

# Embodying cognitive abilities: categorization

Ricardo A. Téllez and Cecilio Angulo

GREC - Knowledge Engineering Research Group,  
UPC - Universitat Politècnica de Catalunya, Spain  
`r_tellez@ouroboros.org`, `cecilio.angulo@upc.edu`

**Abstract** In previous works we have introduced a distributed neural architecture for the generation of complex behaviors in evolutionary robotics. In this paper we show how this architecture is able to create its own categories about the sensed world of a robot by direct interaction of the body with the environment. The distributed elements of the architecture cooperate to express the categories on an inner world that is easily accessible from the outside. We conclude the paper with an explanation of how the inner world created by the robot can be used to gain some insight into the mind-body problem.

## 1 Introduction

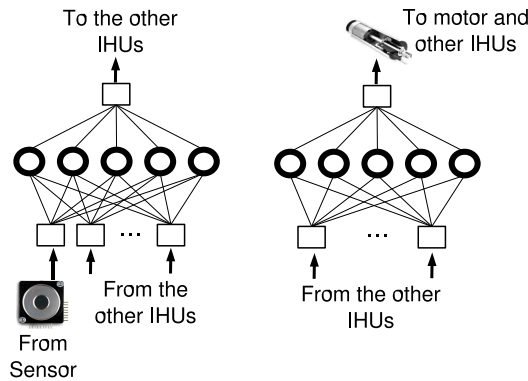
We define categorization as the ability to discriminate between different objects. Categorization is a very important ability of cognitive systems since it allows the cognitive system to determine which elements are good or bad for its survival [6]. When categorizing, an agent immersed in the real world has to be able to make distinctions between different types of objects from the sensed values. Traditional approaches to categorization are based on an information processing approach. However this approach has not solved the problem satisfactorily. This subject has been studied by others researchers like Pfeifer and Scheier [5] and Nolfi [3]. They proposed sensorimotor coordination as the key to categorization, and argued that it was necessary to replace the information processing approach with a sensorimotor approach [8]. This sensorimotor approach has been used in several studies like [1] where sensorimotor couplings gave a meaning to the sensor state through sensory-invariance driven action, or [7] where the external space of the robot was inferred from sensorimotor dependencies. The robot must create its own categories by interacting with its own body on an environment.

In this paper, we will describe a distributed architecture for robot control and how it works for the generation of categories. This architecture will allow us to observe from a different point of view the mind-body problem. Section 2 will briefly describe the architecture. A complete application example will follow in section 3, which shows how the architecture is able to automatically generate the required categories for the resolution of a task. In section 4 we propose a perspective for the mind-body problem, and describe how the architecture can help us gain some insight in this problem using that perspective.

## 2 Brief architecture description

We have created a distributed architecture for the control of autonomous robots, based on neural networks. It is called Distributed Architecture with Internal Representation (DAIR), and a description of its more relevant issues for this paper are included below. The main goal of this architecture is to allow the generation of complex behaviors in complex robots within the evolutionary robotics framework. Because of that, a complete modular distributed architecture was developed.

In the DAIR architecture, each device of the robot is controlled by a small neural network module. For a sensor, the network accepts as input the output of the sensor, and generates as output a processed answer. For an actuator, the network is in charge of producing as output the command for the actuator. In order to allow coordination between the elements, the output of all modules are connected to the inputs of the other elements. Neural networks learn to handle their associated device and to cooperate with other elements by means of an evolutionary algorithm.



**Figure 1.** Connection schematics of the processing element to the associated device (sensor or actuator).

A complete description and comparison of the architecture against other evolutionary robotics architectures can be found in [10]. The application of the architecture to a complex Aibo robot using staged evolution can be found in [9,10].

## 3 Application to the garbage collector problem

In order to test how the architecture described in the previous section leads to category creation, a Khepera robot simulation was used as test bed. Experiments

consisted of the implementation of the DAIR architecture for the control of a Khepera robot while performing a cleaning task called the garbage collector [3]. In this task, a khepera robot is placed inside an arena surrounded by walls and filled with five sticks. The robot should look for any of the sticks randomly distributed on the space, grasp it, and take it out of the arena. For the resolution of this task, the robot must learn to differentiate between sticks and walls, based on the infrared sensor lectures.

### 3.1 Experiment setup

All the experiments reported for the Khepera robot were done on the Webots simulator [2]. In the simulation we used the six front infrared sensors and the gripper sensor. As actuators, the robot has two motors (left and right), but it is also possible to control the position of the gripper arm and the status of the gripper fingers (open or close). The control of the gripper is done by means of two procedures: the first procedure, when activated, moves the arm down, closes the gripper fingers and moves the arm up again, picking a stick up; the second procedure moves the arm down, opens the gripper fingers, and moves the arm up again, releasing the stick.

