"International Journal of Social Robotics, November 2011, vol. 3 (4), 427-441

ISSN: 1875-4791.

DOI: http://dx.doi.org/10.1007/s12369-011-0113-z

This work has been supported by the Spanish Government through the project called "Peer to Peer Robot-Human Interaction" (R2H), of MEC (Ministry of Science and Education), the project "A new approach to social robotics" (AROS), of MICINN (Ministry of Science and Innovation), the CAM Project S2009/DPI-1559/ROBOCITY2030 II, developed by the research team RoboticsLab at the University Carlos III of Madrid.

Learning the Selection of Actions for an Autonomous Social Robot by Reinforcement Learning Based on Motivations

**Álvaro Castro-González · María Malfaz ·
Miguel A. Salichs**

Abstract Autonomy is a prime issue on robotics field and it is closely related to decision making. Last researches on decision making for social robots are focused on biologically inspired mechanisms for taking decisions. Following this approach, we propose a motivational system for decision making, using internal (drives) and external stimuli for learning to choose the right action. Actions are selected from a finite set of skills in order to keep robot's needs within an acceptable range. The robot uses reinforcement learning in order to calculate the suitability of every action in each state. The state of the robot is determined by the dominant motivation and its relation to the objects presents in its environment.

The used reinforcement learning method exploits a new algorithm called Object Q-Learning. The proposed reduction of the state space and the new algorithm considering the collateral effects (relationship between different objects) results in a suitable algorithm to be applied to robots living in real environments.

In this paper, a first implementation of the decision making system and the learning process is implemented on a social robot showing an improvement in robot's performance. The quality of its performance will be determined by observing the evolution of the robot's wellbeing.

Á. Castro-González (✉) · M. Malfaz · M.A. Salichs
RoboticsLab, Carlos III University of Madrid, 28911 Leganés, Spain
e-mail: acgonzal@ing.uc3m.es

M. Malfaz
e-mail: mmalfaz@ing.uc3m.es

M.A. Salichs
e-mail: salichs@ing.uc3m.es

Keywords Motivations · Decision-making · Autonomy · Reinforcement learning · Social robot

1 Introduction

Social Robots are intended for interacting with humans and assisting them in several tasks. *It is desired that such tasks are accomplished by them without surveillance and* this idea implies a certain level of autonomy.

Autonomy is a term widely used in literature and its meaning ranges from very different values. Bekey [9] refers autonomous systems as *systems capable of operating in the real-world environment without any form of external control for extended periods of time*. Cañamero [18] also gives a similar definition of autonomous agents and affirms that they are natural or artificial systems, which must satisfy a set of possible conflictive goals, in constant interaction with dynamic and unpredictable environments with limited resources.

Moreover, Bellman [10] states that autonomy implies a decision making process and this requires some knowledge about the current state of the agent and environment, including its objectives.

We consider the level of autonomy of robots as the amount of decisional mechanisms they are endowed with [23]. In our approach, an autonomous robot must know what action to execute in every situation in order to fulfill its goal. In the case that this robot does not have this knowledge, it must learn this relation between situations and actions. The robot learns this relation by interacting with its own environment where several objects can exist.

According to Mataric [41], learning has been denominated as one of the distinctive marks of the intelligence and introducing adaptation and learning skills in artificial systems is one of the greatest challenges of the artificial intelligence. Moreover, Gadanho [20] states that learning is an important skill for an autonomous agent, since it gives the agent the plasticity needed for being independent.

In this work we propose a motivational system for autonomous decision making which endows robots with the capacity to learn to decide what to do and when to do it to satisfy its inner needs.

Since the robot will be learning on real environment interacting with several objects, it is desired to achieve it on an acceptable range of time. In addition, the relationship among the objects is a key aspect. For example, to drink water, it has to be learnt that, first, you have to grasp a glass, then you fill it with water and eventually you drink it. This process implies different objects.The new reinforcement learning algorithm called Object Q-Learning will deal with all these issues in Sect. 3.

The rest of the paper is organized as follows. Next section presents some previous works about autonomy and decision making in robots. Afterwards, reinforcement learning and its well-known Q-Learning algorithm are introduced. Following, Sect. 4 defines how the state of the robot is formed. Then the Object Q-Learning algorithm is presented. After that, our theoretical approach is explained and then how it has been implemented in a real platform will be shown. Coming after, the results of the experiments with out robot Maggie are presented and they are discussed in the next section. Finally conclusions are summarized and the future works are roughly exposed.

2 Autonomy, Decision Making and Learning in Robots

Autonomy, and more precisely, autonomous robots have been extensively studied in the last years. More general researches have been focused on autonomous agents. As said in [2], agent autonomy is defined like the capability to interact independently, exhibiting control over its own internal state. In addition, Verhagen [60] mentioned that autonomy is the degree in which one agent is independent with respect to another entity, the environment, other agents or even its own developer. Additionally, Barber and Martin [5] consider autonomy as a measure and adjustable value.

Regarding robots, Alami et al. [1] and Estlin et al. [19] present autonomy in robots as the capacity of producing plans for reaching a given goal. These plans could be modified on the fly and goals are given by users. At [24] adaptive autonomy is implemented in a tele-operation robot and the change in autonomy level is made dynamically.

As it has been mentioned before, autonomy in robots is very close to decision making [47]. Different approaches have been used for decision making in robots. Smith [52] implemented fuzzy decision making for navigation purposes in a robot. In contrast, Schermerhorn [49] makes decisions computing goals priorities based on its importance and its urgency. Schermerhorn also remarks the importance of dynamic autonomy in the performance of a robot team. Scheutz [50] presents a decision making using likelihood of success, the benefit and the cost of an action. In [31], decisions are made considering information gathered by humans and robots, and operators are treated as perceptual resources of information; operators are queried depending of the amount of uncertainty in robot's beliefs.

As in many others scientific fields, researches try to imitate humans' mind and last investigations emulate humans' decision making. Accordingly, emotional and motivational models are suggested. As it has been exposed at [8], humans' decision-making is not affected only by the possible outcomes, but also emotions play a main roll. In view of it, several authors propose decision making systems based on motivations, drives, and emotions [17, 21, 37, 42, 59]. In this work, we follow this approach, and the decision making system is based on drives and motivations. Emotions are also included in previous work [39], but in the present implementation they are not included yet.

Our approach has been inspired mainly by Cañamero's [16, 18], Gadanho's [20, 22], and Velasquez's [57, 58] works.

