Multilevel Darwinist Brain (MDB) Lifelong Learning by Evolution in Real Robots

Priya Vinod
Assistant Professor,
Department of Computer Science & Application,
Mercy College, Palakkad

Abstract-- **A cognitive architecture is a blueprint for intelligent agents. Cognitive architecture is a modular structure that is composed of a number of multilevel interconnected components, each with distributed, module specific memory. Modular structure allows it to be transported across platforms. Multilevel Darwinist Brain or MDB is a cognitive architecture that enables robots to have lifelong learning. It has been tested and proved with on-line learning robots. With help of MDB it is possible to achieve the operation real time. Hence with the help of this current architecture it is possible to implement a real robot with lifelong learning capabilities and with minimal intervention from the designer of the system. MDB uses evolutionary algorithms in the knowledge acquisition process which means that computational implementation must be efficient. This approach considers evolution, especially neuro evolution which helps robots to learn tasks and objectives. MDB has Charles Darwin's name included within, because he proposed the Theory of Evolution. Even though here evolution of animals is not considered, rather another approach will be discussed. MDB is not intended as a biological entity rather it is a computationally effective way of providing the required functionality in real-time.**

I. INTRODUCTION

Cognition may be defined as “The mental process of knowing, including aspects such as awareness, perception, reasoning, and judgment”. A cognitive robot is characterized by its capacity of acquiring knowledge in an autonomous and adaptive way, that is, a cognitive robot has adaptive learning capabilities and, as a consequence, its behaviour is really autonomous. It seems that, in autonomous robotics, instead of designing intelligent robots, researchers for a long time have been designing just controllers.

From a computational perspective, cognition can be considered as a collection of emerging information technologies inspired by the qualitative nature of biologically based information processing and information extraction found in the nervous system, human reasoning, human decision-making, and natural selection. There is a process of extracting models from data. These models must be used in order to decide the appropriate actions so as to fulfill the robot’s motivations. Cognitive architectures differ from each other by, how the model is made and the process of determination of an action.

Selection of actions is in semantics, to explore the action space that tries to fulfill the motivations or objectives, that is, finding an optimal action for the agent. A cognitive mechanism must involve some type of optimization strategy or algorithm.

Taking the MDB one step further, incorporated into the cognition is the the concept of behavior controller as a way to implement actions and behaviours on the part of the robot. Behaviors that are repeatedly and successfully used will become reflex in terms of not having to go through the whole cognitive process in order to be activated under particular circumstances.

To manage all these learning processes in a practical way, the cognitive developmental robotics (CDR) approach is followed. The main objective of the cognitive developmental robotics (CDR) field is to create open-ended, autonomous learning systems that continually adapt to their environment, as opposed to constructing robots that carry out particular, predefined tasks.

The developmental process consists of two phases: the individual development at an early stage and the social development through interaction between individuals later on.

When dealing with physical robots that interact in real environments and, consequently, must learn in real time, the number of cognitive architectures that can be found in literature decreases. Although, there are models like SASE which is based in the autonomous creation of models (using Q-learning and Partially Observable Markov Decision Processes) from the sensory information obtained with the external and internal sensors that are used to fulfill the objectives of robots.

II. UTILITARIAN COGNITIVE MODEL

A cognitive system makes no sense without its link to the real (or virtual) world in which it is immersed and which provides the data to be manipulated into information and knowledge, thus requiring the capability of acting and sensing and of doing so in a timely and unconstrained manner.

In this case, a utilitarian cognitive model has been adopted which starts from the premise that, to carry out any task, a *motivation* must exist that guides the behavior as a function of its degree of satisfaction. The external perception *e(t)* of an agent is made up of the sensory information it is capable of acquiring through its sensors from the environment in which it operates. The environment can change due to the actions of the agent or to factors uncontrolled by the agent. Consequently, the external perception can be expressed as a function of the last action performed by the agent *A(t-1)*, the sensory perception it had of the external world in the previous time instant *e(t-1)* and a description of the events occurring in the environment that are not due to its actions *Xe(t-1)* through a function *W*:

