# Acquisition of meaning through distributed robot control

Ricardo A. Téllez and Cecilio Angulo

***Abstract*— We introduce a distributed neural network based architecture for the control of autonomous robots. This architecture is able to create a meaningful internal representation of the robot current situation directly grounded on its sensorimotor system. The representation is easily accessible from the outside and could be used for further deliberative purposes. An application example is provided for the garbage collector problem, where a robot must learn how to differentiate between garbage and walls, and attach those meanings to different sensor values.**

## I. Introduction

In this paper, we address the problem of attaching meaning to a robot sensor state. The attachment of meaning to robot situations has been mainly done on a manual basis by the designers. In those cases, like for example in experts systems or in voice commands based robots like Aibo, the system operates over the syntax, and the semantic meaning is provided by a human who interprets the system answer, or includes it within the system itself [1]. This approach to semantic handling has been called *conventional functionalism*[2], and is characterized by a complete disentanglement between syntax and semantics. We are more interested, though, on the acquisition and maintenance of meanings by the artificial system itself. Systems equiped with this skill for automatic meaning acquisition are called *natural semantic* systems [3]. A natural semantic system creates and maintains its own meanings from its interactions with the environment.

Natural semantic systems are rare. However, there exists already some examples. For instance, Pierce and Kuipers [4] addressed the problem about a robot learning a model of itself and its environment without initial knowledge of the meanings of the sensors and actuators signals, and how all this knowledge could be used for prediction and navigation. A similar goal was achieved by Philipona et al. [5][6] where a robot was capable of inferring the external space to itself by studying the relations between motor commands and changes in the perception, otherwise called sensorimotor dependencies, by using a set of a priori unknown sensors and actuators. Another example of natural semantic system can be found in [7][8] where sensorimotor couplings were used to acquire the meanings of the robot sensors through sensory-invariance driven action. Finally, in [3], a robot learned to use its a priori unknown effector procedures to achieve its own internal goals.

All those works have in common that meanings are created through a sensorimotor coordination. The use of sensorimotor coordination for meaning acquisition is a real shift from the information processing approach used in most semantics free systems. This change from one approach to the other was proposed by Pfeifer and Scheier in [9][10] and by Nolfi in [11]. In their work, Pfeifer and Scheier view the problem of acquiring meaning as a problem of categorization. Categorization allows an agent immersed in the real world to make distinctions between different types of objects from the sensed values. When using an information processing approach, categorization is only seen as a mapping of sensory stimulation onto a library of stored internal representations. The sensorimotor approach instead, proposes the use of both sensor and effector in a coordinated way to perform the categorization. This approach states that both sensor and motor play an important part in the act of categorizing and by hence, in the acquisition of meaning.

In this paper we present a distributed architecture which allows a robot to automatically acquire the meaning of its sensory inputs, creating an internal representation of it. This representation is like an internal meaningful categorization of the robot situation, created through sensorimotor coordination. Furthermore, this categorization is directly accessible as the output of some modules, hence, it is suitable for its use by other modules. The rest of the paper continues with a description of the architecture employed (in section 2), and follows with an application of the architecture to the garbage collector problem (section 3). The results obtained from the resolution of the garbage problem are used to analyze the inner workings of the architecture and see how meanings are created (section 4). Section 5 discusses the results obtained, and section 6 concludes and points to future work.

## II. 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. The use of such degree of modularity allows the staged evolution of controllers for robots with several sensors and actuator in a process that we call progressive design. A complete description and comparison of the architecture against other evolutionary robotics architectures can be found in [12]. The description of the staged evolution process for progressive design is described in [13]. The application of the

Technical University of Catalonia, Avinguda Víctor Balaguer s/n, 08800 Vilanova i la Geltrú - Barcelona, Spain rtellez@lsi.upc.edu , cecilio.angulo@upc.edu