In Cañamero's works, the original idea was that the behaviours of an autonomous agent are directed by motivational states and its basic emotions. The motivations, according to Cañamero, can be viewed as homeostatic processes that maintain a physiological variable controlled within a certain range. When the value of this variable is not equal to its ideal value, the drive emerges. The intensity of the motivation is a function of its related drive and a certain external stimulus. Once the highest motivation is obtained, the intensity of every behaviour, linked to this motivation, is calculated and the one with the highest intensity is executed.

Gadanho proposed a control architecture for autonomous robots with a system based on a set of homeostatic variables which must be maintained within a certain range. In her approach, the robot adapts to its environment using an adaptive controller adjusted using reinforcement learning. The goals are explicitly associated to the homeostatic variables. In order to reflect the hedonic state of the agent, a wellbeing value is created. This value mainly depends on the value of the homeostatic variables. In this approach, this wellbeing value is used as the reinforcement function.

Another approach was developed by Velásquez, who proposed an architecture developed for autonomous agents and contains an emotion generation model and a system of drives in order to develop a decision making model based on emotions. The drives system even exploit its influence in order to select specific behaviours. In this model, the behaviours compete among them to take the control. Therefore, only one behaviour is active at one time.

As will be shown, we use homeostatic drives that are related to motivations, as Cañamero does. Nevertheless, in our approach, the motivations, and not the behaviours (as referred to in Velasquez's approach) compete among them and the winner does not have any related behaviour that satisfies the associated need, as Cañamero proposes. In our situation, the robot must learn, using a reinforcement learning algorithm, which action to select in order to maintain the wellbeing within an acceptable range. This wellbeing is also a function of the needs of the robot, as Gadanho also considers, and its variation will be considered as the reinforcement function for the learning algorithm.

Lorenz defined learning as the adaptive changes of behaviour and this is, in fact, the reason why it exists in animals and humans [33].

In relation to learning in robotics, Mataric in [41] states that learning is particularly difficult in robots. This is because interacting and feeling in the physical world requires to deal with the uncertainty due to the partial and changing information of the conditions of the environment. Nevertheless, learning is an active area in robotics and reinforcement learning is one of the learning methods that has been most successfully implemented in robots. In fact, according to some authors, reinforcement learning seems to be the natural selection for learning policies of mobile robot control. Instead of designing a low-level control policy, a description of the tasks at high-level can be designed through a reinforcement function. Frequently, for robot tasks, rewards corresponds to physical events in the environment. For instance, for the obstacle avoidance task, the robot can obtain a positive reinforcement if it gets its goal, and negative if it crashes into some obstacle [51].

3 Reinforcement Learning and the Q-learning Algorithm

Reinforcement learning allows an agent to learn behaviour through trial and error interactions with a dynamic environment. An agent is connected with its environment via perception and action. On each step of iteration the agent receives information about the state s of the environment. The agent then chooses an action a. The action changes the state of the environment, and then the agent receives a reinforcement signal r. The goal of the agent is to find a policy, mapping to states to actions, that maximizes some long-run measure of reinforcement [30].

Reinforcement learning has been successfully implemented in several virtual agents and robots [4, 12, 29, 40, 43, 55]. One of the main applications, for robots or agents, is the learning of complex behaviours as a sequence of basic behaviours. Those complex behaviours allow to optimize the adaptation of the agent or robot to its environment. The reinforcement learning algorithm named Q-learning [62] has become one *of the methods that is most used* in autonomous robotics [56].

The goal of the Q-learning algorithm is to estimate the Q values for every state-action pair. The Q value is defined as the expected reward for executing action a in state s and then following the optimal policy from there [51]. Every $Q(s,a)$ is updated according to:

$$Q(s,a) = (1-\alpha)\cdot Q(s,a) + \alpha \cdot (r + \gamma V(s')) \tag{1}$$

where:

$$V(s') = \max_{a \in A}(Q(s',a)) \tag{2}$$

is the value of the new state s' and is the best reward the agent can expect from the new state s'. A is the set of actions, a is every action, r is the reinforcement, γ is the discount factor and α is the learning rate.

The learning rate α $(0 < \alpha < 1)$ controls how much weight is given to the reward just experienced, as opposed to the old Q estimate [28]. This parameter gives more or less importance to the learnt Q values than new experiences. A low value of α implies that the agent is more conservative and therefore gives more importance to past experiences. If α is high, near 1, the agent values, to a greater extent, the most recent experience.

Parameter γ $(0 < \gamma < 1)$ is known as the discount factor, and defines how much expected future rewards affect decision now. A high value of this parameter gives more importance to future rewards. A low value, on the contrary, gives much more importance to current reward [28].

As previously said, the final goal of the agent is to learn the optimal policy, the one that maximize the total expected reward. This policy relates, with probability 1, the actions that must be selected in every state. Once the optimal function $Q^*(s,a)$ is obtained it is easy to calculate the optimal policy, $\pi^*(s)$, considering all the possible actions for a certain state and selecting the one with the highest value:

$$\pi^*(s) = \arg\max_{a} Q(s,a) \tag{3}$$

4 The State Space

In this work, it is assumed that the robot lives in an environment where it can *interact* with other objects. The goal of the autonomous robot is to learn what to do in every situation in order to *survive and to maintain* its needs satisfied. In this system, the state of the agent $s \epsilon S$ is the combination of its inner state and its external state:

$$S = S_{inner} \times S_{external} \tag{4}$$

where S_{inner} and $S_{external}$ are the sets of internal and external states of the robot, respectively.

The inner state of the robot is related to its internal needs (for instance: the robot is "hungry") and the external state is its state in relation to all the objects present in the environment:

$$S_{external} = S_{obj_1} \times S_{obj_2} \cdots \tag{5}$$

therefore,

$$S = S_{inner} \times S_{external} = S_{inner} \times S_{obj_1} \times S_{obj_2} \cdots \tag{6}$$

where S_{obj_i} is the set of the states of the robot in relation to the object i.

For every object, the robot could be in n different states, so, the number of states will increase as the number of objects in the environment grows. For example, if for every object there are four different binary variables describing the relation of the robot with it, then, for every object we would have: $2^4 = 16$ states in relation to it. Assuming that there are, for example, 10 objects in the environment, then, according to (5), the number of external states would be 16^{10}. Finally, since the state of the robot is its combination between the inner and the external state (6), the final number of states would be even bigger since we must multiply the number of external states by the number of internal states. Moreover, assuming that the robot can execute a certain amount of actions, or skills, with each object, the number of utility values, $Q(s, a)$, for every state-action pair, could become really high. This great number of Q values to calculate presents problems since it would take a long time for those values to converge. This is because in order to those values converge, every state-action pair must be visited by the agent an infinite number of times [28].