$$e(t) = W\,[e(t\text{-}1), A(t\text{-}1), Xe\ (t\text{-}1)]$$

The internal perception *i(t)* of an agent is made up of the sensory information provided by its internal sensors. Internal perception can be written in terms of the last action performed by the agent, the sensory perception it had from the internal sensors in the previous time instant *i(t-1)* and other internal events not caused by the agent's actions *Xi (t-1)* through a function *I*:

$$i(t) = I\,[i(t\text{-}1), A(t\text{-}1), Xi\ (t\text{-}1)]$$

The satisfaction *s(t)* of the agent can be defined as a magnitude or vector that represents the degree of fulfillment of the motivation or motivations of the agent and it can be related to its internal and external perceptions through a function *S*. The first approximation can be done as follows,

$$s(t) = S[e(t), i(t)] = S\,[W\,[e(t\text{-}1), A(t\text{-}1)], I\,[i(t\text{-}1), A(t\text{-}1)]]$$

The main objective of the cognitive architecture is the satisfaction of the motivation of the agent, which, without any loss of generality, may be expressed as the maximization of the satisfaction *s(t)* in each instant of time. i.e.

$$max\{s(t)\} = max\ \{S\,[W\,[e(t\text{-}1), A(t\text{-}1)], I\,[i(t\text{-}1), A(t\text{-}1)]]\}$$

To solve this maximization problem, the only parameter the agent can modify is the *action* it performs, as the external and internal perceptions should not be manipulated. To obtain a system that can be applied in real time, the optimization of the action must be carried out internally, so *W*, *I* and *S* are theoretical functions that must be somehow obtained.

Where,

1. World Model (W) - function that relates the external perception before and after applying an action.
2. Internal Model (I): function that relates the internal perception before and after applying an action.
3. Satisfaction Model (S): function that provides a predicted satisfaction from predicted perceptions provided by the world and internal models.

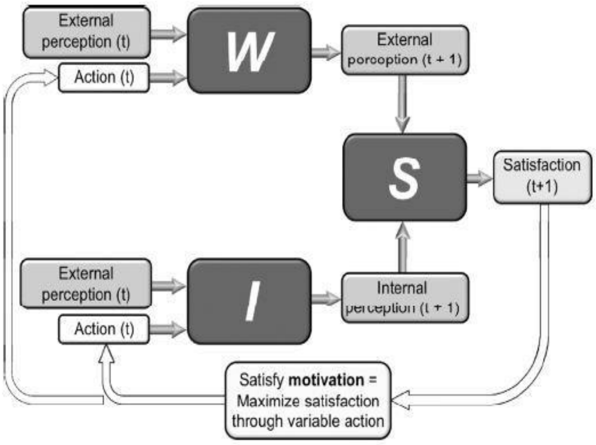
External perception (t)
Action (t)
W
External poroeption (t + 1)
S
Satisfaction (t+1)
External perception (t)
Action (t)
I
Internal perception (t + 1)
Satisfy motivation = Maximize satisfaction through variable action

Figure 1: Function Diagram of the Cognitive Model

The main starting point in the design of a developmental cognitive architecture was that the acquisition of knowledge should be automatic. Thus we establish that the three models *W*, *I* and *S* must be obtained during execution time as the agent interacts with the world.

The information can be extracted from the real data the agent has after each interaction with the environment, this data is called *action-perception pairs* and are made up of the sensorial data in instant t, the action applied in instant t, the sensorial data in instant t+1 and the satisfaction in t+1.

For every interaction of the agent with its environment, two processes must be solved:

1. The modeling of functions W, I and S using the information in the action-perception pairs.
2. The optimization of the action using the models available at that time.

To create models is to try to minimize the difference between the reality that is being modeled and the predictions provided by the model, which involves some type of optimization strategy or algorithm.

III. MULTILEVEL DARWINIST BRAIN (MDB)

The Multilevel Darwinist Brain (MDB) is a Cognitive Architecture that permits an automatic acquisition of knowledge (models) in a real robot through the interaction with its environment, so that it can autonomously adapt its behaviour to achieve its design objectives.

The background idea of the MDB of applying artificial evolution for knowledge acquisition takes inspiration from classical biopsychological theories by Change aux , Conrad, and Edelman in the field of cognitive science relating the brain and its operation through a Darwinist process.

It follows the original cognitive model that has been generalized by basically adding two new aspects:

1. *Behavior structures*: they generalize the concept of action, providing sequences of actions from the sensorial inputs. That is, the robot could have a

behavior for "wall-following," another for "wandering," etc.