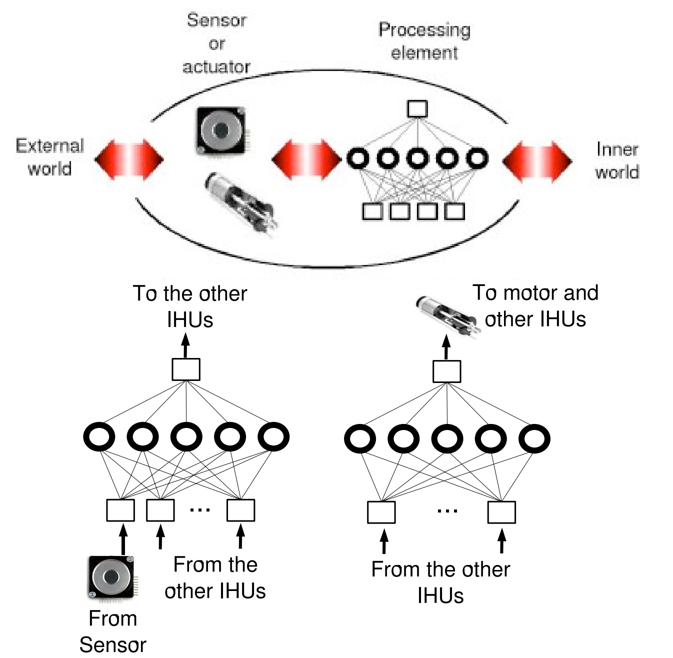


Fig. 1. IHU schematics (above), and connection schematics of the processing element to the associated device (sensor or actuator) (below).

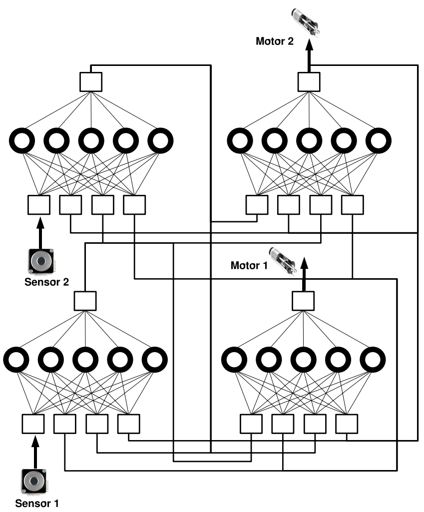


Fig. 2. Application of the DAIR architecture for the control of a simple robot composed of two sensors and two motors. Four IHUs are required.

architecture to a complex Aibo robot using staged evolution can be found in [14], [12].

The DAIR architecture is a distributed modular approach to autonomous robot control. Modularity is implemented by creating a small processing module around each of the robot sensors and actuators. Each module is created by what is called an Intelligent Hardware Unit (IHU) whose schematics is shown in figure 1.

Every IHU is composed of a sensor or an actuator and a processing element which processes the information of its associated device, that is, received sensor information for sensors, and commands sent to the actuator for actuators. It is said that the processing element is in charge of its sensor/actuator. This type of connectivity means that the processing element is the one that decides which commands must be sent to the actuator, or how a value received from a sensor must be interpreted. All IHUs are connected to each other, allowing to each IHU know what the other IHUs are doing. This implies that the processing element is also in charge of deciding what to communicate to the other elements as well as to interpret what the others are communicating.

Hence, the architecture allocates one module for each device. Eventhough each module is independent and perform its own program associated to its device, modules will still have strong couplings between each other. This type of modularity implies that the optimal solution for the control of one device by its IHU will in fact depend on the optimal solutions found by the other IHUs. This type of modularity where great couplings between modules exist has been called decomposable modular system [15], [16].

As a processing element, a neural network was selected. Neural networks are easily evolvable using evolutionary robotics procedures, and present several advantages like inmunity to noise, allow the progresive evolution of its weights, and present a graceful degradation. The type of neural network used will depend of the task to be solved. For instance, in [14] a simple FeedForward neural network with hidden units was used on a standing up behavior. In [12] a Continual Time Recurrent Neural Network was used for the generation of a walking behavior. In this paper a simple FeedForward net with no hidden units was used (see figure 4-bottom). The structure of a IHU can be seen in figure 1, and figure 2 shows how a complete neural controller would be constructed for a simple robotic system composed of two sensors and two actuators. It should be stated that when put several IHU together on a control task, each element has its own particular vision of the situation because each one is in charge of its own sensor or actuator. This means that there is no central coordinator. Each unit knows what the others are doing but needs to select an action for its actuator or sensor output, based on its knowledge of the global situation and the current state of its particular device.