4.1 Reduced State Space

As previously stated, as the number of variables increases linearly, the number of states increases exponentially. This problem is known as the curse of dimensionality [61] . Many authors have proposed several solutions to deal with this problem. One solution would be to use the generalization capabilities of function approximators. Feedforward neural networks are a particular case of such function approximators that can be used in combination with reinforcement learning. Nevertheless, although the neural networks seem to be very efficient in some cases of large scale problems, there is no guarantee of convergence [14].

Other authors propose some methods in order to reduce the state space. According to Sprague and Ballard, this problem can be better described as a set of hierarchical organized goals and subgoals, or a problem that requires the learning agent to address several tasks at once [53]. In [26] and [61] the learning process is accelerated by structuring the environment using factored Markov Decision Processes (FMDPs). The FMDPs are one approach to represent large, structured MDPs compactly, based on the idea that a transition of a variable often depends only on a small number of other variables.

In [32], the authors present a review of other approaches which propose a state abstraction, or state aggregation, in order to deal with large state spaces. Abstraction can be thought of as a process that maps the original description of a problem to a much compact and easier one to work with. In these approaches the states are grouped together if they share, for example, the same probability transition and the reward function [13, 25]. Others consider that states should be aggregated if they have the same optimal action, or similar Q-values [15], etc.

In this work, we propose a different solution to reduce the state space: the states related to the objects are going to be treated as if they were *independent of one another*. This assumption is based on human behaviour, since when we interact with different objects in our daily life, one, for example, takes a glass without considering the rest of objects surround.

As a consequence, the external state is considered as the state of the robot in relation to each object separately. This simplification means that the robot, in each moment, considers that its state in relation, for example, to obj_1 is independent from its state in relation to obj_2, obj_3, etc. so the robot learns what to do with every object by separate. This simplification reduces the number of states that must be considered during the learning process of the robot.

$$S^{red}_{external} = \{S_{obj_1}, S_{obj_2}, S_{obj_3}, \ldots\} \tag{7}$$

where $S^{red}_{external}$ is the set of the reduced external states.

For example, following the example presented in the previous section, for the 10 objects present in the world we would obtain $10 \times 16 = 160$ external states, those ones related to the objects. Therefore, the total number of utility values $Q(s, a)$ would be greatly reduced.

Finally, the total state of the robot in relation to each object i is defined as follows:

$$s \epsilon S_i = S_{inner} \times S_{obj_i} \tag{8}$$

where S_i is the set of the reduced states in relation to the object i.

Using this simplification, the robot learns what to do with every object for every inner state. For example, the robot would learn what to do with the docking station when it needs to recharge, or what to do with the player when it is bored, and so on without considering its relation to the rest of objects.

5 Collateral Effects and Object-Q Learning

The simplification made in order to reduce the state space considers the objects in the environment as if they were independent. This assumption implies that the effects resulting from the execution of an action, in relation to a certain object, do not affect to the state of the robot in relation to the rest of objects. Let us give an example: if the robot decides to move towards the player, this action will not affect to the rest of objects. Nevertheless, if previously, the robot was recharging in the docking station, this action (to move towards the player) related to the object player, has affected to the state in relation to the docking station.

The "collateral effects" are those effects produced by the robot on the rest of the objects when interacting with a certain object. Therefore, in order to take into account these collateral effects, a modification of the Q-learning algorithm is proposed: the Object Q-learning, introduced in previous works [36, 38], where it was successfully implemented on virtual agents. Using this algorithm, the Q values are updated in the following way:

$$Q^{obj_i}(s,a) = (1-\alpha)\cdot Q^{obj_i}(s,a) + \alpha\cdot(r+\gamma\cdot V^{obj_i}(s')) \tag{9}$$

where the super-index obj_i indicates that the learning process is made in relation to the object i. Therefore, $s \epsilon S_i$ is the state of the robot in relation to the object i, A_{obj_i} is the set of the actions related to the object i and s' is the new state in relation to the object i. Parameter r is the reinforcement received, γ is the discount factor and α is the learning rate. Moreover,

$$V^{obj_i}(s') = \max_{a\in A_{obj_i}}(Q^{obj_i}(s',a)) + \sum_{m\neq i}\Delta Q^{obj_m}_{\max} \tag{10}$$

is the value of the object i in the new state considering the possible effects of the executed action with the object i, on the rest of objects. For this reason, the sum of the variations of the values of every other object is added to the value of the object i in the new state, previously defined in (2).

These increments are calculated as follows:

$$\Delta Q^{obj_m}_{\max} = \max_{a\in A_{obj_m}}(Q^{obj_m}(s',a)) - \max_{a\in A_{obj_m}}(Q^{obj_m}(s,a)) \tag{11}$$

Each of these increments measures, for every object, the difference between the best the robot can do in the new state, and the best the robot could do in the previous state. In other words, when the robot executes an action in relation to a certain object, the increment or decrement of the value of the rest of objects is considered.

Considering the example presented at the beginning of this section, if the objects the robot can interact with are limited to a *player* and the *charger*, the current states related to these objects are *far* from the player and *plugged* to the charger. Once the action *move towards* the player is executed, then the new states are *close* to the player and *unplugged* from the charger. Therefore Object Q-Learning is applied as follows.[1] From (9), (10) and (11):

$$\begin{aligned} &Q^{player}(far, move\_towards) \\ &\quad = (1-\alpha)\cdot Q^{player}(far, move\_towards) \\ &\qquad + \alpha\cdot(r+\gamma\cdot V^{player}(close)) \end{aligned}$$

where $V^{player}(close)$ is:

$$\begin{aligned} V^{player}(close) = &\max_{a\in A_{player}}(Q^{player}(close,a)) \\ &+ \sum_{obj_m\neq player}\Delta Q^{obj_m}_{\max} \end{aligned}$$

and a can be any action with the *player*. The collateral effects are:

$$\begin{aligned} &\sum_{obj_m\neq player}\Delta Q^{obj_m}_{\max} = \Delta Q^{charger}_{\max} \\ &\quad = \max_{a\in A_{charger}}(Q^{charger}(unplugged,a)) \\ &\qquad + \max_{a\in A_{charger}}(Q^{charger}(plugged,a)) \end{aligned}$$

where a is any action related to *charger*.

6 The Decision Making System

Authors propose a decision making system for a social robot based on motivations where no specific goals are given in advance; the objective of robot's life is to be fine, in the sense that it has to keep its needs (drives) within an acceptable range, but the way to achieve it is not defined.