2. *Memory elements*: a short-term and a long-term memory are required in the learning processes.

The MDB is structured into two different time scales, one devoted to the execution of the actions in the environment (reactive part) and the other dealing with the learning of the models and behaviors (deliberative part).

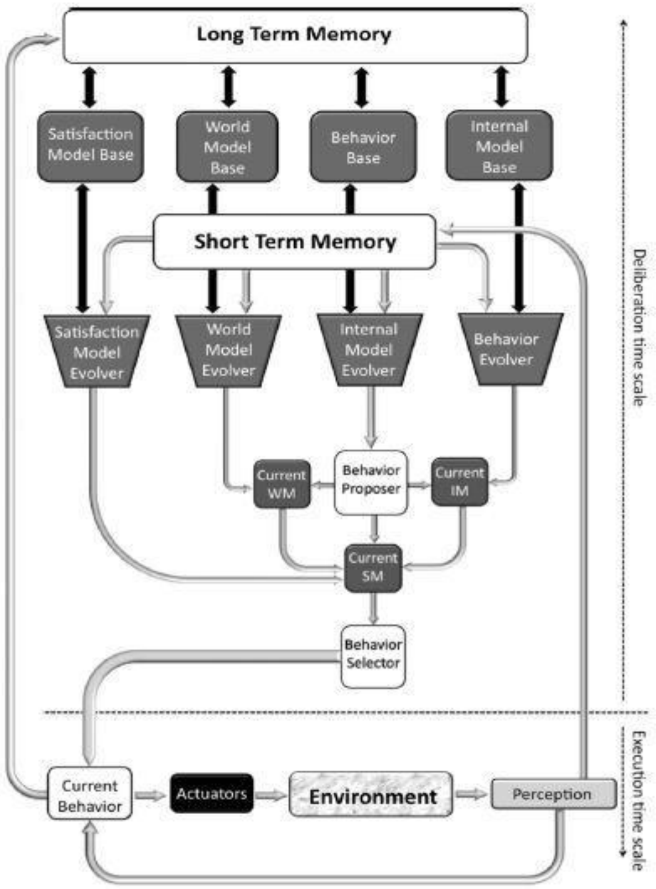

Figure 2: Multilevel Darwinist Brain elements and basic workflow

The operation of the MDB can be described in terms of these two scales:

1. Execution Time Scale : that continuously repeats these steps in a sequential manner:
   1. We can start considering that there is a *current behavior*, which has been selected in the deliberative process, and that provides the action the robot must apply in a given instant of time.
   2. The selected action is applied to the *environment* through the *actuators* of the robot obtaining a new *perception*, that is, new sensing values.
   3. From this perception, the *current behavior* selects the next action to be applied.
2. Deliberation Time Scale: They take place in different time scales and they are not sequential.
   1. The acting and sensing values obtained after the execution of an action in the environment in the execution time scale provide a new action-perception pair that is stored in a *Short-Term Memory* (STM).
   2. The evolutionary model learning processes (for world, internal and satisfaction models) try to find functions that generalize the real samples stored in the STM.
   3. The best models in a given instant of time are taken as *current world, internal and satisfaction models* and are used by the *behavior evolver* to select the best behavior with regards to the predicted satisfaction of the motivation.
   4. The behavior evolver is continuously proposing new behaviors that are better adapted to the STM contents. Upon request, the behavior selector provides the best one to the reactive part of the MDB, the *current behavior* block, which replaces the present one, better adapted to the STM and, consequently, to the current "reality" of the robot.
   5. The *Long-Term Memory* (LTM) block stores those models and behaviors that have provided successful and stable results in their application to a given task in order to be reused directly in other problems or as seeds for new learning processes.

Each time the robot executes an action during real time operation, a new action–perception pair is obtained. This real information is the most relevant one in the MDB, as all the learning processes depend on the number and quality of action–perception pairs.

Each interaction of the robot with the environment has been taken as the basic time unit within the MDB, called *iteration*. As more iterations take place, the MDB acquires more information from the real environment and thus the model learning processes should produce better models and, consequently, the behaviors obtained using these models should be more reliable, and the actions provided by them more appropriate to fulfill the motivations.