Hence, a distributed coordination between all the elements is required which allows the whole robot perform the behavior required without the use of a central coordinator. In our case, this is accomplished through an evolutionary process using a neuro-evolutionary algorithm. Due to the fact that the evolutionary process has to evolve different ANNs for different roles on a common task, a co-evolutionary algorithm is required, that is, the simultaneous evolution of several nets with a common fitness. By using such kind of algorithm it is possible to teach to the networks how they must cooperate to achieve a common goal (i.e. the global robot behavior to implement), when every network has its own an different vision of the whole system.

The algorithm selected to evolve the nets is the ESP

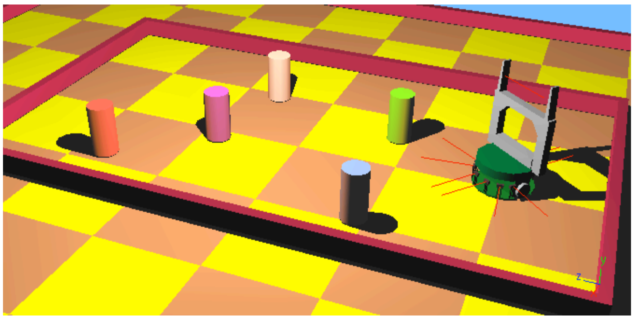

Fig. 3. Simulation of the garbage collector problem on Webots simulator

(Enforced Sub-Populations) [17][18], which has been proved to produce good results on distributed controllers [19]. A chromosome is generated for each IHU network, coding in a direct way the weights of the network connections, and the whole group of neural nets is evolved at the same time with direct interaction with the environment. The fitness function which guides the evolutionary process is created by the designer, depending on the problem that the robot has to solve.

## III. Application to the Garbage Collector Problem

In order to test the theoretical approach presented in the previous section and see how meanings are created, 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. The selected test bed task is called the garbage collector, and follows the description given in [11]. In this task, a khepera robot is placed inside an arena surrounded by walls where it should look for any of the sticks randomly distributed on the space, grasp it, and take it out of the arena (figure 3). The garbage collector behavior requires that the robot completely changes its behavior based on a single sensor value change. When the robot does not carries a stick on the gripper, then its behavior has to avoid walls, look for sticks, approach them, and pick them up. When the robot carries a stick, its behavior has to change to the opposite, avoiding other sticks and approaching walls in order to release the stick out of the arena. This kind of test will allow us to see if the robot creates different classifications for the same object depending on the status of the gripper, or otherwise, the robot has only a single representation for the same object independently of its gripper state, since the object perceived would be the same in both cases.

### A. Experiment setup

All the experiments reported for the Khepera robot were done on a simulator. As simulator, we selected the commercially available Webots simulator by Cyberbotics [20]. This simulator includes, among other things, the simulation of the Khepera gripper, which is the turret capable of grasping objects (see figure 3). The Khepera gripper is composed of an arm that can be moved through any angle from vertical to horizontal, and two gripper fingers that can assume an open or closed position. The gripper is also composed of a sensor that indicates the presence of an object between the fingers.

The robot has eight infrared sensors, six on the front and two on the back. For the resolution of the garbage collector problem only the six front sensors were used, as well as 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 same setup as in Nolfi's work was implemented for the garbage collector task. It is composed of a rectangular arena of 60x35 cm, surrounded by walls, and containing five garbage cylindric sticks. Each stick has 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 creates one IHU element for each device involved. There were eleven devices involved, thus eleven IHUs were created: an IHU for each of the infra-red sensors and the gripper sensor was created (seven in total), two IHUs for the left and right motors, and other two for the two gripper procedures. Each IHU was implemented by a feedforward neural net with eleven inputs, no hidden units, and one output.