In our decision making system, the autonomous robot has certain drives and motivations. The goal is to survive by maintaining all its drives satisfied. For this purpose, the agent must learn to select the right action for every state, in order to maximize its wellbeing. The wellbeing of the robot will be defined, in the next section, as a function of its drives.

First, let us introduce the concepts of drive and motivation. The robot can be parameterized by several variables, which must be at an ideal level. When the value of these variables differs from the ideal one, an error signal occurs:

[1] In order to keep the example simple, the state will be formed just by the external state, and the internal state will not be considered.

the drive. These drives constitute urges to act based on bodily needs related to self-sufficiency and survival [16]. In this approach, the drives are considered as the needs of the agent.

Motivations are those internal factors, rather than external ones, that urge the organism to take action [48]. The motivational states represent tendencies to behave in particular ways as a consequence of internal (drives) and external (incentive stimuli) factors [3]. In other words, the motivational state is a tendency to correct the error, i.e., the drive, through the execution of behaviours.

In order to model the motivations of the agent, we use Lorentz's hydraulic model of motivation as an inspiration [34]. In Lorenz's model, the internal drive strength interacts with the external stimulus strength. If the drive is low, then a strong stimulus is needed to trigger a motivated behaviour. If the drive is high, then a mild stimulus is sufficient [11]. Therefore, the intensities of the motivations are calculated as shown in (12)

$$\begin{aligned} &\text{If } D_i < L_d \text{ then } M_i = 0 \\ &\text{If } D_i \geq L_d \text{ then } M_i = D_i + w_i \end{aligned} \tag{12}$$

where M_i are the motivations, D_i are the related drives, w_i are the related external stimuli, and L_d is called the activation level. Motivations whose drives are below respective activation levels will not compete for being the dominant motivation. The general idea is that we eat when we are hungry and also when we have food in front of us, although we do not really need it.

Once the intensities of the motivations are calculated, the internal state of the robot is determined by the motivation with the highest value. This is the dominant motivation.

As has been previously explained, the state of the robot, defined by (4), is a combination of the inner and external state. The inner state, as has been just explained, is determined by the dominant motivation of the robot, and the external state is its defined by its relation to every object in its environment.

The action selected at each moment will depend on the state of the robot and the potential actions, since the external state restricts the possible actions: for example, we can not eat if we do not have food. It is important to note that the robot does not necessarily know the consequences of its actions, nor the reinforcement that it will receive. The robot just have the knowledge about which actions can be executed with every object.

Pairs formed by the state s and each action a will have an associated value $Q(s,a)$ which represents the suitability of that action on that state. These values will be tuned by interaction between the robot and the environment during the learning process. In particular, the already presented Object Q-learning has been applied in this work. In the next section, the learning process will be explained.

6.1 Learning to Choose

In this implementation, based on the drive reduction theory, which states that the drive reduction is the chief mechanism of reward [27], the reinforcement function will be the variation of the wellbeing of the robot. The robot's wellbeing is a function of its drives and it measures the degree of satisfaction of its internal needs. Mathematically:

$$Wb = Wb_{ideal} - \sum_i \alpha_i \cdot D_i\,, \tag{13}$$

where α_i are the ponder factors that weight the importance of each drive on the wellbeing of the robot and Wb_{ideal} is the ideal value of the wellbeing which corresponds to the value of wellbeing when all drives are satisfied. It is easy to observe that as the values of the needs of the robot (the drives) increase, its wellbeing decreases. Therefore, the reward value for one action correspond to the variation of all the drives during its execution. For example, for action A

$$reward_A = \Delta Wb_A = Wb_{afterA} - Wb_{beforeA} \tag{14}$$

Considering (14), the reward for an action will depend on how fast drives change their values and the duration of the action.

At the beginning of the learning process, all Q-values can be set to the same value. This means that no knowledge is provided in advance, there is no better actions than others for any state. On the other hand, in the same manner animals inherit abilities from their parents, previous knowledge can be assigned, so, they do not have to start from scratch. This can be useful when, for example, we do not want to allow the robot to *die*, i.e. battery is depleted, so the knowledge to survive is initially predefined. However, in this work, the former has been applied. This implies that, at some point, the robot could run out of battery because it does not know yet what to do in that case and the robot will try and fail until it learns.

7 Making it Real in Maggie

In this section, the first implementation of the decision making system and the learning process is presented. Moreover, an explanation of how the decision making system interacts with the other components of the control architecture running in the robot is given.

The intended decision making system will be developed and implemented on the social robot named Maggie [45] which is controlled by the Automatic-Deliberative (AD) architecture [6, 7]. Summarizing, AD is a two levels architecture where communication is accomplished by Events, Short Term Memory and Long Term Memory (Fig. 1). Its essential component is the skill and its located in both levels [44].

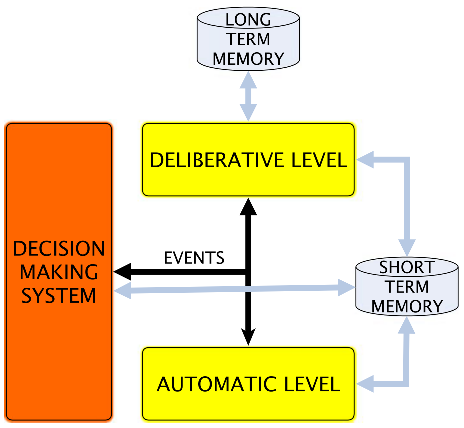


Fig. 1 Components of the AD architecture

An extensive example about how it is applied is presented at [46]. The components of the AD architecture are shown at Fig. 1.

In relation to its hardware, Maggie is built on a wheeled base which allows the robot to move through the environment. Its arms, neck, and eyelids can be moved in a human-like manner. The vision system uses a camera in the head and, thanks to it, Maggie can recognize people and play several games. Laser telemeter and ultrasound sensors are used by the navigation system. By means of an infrared emitter/receiver, Maggie also operates different home appliances such as televisions or music players. Touch sensors on the surface of the body and a touch screen situated in the breast are used for a direct interaction with people. Inside the head, an RFID antenna is placed for identifying objects. In order to provide verbal interaction, our robot is equipped with a text-to-speech module and an automatic speech recognition system.

The proposed decision making system has a bidirectional communication with the rest of the control architecture (Fig. 1). On the one hand, the decision making system selects the proper behaviour which satisfies the most urgent need. This behaviour is translated into a skill (deliberative or automatic one). On the other hand, the decision making system needs information from the environment in order to update the state of the robot and to assess the suitability of the skills activated. This information will be provided by the sensors of the robot.