The physical setup was composed of a rectangular arena of 60x35 cm, surrounded by walls, and containing five garbage cylindric sticks. Each stick had a diameter of 2.3 cm and was positioned randomly inside the arena at every new epoch. In the same way, the robot was also randomly positioned on the arena at the beginning of each epoch.

Experiments consisted of 15 epochs of 200 time steps each, where an evolved controller was tested over the task. The duration of each time step was of 100 ms. Each epoch ended after the 200 steps or after a stick had been correctly released out of the arena.

The DAIR architecture implementation was composed of eleven neural modules: a module for each of the infra-red sensors and the gripper sensor was created, two modules for the left and right motors, and other two for the two gripper procedures. Each module was implemented by a feedforward neural net with eleven inputs, no hidden units, and one output.

A fitness function was created for the evolutionary process which rewarded controllers capable of releasing one stick out of the arena. Controllers that were able to only pick up one stick were also rewarded with a lower fitness.

$$\text{fitness} = \begin{cases} 0.1 & \text{if pick up stick} \\ 1 & \text{if stick released outside arena} \\ 0 & \text{if stick released inside arena} \end{cases} \quad (1)$$

Each controller was tested for 15 epoch per generation, obtaining the final fitness of the controller as the average fitness of all the 15 epochs. Each evolutionary process lasted for 1000 generations. Due to the stochacity of the method employed, the whole evolutionary process was performed ten times.

### 3.2 Results

After 1000 generations, 9 out of the 10 evolutionary runs evolved a maximal fitness behavior (15 sticks released out of 15 epochs), generating a distributed controller able to perform the garbage collector behavior<sup>1</sup>.

### 3.3 Acquisition of categories

In order to solve the garbage collector problem, the architecture must learn to classify and categorize sensor states into walls and sticks. This categorization gives a meaning to the sensory states experienced by the robot. In other words, the meaning is directly grounded in the robot's experiences.

An analysis of the inner workings of the architecture shows how it also makes use of the sensorimotor metaphor for categorization purposes. Furthermore, the categorization made by the architecture can be easily accessed and analyzed from outside the modules.

When observing the outputs of the sensor modules, we noticed that they produce similar output patterns in similar situations. The sensor modules provide the same output values to different sensor values which correspond to the same *conceptual* situation. It can be said that sensor modules are all classifying many different sensor states into the same conceptual category. Given that situation, a *state vector* can be constructed by taking the output values of the sensors modules at each time step. This state vector identifies the situation of the robot at that time step. Thus, the state vector is:

$$\text{state vector} = (O_{S1}, O_{S2}, O_{S3}, O_{S4}, O_{S5}, O_{S6}, O_{S7}) \quad (2)$$

Basically, the state vector can be seen as a categorization of its current situation, or as the construction of an *internal model* of the outside world that the robot is experiencing at that particular moment. This internal representation at the module level contains the meaning of the situation, and that meaning is attributed to the sensor activity pattern and similar ones (the degree of similarity between situations is determined by the distributed coordination mechanism during the evolutionary process). Changes in the values of the sensors do not produce changes in the state vector until they signal that there is a change in the situation of the robot which is relevant to the task to be carried out. Changes between states are not instantaneous, and involve a small transient time where the modules exchange information with each other before finally adopting the new state. All these situations can be easily understood by looking at figure 3, where the components of the state vector are plotted for the case when the robot approaches a wall from a distant place.

The internal representations that map the sensorimotor stimulation to the category actually being experienced are automatically created by the evolutionary process while interacting with the environment. Therefore, the meanings are grounded in the robot's experiences. This means that the actual states identified

---

<sup>1</sup> Video available at [www.ouroboros.org/garbage\\_collector.html](http://www.ouroboros.org/garbage_collector.html)

by the robot have a meaning for the robot. This does not have to correspond to a human meaning but it must be a meaningful state for the robot.

As will be seen below, the robot categorizes only a few, possible, meaningful situations required for the solution of the task at hand, allowing it to reduce the huge number of possible sensorimotor inputs and robot states to a small number of relevant ones.