The architecture was evolved using the evolutionary setup described above. 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.

$$fitness = \begin{matrix} 0.1 & if\, pick\, up\, stick \\ 1 & if\, stick\, released\, outside\, arena \\ 0 & if\, stick\, released\, inside\, arena \end{matrix}$$

Like in the original experiments made by Nolfi, a special mechanism was implemented which artificially added a stick in front of the robot each time it picked one stick up. The reason was to increase the situations where the robot encountered an obstacle in front of it while carrying a stick. One epoch lasted either 200 steps or until a stick was released outside the arena. 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

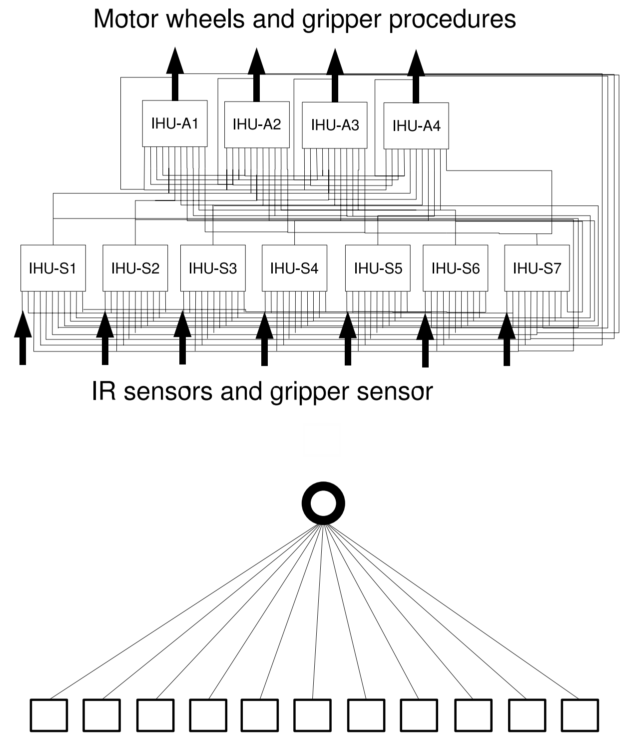


Fig. 4. Modular representation of the architecture implemented for the Khepera robot (top), and the neural network used for each IHU module (bottom)

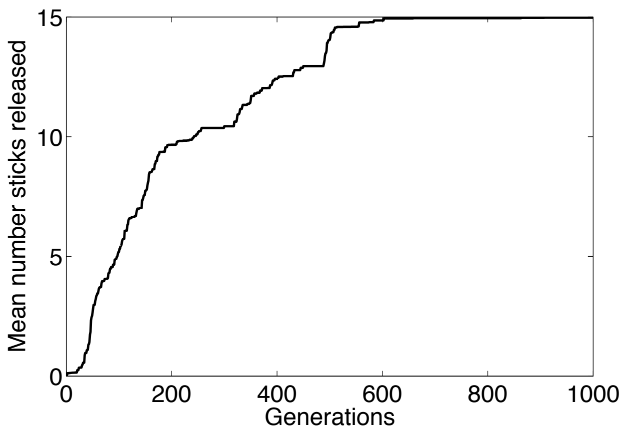


Fig. 5. Mean number of sticks (out of 15) correctly released outside the arena through generations. The curve represents the average in 10 different evolutionary processes.

performed ten times.

### B. 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[1]. Results presented in figure 5 show the evolution of the averaged fitness for those ten runs.

When the controllers obtained are observed on the simulator, we cannot appreciate significant differences in their behavior. All of them perform correctly the behavior of looking for sticks while avoiding walls, pick a stick up, and then release it outside the arena while avoiding other sticks. However, it happens in some special cases that the robot categorizes a stick as a wall, while not carrying a stick. This has not been considered as an error, since this type of wrong classification does not lead to any error in the global behavior. This situation was also observed in Nolfi's original experiments, and could have been avoided in both cases by complexifying the fitness function or by providing to the robot with such strange situations during the evolutionary process as was done with the stick which was put in front of the robot once it picked a stick up.

[1]video of the behavior obtained available at www.ouroboros.org/garbage_collector.html

## IV. ACQUISITION OF MEANINGS

An analysis of the inner workings of architecture shows that the architecture makes use of the sensorimotor metaphor for the apprehension of meaning and its assignment to sensory states. This section will show the meanings generated by the architecture and how can they be accessed from the outside.