The aim of the presented decision making system is to achieve a full autonomous robot which learns to make decisions. Once the learning process has finished, the most appropriated action at each moment will be selected by the decision making module. Choosing the right action depends on the value of the motivations, previous experiences, and the relationship with the environment. All these elements have been modelled in order to be processed by the implemented decision making module.

The whole process can be summarized in the next steps:

1. Compute the dominant motivation (internal state)
2. Sense the state in the world (external state)
3. Evaluate possible actions and select one of them according to their Q-values
4. Updating the suitability of the selected action

At Sect. 8 an example of how to apply our motivational decision making system is presented.

7.1 Internal State: Drives and Motivations

As expressed by (12), each motivation is represented by a value and it is affected by two factors: internal needs and external stimuli. Internal needs are the drives, and their values depend on inner parameters. External stimuli are the objects situated in the environment altering the robot motivations. In addition, each motivation has an activation level: under it, motivations values will not be considered for the dominant motivation.

As mentioned, the internal needs, the drives, represent an internal value. Each motivation is connected to a drive. The selected drives are:

calm: the need of peace.
boredom: the need of fun or entertainment.
energy: this drive is necessary for survival.

Since we want Maggie to be an autonomous social robot, based on past works [39] and experiences, three non-conventional motivations have been defined:

relax: it is linked to a peaceful environment and it is related to the drive *calm*.
fun: this motivation is related to entertainment purposes. Its associated drive is *boredom*.
survival: it refers to the energy dependence. This motivation is connected to *energy* need.

Since drives temporally evolve from scratch, motivations do as well. In our implementation, *boredom*, and *calm* drives linearly increase but with different parameters. It means that, as time goes by, these drives become bigger and bigger, and so do the corresponding motivations. *Boredom* is the fastest drive since *calm* evolves slighter. This is because in social robots, as ours, interaction with people is the most relevant aim and fun motivation can involve to stablish relationship with people (for example, dance accompanied by persons). Hence fun motivation takes priority over the others.

The most relevant inner need, due to the necessity of survival, is the *Energy* drive. If we want to achieve a fully autonomous robot, power autonomy is the first step. Therefore, this drive will evolve as battery level varies. Then, its value will match the battery level.

After a drive is satisfied, it does not intermediately start evolving, there is a *satisfaction time* before it evolves again. The same idea occurs to human beings: once they have eaten, they do not feel hungry again but it takes some time before they need to eat again.

Moreover, in order to avoid an unstopped increase in the value of one of the drives, a saturation level is defined for each one: once a drive has reached its saturation value, it will not grow more. Different drives have different saturation values which will affect the dominant motivation in case of a never-ending expansion of the drives. In our implementation, *energy* has the highest saturation level, and the *boredom* and *calm* drives go after.

7.1.1 External Stimuli

Just like human beings can feel thirsty when they see water, the motivations are influenced by objects in the world and their states. That is called the external stimuli for motivations. These stimuli may have more or less influence: their values depend on the states related to the objects.

7.2 External State: Sensing the World

The world is sensed by the robot in terms of objects and the related states to these objects. Objects are not limited to physical objects but abstract objects too. So, in this first implementation, the world where Maggie is living in is limited to the laboratory and three objects have been defined: a cd player, the music in the lab and the docking station for supplying energy. Also relative states to all these items have to be presented and the transitions from one state to another is detected by several skills running on Maggie. More technical issues about how objects are sensed and what a skill is can be found in the references [44, 46].

In Fig. 2, objects and their states, actions, and transitions between states are shown. Dashed arrows represent transitions triggered by actions from other objects.These transitions correspond to the collateral effects presented at Sect. 5. Continuous ones mean actions executed with the objects, and circles are the states related to each item. If an action does not appear at one state, it means that it is incoherent to execute it from that state, e.g., Maggie cannot *play* a cd if it is *far* from the *player* or it cannot *dance* if it is not *listening* music. Section 7.3 details all these actions.

In order to operate the cd player, the robot has to be close to the player. Two different states have been considered: when the robot is close to the player and it is already working (near-on), and when the robot is close but it is not running (near-off). This is required to avoid to send the same command twice to the player. The information required to determine the position related to the player is provided by the navigation system.

The robot's environment is the lab, and *music* can sound there. Then, the robot can be *listening*, or *not*, *music*. Just when the robot is *listening music*, it is able to *dance*. If *music* is mute, it does not make sense to *dance*. The infrared

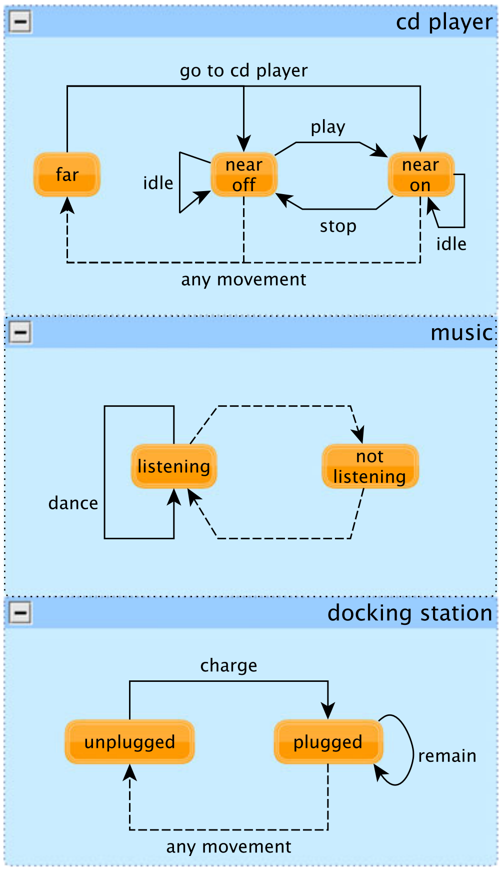


Fig. 2 States and actions for all items: cd player, music, and docking station

emitter is used to play/stop the music when Maggie is close to the player.

The *docking station* is the source of energy. If the robot is *plugged* the battery is charging, otherwise the battery level decreases and the robot is *unplugged*. In order to find the docking station, the robot relies on the navigation system and the information from the laser telemeter. Eventually, to determine if it is plugged or not, a data acquisition device is reading the battery data.