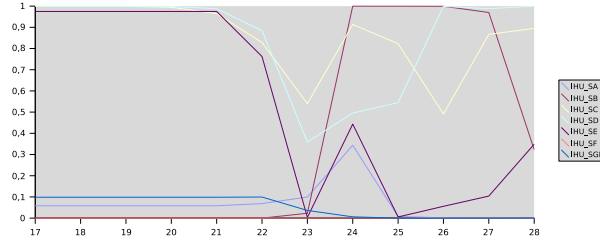
For the garbage collector problem, eight different internal states were identified, each one corresponding to a meaningful situation for the robot. In order to identify the states that the robot evolved, some experiments were performed. Those experiments consisted of allocating the already evolved neural nets to the robot, in a specific situation, and then measuring the values given by the sensor module modules until the situation changed (as a result of the robot's action). Specific situations included placing the robot in free space, and placing it in front of a stick or a wall, at different collision angles and distances. All the situations were tested with and without the robot carrying a stick.

Below is a description of each state identified:

- State a:** This state is observed when the robot is not carrying a stick and does not detect anything. The robot is placed in the middle of the arena, with no obstacles beside it. After an initial transient time, the robot starts moving forward, assuming a stable state where the values of the module outputs do not change at all, which makes the robot advance. This behavior ensures that the robot will eventually detect something, either the wall or a stick.
- State b:** This state is observed when the robot is carrying a stick and does not detect anything. This situation is the same as above, except that, now, the robot has a stick in its gripper.
- State c:** The state observed when the robot detects something but it does not know what it is (a wall or a stick). This state arises when the robot detects something with sensor E but is not capable of classifying what it is. This state produces a special response pattern in the motor modules that makes the robot turn itself round in order to allow other sensors detect the object, and help it to clarify the sensed information.
- State d:** The state observed when the robot is not carrying a stick and is facing a wall. In this case, the robot realizes that there is a wall in front of it, so it starts to move, in order to avoid the wall.
- State e:** This state occurs when the robot is not carrying a stick and is facing a stick. Now the robot detects the stick and recognizes it as such. Therefore, it activates the pick-up procedure in order to pick the stick up.
- State f:** The state observed when the robot is carrying a stick and detects another stick. In this situation, the robot changes its behavior to avoid the stick it has detected.
- State g:** This state is observed when the robot is carrying a stick and detects a wall. In this case, the robot categorizes the obstacle as a wall and then activates the stick-release procedure.

The states observed indicate that tactical modularity does, in fact, use the sensorimotor coordination metaphor in order to create its categorization. The

most clear example of this is the result obtained in state 'c', where the robot detects something but it cannot identify what it is. This situation indicates that the robot is experiencing perceptual aliasing. Its strategy is to move itself into a more suitable position which provides it with a more accurate sensor input that allows it to determine what it is facing.

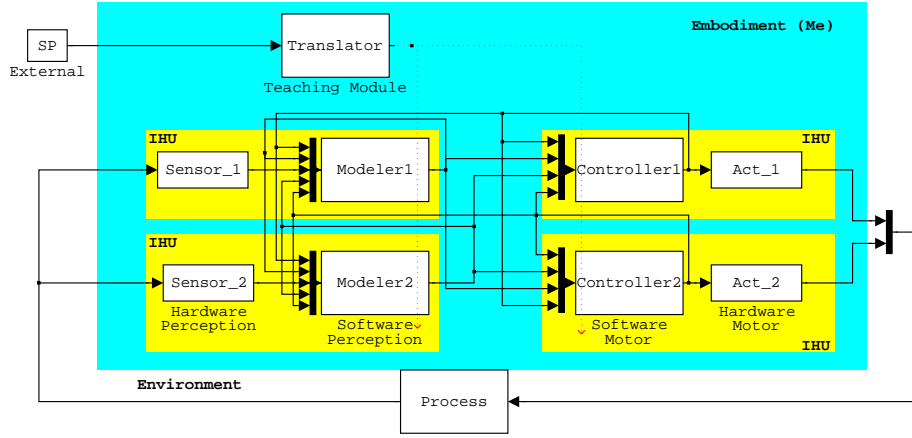


**Figure 2.** Graphic representation of the evolution of the vector state when the robot travels without stick and encounters a wall with a  $45^\circ$  angle on his right. At the beginning, the state of the robot remains unchanged even when the sensor D detects the wall. At time step 22, the sensors E and F detect too the wall what makes the robot identify the obstacle as a wall, changing the vector state to that value and activating an avoidance response (not included here). Transition from the free-space state to the wall-detected state is performed in several time steps which are used for coordination between IHUs (interchange of information). After those steps, the new state is adopted.