When analyzing the outputs of the evolved sensor IHU modules, we observe that they produce similar output patterns to similar situations. The sensor IHUs provided the same output values to different sensor values which corresponded to the same *conceptual* situation. This implies that the sensor IHUs were classifying all a bunch of different sensor states into the same conceptual category or meaning. The different categories or meanings can be accessed by using what we call the *state vector* of the robot at a given time step. The state vector is formed by the concatenation of the output values of the sensors IHUs at each time step, that is:

$$state\ vector = (IHU_{S1}, IHU_{S2}, IHU_{S3}, IHU_{S4}, IHU_{S5}, IHU_{S6}, IHU_{S7})$$

This state vector identifies the situation of the robot at that time step. Basically, it can be seen as a categorization of its current situation, or as an internal modeling of the outside world that the robot is experiencing at this particular moment. This internal representation at the IHU level contains the meaning of the situation, and that meaning is attached to the present sensor activity pattern. Changes in the values of the sensors did not change the state vector, unless a change in the situation of the robot, relevant for the task to solve, was produced. Changes from one state to another one are not instantaneous and involve a transient time where the IHUs exchange information and finally adopt the new state.

The internal representations that map the sensory stimulation to the category actually been experienced are automatically created by the evolutionary process while interacting with the environment. Therefore, the meanings are grounded into the robot experiences. This means that the actual states identified by the robot have a meaning for the robot. However, this meaning does not have to correspond to a human meaning, but a meaningful state for the robot for the task to be solved.

As will be seen below, for the garbage collector problem, the robot identifies only a few possible states as required for the solution of the task at hands, allowing it to reduce the huge number of possible sensor inputs and robot states to that few number of relevant ones. This means that a group of sensors values will always correspond to a unique single meaning or category. This represents a huge reduction from the high number of possible situations that raw sensed data provide. The internal states created by the system identify those states that have a real semantic value, and that value is grounded to the experiences of the robot.

For the garbage collector problem, there have been identified eight different internal states, each one corresponding to a meaningful situation for the robot. In order to identify the states that the robot evolved, some experiments where performed. Those experiments consisted of allocating the robot on a special situation, and then measure the values given by the sensor IHU modules until the situation changed (by means of the robot action). Special situations included putting the robot on free space, and putting the robot in front of a stick or a wall with different collision angles and distances. All situations were tested with and without carrying a stick.

By observing the graphics produced by the IHUs on each of the special situation experiments, we obtained the IHU output values presented in table 1. This table represents the state vectors obtained with one of the 9 controllers evolved. Values presented here were not clear and neat values, but small variations of the order of 0.05 were observed in the same state in different situations. Furthermore, the table represents the values obtained for only one of the 9 distributed controllers obtained. The same concept states were obtained for the other controllers, but their vectorial values were not the same, since the evolutionary process is of stochastic nature, which leads to the evolution of different vector values for the same conceptual states.

It follows a description of the identified states:

State a: This state is obtained when the robot does not carry a stick and does not detect anything. The robot is put in the middle of the arena and no obstacles are put besides it. After an initial transient time, the robot starts moving forward, assuming a stable state where the values of the IHUs outputs do not change at all, making the robot advance forward. This behavior ensures that the robot will eventually detect something, either the wall or a stick.

State b: This state is obtained when the robot carries a stick and does not detect anything. This situation is the same as in the previous state, but now the robot has a stick on its gripper. Basically, the state of the robot is the same as in the previous one, except that the IHU of the gripper sensor indicates that there is a stick on it.

State c: State obtained when the robot detects something but it does not know what it is (a wall or a stick). This state happens when the robot detects something with sensor E but it is not capable of classify what it is. This state, motivates a special response pattern in the motor IHUs that makes the robot turn over itself in order to allow sensors C and D detect the object, and help it to disambiguate the sensing information. Value v1 changes depending on the distance to the object.

State d: State observed when the robot does not carry a stick and it is in front of a wall. In this case, the robot realizes that there is a wall in front of it, so it starts a movement in order to avoid it.