7.3 Acting in the World: What Does Maggie Do Now?

Maggie interacts with the world through the objects and their potential actions. These actions are implemented as skills in the AD architecture. The actions cause effects over the drives. When the actions have ended, i.e. when the skill associated has been blocked because it has reached its goal, the effects are applied. If an error occurs during a skill execution or it is not successful, this situation is notified and its effect is not applied. In our experiments, the effects satisfy a drive, which becomes zero, or they could decrease it. Actions could also "damage" some drives of the robot increasing their values but this has not been considered in this work.

Following, the collection of skills are introduced. The possible actions with *cd player* item are:

- Go to player: Maggie will approach cd player to operate it. If the robot was plugged, this action will unplug the robot.
- Play a cd: a cd is played at the cd player when it is off. This action will produce a change of state in other object: the music, from *non-listening* to *listening*.
- Stop a cd: a cd is stopped when it is being played.This action produces a change of state in other object: the music, from *listening* to *non-listening*.
- Idle: it just represents the possibility to remain next to the player.

About the music, there is just one possible action:

- dance: the robot moves its body with the music. This action is just executed when Maggie is *listening* music.

With the docking station, attainable actions are:

- charge: Maggie plugs to the station and it stays there until the batteries are full. At the end of this action the robot will be *plugged*.
- remain: it holds plugged for a while.

It is important to mention that the transitions between two states and the effects of the actions are unknown by the system, this is, the model of the world is not provided in advance. Therefore, this is a model-free approach. These transitions will be learnt.

Next, once the environment where the robot will live has been presented, how the decision making system operates is explained. After the actions have been presented, the decision making system will select one of them to be executed. First of all, when the system starts, drives begin to evolve independently from their initial value zero, and skills start monitoring the related states to items. When a transition to a new state is detected, an specific event is emitted and the states are written in the short-term memory. The decision making module receives this event and states data are updated. Within robot lifetime, the action selection loop is executed in order to determine the next skill to be activated. At each iteration, the dominant motivation is computed as the maximum motivation whose value (internal needs plus external stimulus) is over its activation level. This parameter has been fixed to 10 for every motivation. Using the dominant motivation, the current states related to objects, and the Q-values associated to each feasible action, the next action will be chosen. This Q-values will the best possible action at each world configuration (the dominant motivation plus the state related to each object).

In some cases, the states and the actions are impossible. For example, for *playing a cd* in the *player* Maggie has to be near to the *player* while other cd is not being played (*near-off* state). It does not make sense if *charge* action is activated when the robot is *unplugged*. At this point, values for these combinations are minimal and they will never be selected for execution.

During our first trials, after all values were fixed, the robot was programmed in such way that it always selected the best actions, it is those actions with the highest associated values. This leads to monotonous behaviours and the robot actions become very predictable. In order to allow *risky* behaviours, we have to face the dilemma of exploration vs. exploitation, several times refereed in the field of reinforcement learning [54]. Level of exploration represents the probabilities of executing actions different than those with the highest values. Using Boltzmann distribution, probabilities for selecting an action in a state is given by (15).

$$P_s(a) = \frac{e^{\frac{Q(s,a)}{T}}}{\sum\limits_{b \epsilon A} e^{\frac{Q(s,b)}{T}}} \tag{15}$$

where $Q(s,a)$ is the given Q-value for action a in state s, and A represents all possible actions in state s; T is the *temperature* and it weights exploration and exploitation. High T gives the same likelihood of selection to all possible actions and the next action will be almost randomly selected; low T enforces actions with high values. As it is presented in [35], T value will be set according to (16).

$$T = \delta * \bar{Q} \tag{16}$$

where $\bar{Q}$ is the mean value of all possible Q-values. According to (16), high δ implies high temperature and, therefore, exploration will dominate: actions will be selected with the same probability. Low δ produces low temperatures and, consequently, exploitation will prevail: actions with high Q-values associated will be likely chosen.

8 Experimental Results

As has been established along the whole paper, our goal is to create a robot which is endowed with enough capabilities to autonomously make the proper decisions at each moment. In order to do that, at the beginning, the robot has to learn the right relation between states and actions. Later it will use this knowledge to keep its needs at low level. The experiments will be split into two phases: exploration phase, which corresponds to the learning period, and exploitation, i.e. when the robot applies the acquired knowledge.

All parameters set during the experiments will shape a specific personality for the robot. Changing these parameters, new personalities will be exhibited by the robot. The performance of different personalities will be studied in the future.

In our implementation, all external stimuli values have been fixed empirically. All external stimuli and their values

Table 1 This table shows external stimuli, objects states linked to them, their value, and the affected motivations

Motivation	Ext. stim.	State related to ext. stim.	Value
fun	music	listening	2
survival	docking station	plugged	2

Table 2 Effects of actions

Action	Object	Drive to satisfy
dance	music	boredom
stop	cd player	calm

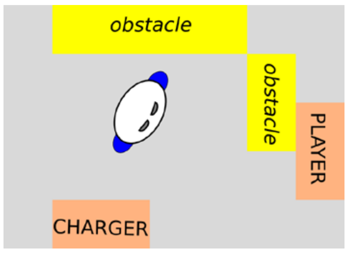


Fig. 3 Experiment set-up

are shown at Table 1. Due to the fact that Maggie really likes to dance, fun motivation is affected when it is listening music. Besides, when the robot *feels* the charger, i.e. when it is plugged, survival motivation receives an increase of two units.

Some actions affect the drives. The effects of these actions are presented in Table 2.

Figure 3 shows the experiment set up in the laboratory. The robot moves in the area between the music player and the docking station and it avoids collisions.

8.1 Exploring

The learning was achieved using the Object Q-Learning explained at Sect. 5. Initially, all Q-values were set to 1, so, there is no previous knowledge. This means that the new state after an action and the reward are unknown. Additionally, since we desire the robot to learn fast according to (9), α will be set to 0.3. This value gives a relevant value to the future rewards so the learning is accelerated. After a while, we started to exploit a bit by gradually decreasing α from 0.3 at iteration 200 to 0.13 at iteration 300. During the rest of the experiment α did not change any more.

Learning must to try every combination for state-action pair. The way to achieve it is setting a high temperature T for the action selection (Sect. 7.3). Therefore, in this phase δ was set to 10.

The next graphs show the evolution of Q-values during a learning session which lasted six hours.

Due to the large extension of the results, just the most relevant results are shown. The rest will be presented on detail on future works.

Figures 4 and 5 plot all the states and the actions when fun and survival are respectively the dominant motivations. Considering that the state of the robot respects to the objects is formed by the state related to each object, the potential set of actions for a particular world configuration will be formed by the possible actions with each object.