(a) Neuroevolution

The main difference of the MDB with respect to other architectures lays in the way the process of modeling functions W, I and S, is carried out by exploring different actions and observing their consequences and using evolutionary techniques. To achieve the desired neural adaptation through evolution established by the Darwinist theories that are the basis for this architecture, Artificial Neural Networks (ANN) is used to for the representation for the models.

The parameters of the ANN will be adjusted using an evolutionary algorithm. The acquisition of knowledge in the MDB is a neuroevolutionary process. Neuroevolution is a reference learning tool due to its robustness and adaptability to dynamic environments and non-stationary tasks. Evolutionary techniques permit the gradual learning process by controlling the number of generations of evolution for a given content of the STM. Thus, if evolutions last just a few generations per iteration, gradual learning by all the individuals is achieved.

To obtain general modelling properties in the MDB, the population of the evolutionary algorithms must be preserved between iterations, leading to a sort of inertia learning effect where, what is being learnt is not the contents of the STM in a given instant of time, but of sets of STMs that were previously seen. This strategy of evolving for a few generations and preserving populations between iterations permits a quick adaptation of models to the dynamics of the environment.

Behaviours can be viewed as neural controllers that provide the action the robot must apply in the environment according to the sensorial inputs.

Due to the peculiarities of the learning processes a new algorithm was developed, a new neuro evolutionary algorithm which is able to deal with general dynamic problems, that is, combining both memory elements and the preservation of diversity. This algorithm was called the Promoter Based Genetic Algorithm (PBGA).

(b) Memories

The management of the Short Term Memory is critical in the real time learning processes in the MDB because the quality of the learned models depends on what is stored in this memory and the way it changes. On the other hand, Long Term Memory is necessary if model knowledge is to be stored.

The STM stores action perception pairs that are taken as the input and target values for the evolution of the world, internal and satisfaction models. The data stored in the STM are acquired in real time as the system interacts with the environment. The STM should basically store the most general and salient information of the environment, and not necessarily the most recent.

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. The LTM stores the knowledge acquired by the agent during its lifetime. 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.

For a LTM to work adequately and, unless it is infinite in size, it is necessary to introduce some type of replacement strategy. This strategy will determine when a model is good enough to go into the LTM and what model in the LTM should be erased to leave room for the new one. The process is not evident, as models are generalizations of situations and it is not easy to see if a model is the same or similar to another one or not. This is the reason for storing the context together with the model as it allows the MDB to test how good models are in predicting other model's contexts and thus provide a measure of similarity among models in terms of their application.

Consequently the information that is 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 will be polluted by information from previous situations and, consequently, the models generated will 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. When a model fails, it can be understood that the context has changed. This will ensure regulation of the models in the STM.

IV. CONCLUSION

MDB cognitive architecture for robots is a developmental approach to provide real robots with autonomous lifelong learning capabilities. The knowledge acquisition is carried out by means of neuro evolutionary processes that use the real data obtained during the operation of the robots. The possibility of a dynamic change of motivations adapted to the robot behavior and environmental conditions is an aspect that should be included in the architecture in order to produce really autonomous and adaptive systems. The current version of the MDB does not take into account the social aspects of autonomous operation. Several experiments with real robots show how promising this approach is, especially once the main practical implementation drawbacks of evolutionary approaches in real time operation.

REFERENCES

[1] A Cognitive Developmental Robotics Architecture for Lifelong Learning by Evolution in Real Robots - F. Bellas, *Member, IEEE*, A. Faiña, G. Varela, R. J. Duro, *Senior Member, IEEE, 2010*

[2] Multilevel Darwinist Brain (MDB): Artificial Evolution in a Cognitive Architecture for Real Robots - Francisco Bellas, *Member, IEEE*, Richard J. Duro, *Senior Member, IEEE*, Andrés Faiña, and Daniel Souto, 2010

[3] A Procedural Long Term Memory for Cognitive Robotics Optimizing Adaptive Learning in Dynamic Environments - R. Salgado, F. Bellas, P. Caamailo, B. Santos-Diez, R. J. Duro,2010

[4] A Novel Cognitive Architecture for Simulated Robots in an Artificial World - Arvin Agah & George A. Bekey, 1994

[5] http://en.wikipedia.org/wiki/Cognitive_architecture