State e: This state occurs when the robot does not carry a stick and it is in front of a stick. Now, the robot detects the stick and recognizes it as that. Therefore, it activates the pick-up procedure in order to pick the stick up. Value v2 changes depending on the distance to the object.

State f: State observed when the robot carries a stick and detects another stick. In this situation, the robot changes its behavior to avoid the detected stick. Strangely, this state is different from the state where the robot did not carry a stick and detected a wall. Value v3 changes depending on the distance to the object.

State g: This state is observed when the robot carries a stick and detects a wall. In this case, the robot categorizes the obstacle as a wall and then activates the releasing stick procedure.

Those observed states indicate that the DAIR architecture uses indeed the sensorimotor coordination metaphor in order to produce its categorization. The most clear example is the result obtained in *state c*, where the robot detects something but it can not identify what it is. This situation indicates that the robot is having perceptual aliasing. Its strategy is to move itself into a more convenient position which provides it with a more convenient sensor input that allows it to determine what it is in front of. This type of behavior is just what has been called as *active perception* [21] or as we have being calling it during this paper, sensorimotor coordination.

## V. Discussion

We have shown how a distributed architecture can create and use meaningful representations for the resolution of the garbage collector task. However, it can be argumented that this representation was also generated on the original experiments by Nolfi, because he was able too to solve the garbage problem. The advantage of the DAIR architecture is that the categorization created is directly accessible to an observer external to the networks, that is, the meanings not internally coded in the network weights. This means that it is possible to direct access the present situation of the robot from a conceptual point of view by just looking the IHU sensor outputs. This type of direct access to the generated meanings may not be necessary in biological intelligent systems, but scientists feel more confortable when such differentiation is possible because allows an easier understanding of the whole process. Furthermore, it may help in the maintenance

| Sensors | IHU A | IHU B | IHU C | IHU D | IHU E | IHU F | IHU Gripper |
|---|---|---|---|---|---|---|---|
| State a | 0.06 | 0 | 1 | 1 | 0.97 | 0 | 0.1 |
| State b | 0 | 0 | 1 | 1 | 0.99 | 0 | 1 |
| State c | 0.06 | 0 | 1 | 1 | v1 | 0 | 0.1 |
| State d | 0.06 | 0 | 0 | 0 | 0 | 0 | 0.01 |
| State e | 0.6 | 0 | v2 | 0 | 0.19 | 0 | 0.01 |
| State f | 0.6 | 0.96 | v3 | 0 | 0 | 0 | 1 |
| State g | 0.01 | 1 | 0.05 | 0.17 | 0 | 0 | 0.97 |

TABLE I

TABLE CONTAINING THE OUTPUT VALUES OF EACH IHU SENSOR FOR THE INTERNAL STATES CREATED. LETTERS A TO F INDICATE EACH OF THE IR SENSORS FROM LEFT TO RIGHT.

of a correspondence between syntax and semantics. This could be achieved, by accessing to the meanings created by a more deliberative superior layer, which would use them to (syntactically) reason about its situation, propagating in this way the robot acquired meanings to more syntactic processes.

From another point of view, we can see the actuation of the architecture as an extractor of meaningful events which are relevant for the resolution of the task. We have seen that the architecture is capable of converting a continuous flow of sensor data into a discrete number of meaningful situations. We will call this situations events. A new event is generated each time that the situation for the robot changes. And the situation changes when the robot itself *thinks* that the new sensory flow corresponds to something really different from previous situation. In fact it creates a categorization of experiences useful for the task at hands. This behavior is similar to the ARAVQ event extractor algorithm [22], with the difference that the ARAVQ extracts the events from the information gathered by a robot that already knows how to solve the task. Instead, the introduced architecture learns to extract the events while learning the resolution of the task.

## VI. CONCLUSION

We have presented a distributed architecture able to control a robot through sensorimotor coordination. The architecture creates its meanings from interaction with the environment, and uses those meanings to classify and solve a garbage collector task. Future work will continue with this bottom-up approach, exploring how to use the state vector generated by the sensor modules to integrate deliberative processes which decide depending of the current situation of the robot, as it is perceived by the robot itself.