Figure 4 presents the learnt values when *fun* is the dominant motivation. It is easy to appreciate that, when the music is *listening*, Fig. 4(f), *dance* action is the most suitable action because it satisfies the *boredom* drive. Moreover, in relation with other objects, if the robot is at state *near-off*, Fig. 4(c), both possible actions, *play* and *idle*, have a relative high Q-value. *Play* is high due to the fact that the robot has learnt that the next best thing the robot can do after *play* is to dance, which has the highest Q-values. *Idle* is a good action because the energy waste is minimal (i.e. the energy drive roughly increases) and the next best action is to *play*, which is high as well. When the robot is far from the player, Fig. 4(e), *go to player* action slightly augments because it has learnt that after this action, it is close to the player and, sometimes, it will be *near-on*, Fig. 4(c), and, others, *near-off* 4(d). The final learnt Q-value represents the average of the values of the two possible states which is a little increase. Actions at *near-on* state, Fig. 4(c), does not have high Q-values because from this state, there is nothing better to do with the player so actions available at this state are penalized. Since the robot needs to have fun, all actions related to the docking station have low Q-values. In fact, when the robot is *unplugged*, Fig. 4(b), and goes to the charger, *charge* action, this not a good action because it implies a waste of energy when the energy is not required. *Remain* plugged, Fig. 4(a), is not either good for the reason that energy drive is not changing but the other drives do. Then the wellbeing is reduced.

All the relations between actions from different objects are collateral effects learnt by the robot.

In Fig. 5, Q-values when survival is the dominant motivation are plotted. In this situation, when the robot is unplugged, Fig. 5(b), *charge* action will have always the highest value because it satisfies the need of energy. Besides, once the robot is plugged, Fig. 5(a), *remain* in that state is not profitable since the batteries are full after the *charge* action. This action is hardly executed because the energy drive has been satisfied and a new dominant motivation arises. However, due to the dynamics of the battery level this is not always true, and this action is few times executed. In other states, actions do not affect energy drive but other drives do. In this case their Q-values increase because there is always

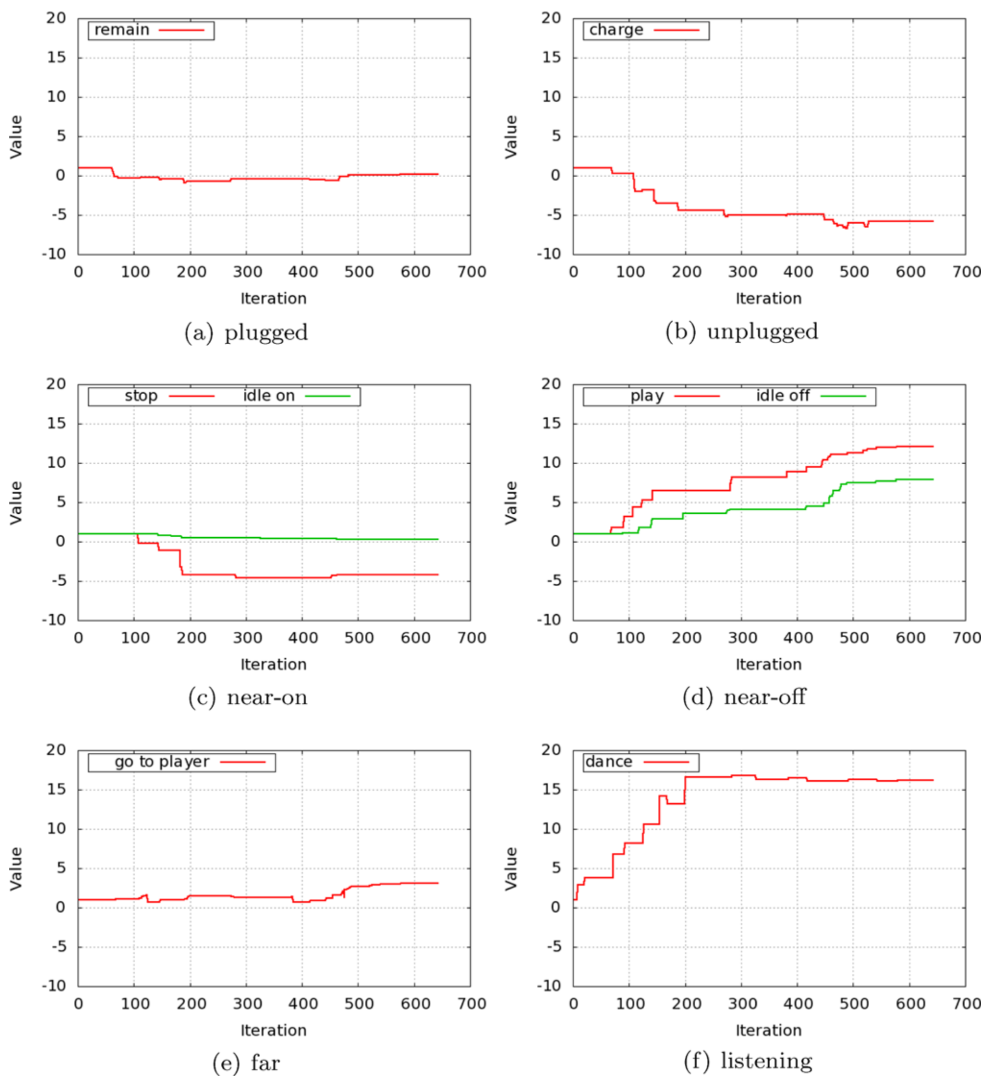


Fig. 4 Q-values when the dominant motivation is fun. Sub-figures plot the potential actions on each state. The states are: plugged (**a**) or unplugged (**b**) to the docking station, near the player when it is working (**c**), or not (**d**), or if it is far from it (**e**), and when the music is listening (**f**)

a reward due to the fact that other drives are satisfied. This happens for example at *listening* with *dance* or at *near-off* with *play*.

8.2 Exploiting

After exploring, the learnt values must be exploited in order to check if the learnt policy is right. Learning is reduced by setting α to 0.0001. This means that Q-values will minimally vary so learning is almost frozen. In addition, robot will tend to select the most suitable actions so low temperature T is needed. However, considering the social aspects of our robot, we do not desire a robot executing always the same sequence of actions which will lead to monotonous behaviors. Then temperature cannot be extremely low. In the experiment $T = 2$.

