changeux, J.P., and Danchin, A (1976). Selective Stabilization of Developing Synapsis as a Mechanism for the Specification of Neural Networks, *Nature* 264, pp. 705-712.

Changeux, J.P., Heidmann, T., and Patte, P. (1984). Learning by Selection. *The Biology of Learning*, P. Marler and H.S. Terrace (Eds.), pp. 115-133, Berlin, Springer-Verlag.

Conrad, M. (1974). Evolutionary Learning Circuits. *J. Theor. Biol.* 46, pp. 167.

Conrad, M. (1976). Complementary Molecular Models of Learning and Memory, *BioSystems* 8, pp. 119-138.

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

Gould, J.L., and Marler, P. (1984). Ethology and the Natural History of Learning, *The Biology of Learning*, P. Marler and H.S. Terrace (Eds.), pp. 47-74, Berlin, Springer-Verlag.

Harvey, I., Husbands, P., and Cliff, D. (1993). Issues in Evolutionary Robotics, J-A. Meyer, H. Roitblat, and S. Wilson (Eds.). *From Animals to Animats* 2. *Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB92)*, MIT Press, Cambridge, MA, pp. 364-373.

Heiligenberg, W. (1974). Processes Governing Behavioral States of Readiness. *Adv. Study Behav*. 5, pp. 173-200.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*, Ann Arbor : MIT Press.

James, W. (1909). *Précis de Psychologie*. Paris. Marcel Rivière.

Nordin, P., Banzhaf, W., and Brameier, M. (1998). Evolution of a World Model for a Miniature Robot Using Genetic Programming. *Robotics and Autonomous Systems*, Vol. 25, pp. 105-116.

Spencer, H. (1986). *The principles of Psychology*, 3rd ed. New York: Appleton.

Steels, L. (1995). Emergent Functionality in Robotic Agents through On-line Evolution. *Artificial Life* IV, pp. 8-14.

Watson, J.B. (1994). Behavior-Based Control for Autonomous Robotics. *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, Sebald, A.V. and Fogel, L.J. (Eds.), World Scientific, Singapore , pp. 185-190.

Weiss, G. (1994). Neural Networks and Evolutionary Computation. Part II: Hybrid Approaches on the Neurosciences. *Proceedings of the First IEEE Conference on Evolutionary Computation*, Vol. 1, pp. 273-277, Florida