## VII. ACKNOWLEDGEMENTS

The authors would like to thank Dr. Olivier Michel for his support in the development of the garbage collector simulation, and Diego Pardo for insightful suggestions. This work has been partially founded by the Spanish Ministry of Education and Science ADA project number DPI2006-15630-C02-01.

## REFERENCES

[1] D. Dennett, “Do-it-yourself understanding,” in *Brainchildren*. MIT Press, 1998. [Online]. Available: http://pp.kpnet.fi/seirioa/cdenn/doityour.htm

[2] N. Block, “What is functionalism?” Published on web from a revised version of the entry on functionalism in The Encyclopedia of Philosophy Supplement, Macmillan, 1996.

[3] P. Cohen and C. Beal, “Natural semantics for a mobile robot,” University of Massachusets, Department of Computer Science, Tech. Rep. 00-59, 2000.

[4] D. Pierce and B. Kuipers, “Map learning with uninterpreted sensors and effectors,” *Artificial Intelligence Journal*, vol. 92, pp. 169–229, 1997.

[5] D. Philipona, J. K. O’Regan, and J.-P. Nadal, “Is there something out there ? inferring space from sensorimotor dependencies,” *Neural Computation*, vol. 15, no. 9, 2003.

[6] D. Philipona, J. K. O’Regan, J.-P. Nadal, and O. J.-M. Coenen, “Perception of the structure of the physical world using unknown sensors and effectors,” *Advances in Neural Information Processing Systems*, vol. 15, 2004.

[7] Y. Choe and S. Bhamidipati, “Autonomous acquisition of the meaning of sensory states through sensory-invariance driven action,” in *Biologically Inspired Approaches to Advanced Information Technology, Lecture Notes in Computer Science*, A. J. Ijspeert, M. Murata, and N. Wakamiya, Eds., vol. 3141. Springer, 2004, pp. 176–188.

[8] Y. Choe and N. Smith, “Motion-based autonomous grounding: Inferring external world properties from internal sensory states alone,” in *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006, pp. 936–941.

[9] 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"*, vol. 20, pp. 157–178, 1997.

[10] ——, *Understanding intelligence*. The MIT Press, 1999.

[11] S. Nolfi, “Using emergent modularity to develop control systems for mobile robots,” *Adaptative Behavior*, vol. 5, no. 3-4, p. 343:364, 1997.

[12] R. 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.

[13] R. Téllez and C. Angulo, “Progressive design of neural controllers through staged evolution,” in *Technical report for the Technical University of Catalonia (U.P.C.)*, 2007.

[14] R. 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.

[15] R. A. Watson, G. S. Hornby, and J. B. Pollack, “Modeling building-block interdependency,” in *Late Breaking Papers at the Genetic Programming 1998 Conference*, J. R. Koza, Ed. University of Wisconsin, Madison, Wisconsin, USA: Stanford University Bookstore, 22-25 July 1998.

[16] R. Watson, “Modular interdependency in complex dynamical systems,” in *Proceedings of the 8th International Conference on the Simulation and Synthesis of Living Systems*, 2002.

[17] F. Gomez and R. Miikkulainen, “Incremental evolution of complex general behavior,” University of Texas, Tech. Rep. AI96-248, 1, 1996.

[18] F. Gómez and R. Miikkulainen, “Solving non-markovian control tasks with neuroevolution,” in *Proceedings of the IJCAI99*, 1999.

[19] H. Yong and R. Miikkulainen, “Cooperative coevolution of multiagent systems,” Department of computer sciences, University of Texas, Tech. Rep. AI01-287, 2001.

[20] O. Michel, “Webots: Professional mobile robot simulation,” *Journal of Advanced Robotics Systems*, vol. 1,

no. 1, pp. 39–42, 2004. [Online]. Available: http://www.ars-journal.com/ars/SubscriberArea/Volume1/39-42.pdf
[21] S. Nolfi and D. Marocco, "Active perception: A sensorimotor account of object categorization," in *Proceedings of the VII International Conference on Simulation of Adaptive Behavior*, 2002.
[22] F. Linåker and L. Niklasson, "Time series segmentation using an adaptive resource allocating vector quantization network based on change detection," in *Proceedings of the International Joint Conference on Neural Networks*, 2000.