#### 4 Using internal representation to approach the mind-body problem

From a control systems point of view we define the *inner world* of a system as the part of the control system corresponding to controller-based units. Similarly, the *outer world* is defined as the part of the control system corresponding to process-based units, i.e., the physical world in which the autonomous agent is situated. From a basic control engineering perspective, so that the whole system reaches the set point (SP), the control elements (inner world) must be designed using a model as close as possible to the outer world, the so-called *process model*. Controller design procedures in control engineering are traditionally model-based, so the performance of the whole system depends on how well the process has been modeled: the internal model of the outer world used to generate the inner world must be as close as possible to the outer world.

If we use those concepts in the design of a robot controller, the architecture can be seen to take control of the robot mind through three elements, which we call *modellers*, *controllers* and *translators*. These elements must adapt the relationship between the robot body and the environment.



**Figure 3.** Schematics of an agent's mind and body using our architecture.

The modeller is a control element that tries to adapt what the sensor perceives from the outer world to what the a deliberative controller may sees in its inner world by also considering the actions of the controller. The translator is a control element that tries to translate the external set point (a behaviour associated to a task) as an interpretation of the outer world. The controller is a control element that deals with both, the internal perception of the 'outer world' (sensor) in the form of the 'inner world' units (modeller), and the task to be accomplished (external set point) translated to a internal set point, also in the 'inner world' units (translator); then appropriated commands are sent to its associated actuator.

Our answer to the mind-body separation problem is shown in figure 3 using the module control concept, i.e., mind is based on the sensors and actuators, which are governed by a learning module that translates external SPs into the driving of fitness functions.

Broadly speaking, Me depends on the goal (goal-directed training) interpreted by the translator, on the environment (outer world) interpreted by the conditioner, and on the body (mind, sensor and actuator) acting through the controller. Hence, Me is based on the mind (translator-conditioner-controller). Information from the environment is *mentally presented*, instead of *mentally represented*: there is no need, as in the traditional approach, to consider any accurate correspondence between the internal model and the real world via a process model. The internal model is built from interaction of the body with the environment; however, in contrast to Parisi [4], it does not try to exactly imitate the world, but is an interpretation of it. The important point is that the

agent's view of the outer world makes sense to the agent himself. Experience and information obtained from the world are therefore highly subjective.

Our modular concept represents a new reactive interpretation of the mind based on internal representations of the real world for its design, i.e., the control elements, to successfully carry out a task. The translator module, converted in a teaching module external to the decentralized reactive mind, contains the fitness functions associated with the tasks that drive learning in the modeler and controller modules. These latter modules that translated from (to) the world signals to (from) the internal representations are usually neural networks. Ideally, deliberative control, which is outside the scope of this study, must take be involved in the translator. Hence, fitness functions related to the task in hand have been assumed to be known to the expert.

## Acknowledgements

This research has been partially supported by the research project EXODUS-ADA (DPI2006-15630-C02-01) of the Spanish Ministry of Education and Science.

## References

1. Y. Choe and S.K. Bhamidipati. Autonomous acquisition of the meaning of sensory states through sensory-invariance driven action. In A. J. Ijspeert, M. Murata, and N. Wakamiya, editors, *Biologically Inspired Approaches to Advanced Information Technology, Lecture Notes in Computer Science*, volume 3141, pages 176–188. Springer, 2004.
2. O. Michel. Webots: Professional mobile robot simulation. *Journal of Advanced Robotics Systems*, 1(1):39–42, 2004.
3. S. Nolfi. Using emergent modularity to develop control systems for mobile robots. *Adaptive Behavior*, 5(3-4):343:364, 1997.
4. D. Parisi. Internal robotics. *Connection science*, 16(4):325–338, 2004.
5. R. Pfeifer and C. Scheier. Sensory-motor coordination: the metaphor and beyond. *Robotics and Autonomous Systems. Special Issue on "Practice and future of autonomous agents"*, 20:157–178, 1997.
6. R. Pfeifer and C. Scheier. *Understanding intelligence*. The MIT Press, 1999.
7. D. Philipona, J. K. O'Regan, and J.-P. Nadal. Is there something out there ? inferring space from sensorimotor dependencies. *Neural Computation*, 15(9), 2003.
8. R. te Boekhorst, M. Lungarella, and R. Pfeifer. Dimensionality reduction through sensory-motor coordination. *Lecture Notes in Computer Science*, 2714:496–503, 2003.
9. R.A. Téllez, C. Angulo, and D. Pardo. Highly modular architecture for the general control of autonomous robots. In *Proceedings of the 8th International Work-Conference on Artificial Neural Networks*, 2005.
10. R.A. Téllez, C. Angulo, and D. Pardo. Evolving the walking behaviour of a 12 dof quadruped using a distributed neural architecture. In *Proceedings of the 2nd International Workshop on Biologically Inspired Approaches to Advanced Information Technology*, 2006.