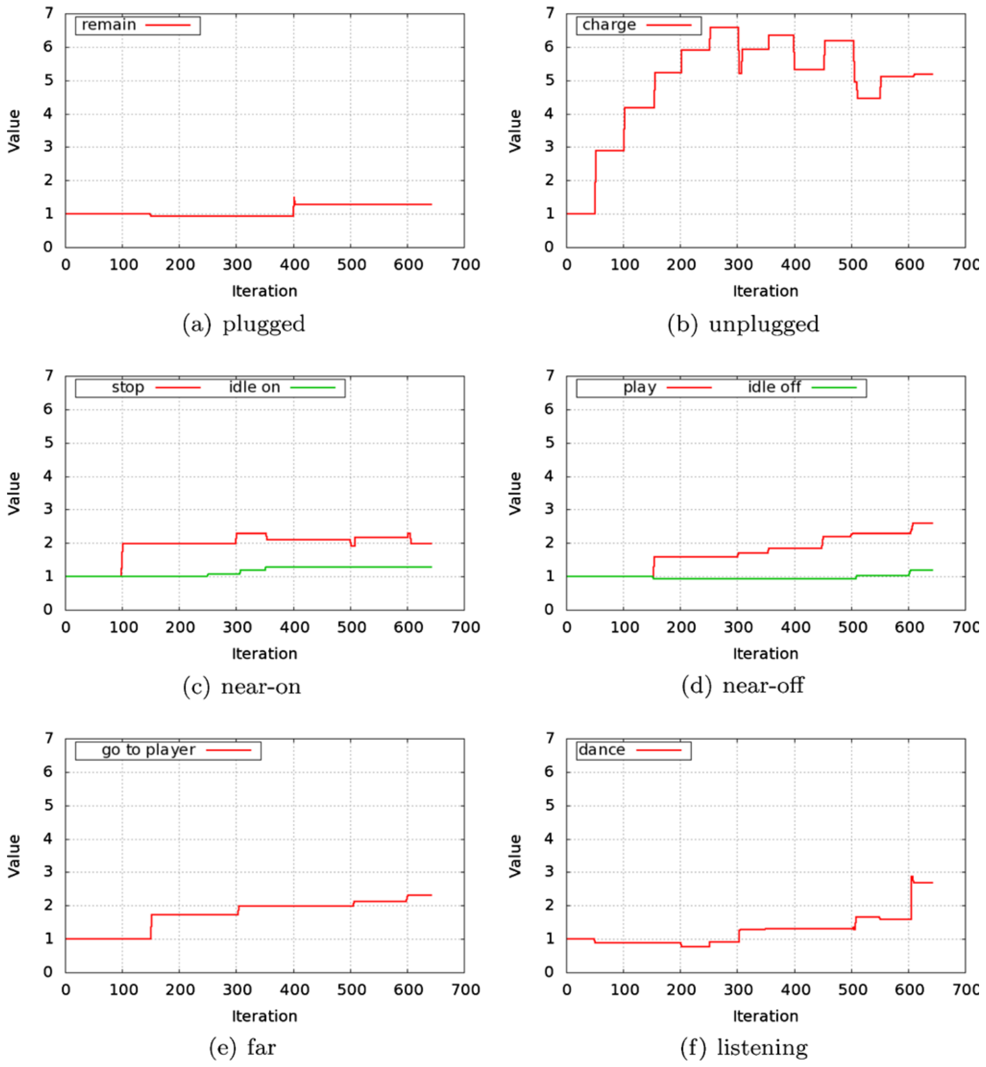
Since wellbeing was employed as reward during the learning process, the robot will tend to maximize the wellbeing. Therefore, the average wellbeing, $\bar{Wb}$, will be used as a performance rate. In addition, a secure area has been defined. When the robot's wellbeing is inside this area, it can be said that the robot is *fine* because the wellbeing is high. The percentage of time in this area will provide an idea about how much time the robot is well. Considering that in the experiments the ideal wellbeing was set to 20, Table 3 presents the results.

Table 3 Exploring vs. Exploiting

Phase	$\bar{Wb}$	Time at secure area
exploring	10.258	55%
exploiting	11.078	69%

We can see that the average wellbeing during the exploiting phase is higher than during the exploring one. This is because when exploiting, the robot is using the learnt policy which will keep the wellbeing at high values. Besides, focusing on the time at the secure area, exploiting provides

Fig. 5 Q-values when the dominant motivation is survival. Sub-figures plot the potential actions on each state. The states are: plugged (**a**) or unplugged (**b**) to the docking station, near the player when it is working (**c**), or not (**d**), or if it is far from it (**e**), and when the music is listening (**f**)



a relevant increase on this value. During the learning phase, 55% of the time the robot was fine. However, explorations rises this value to 69% of the time. This is a significant improvement and we can affirm that the learnt states-actions relationships improve the performance.

9 Discussion

As a primary result, the robot has learnt the right behaviour in order to satisfy a particular drive, e.g. the robot has learnt that it must *dance* in order to satisfy *boredom*, or it has to execute *charge* action to satisfy *energy*. Moreover, additional actions which are required to satisfy the most urgent drive will be selected in the proper order even if they are related to other objects. For example, if the robot is *far* from the *player* and the *music* is off, in order to satisfy the *boredom*, the most likely sequence of actions will be: first, it *goes to the player*, then *play* action is executed (so the *music* is on), and finally it will *dance*. This sequence has been learnt due to the collateral effects because *dance* is related to the *music* but the previous actions correspond to another object, the *player*. This is a clear example about how collateral effects considered in the Object Q-Learning enhance the learning when different objects are involved.

The percentage of time the robot's wellbeing is in the secure area seems to be a reliable measure of the performance. Since during the exploiting phase a little of randomness still exists, the percentage of time at the secure area is lower than if the most suitable action (this is the action with the highest q-value) is always picked. This is a trade-off between the performance (this is the time it stays at the secure area) and non-monotonous robot's behaviours which execute different actions at the same situation.

When the next state after an action (a) is executed from state s is not deterministic, the resulting q-value corresponding to the state-action pair $Q(s,a)$ will average all the received rewards. For example, when the robot is *far* from *player*, the action *go to player* transits to *near-on* or *near-*

off. The final q-value will average all the rewards received from both transitions.

10 Conclusions

In this paper authors present a decision making system implemented on the robot Maggie based on motivations where no predefined goals are provided in advance. This decision making system has learnt from scratch the state-action relationship in order to keep robot's wellbeing in an acceptable range.

Learning has been accomplished by the robot interacting with the environment. The new Object Q-Learning algorithm has been used as the learning algorithm. This algorithm takes advantage of the proposed reduction on the state space: the states related to objects will be considered as independent. Using this simplification the state space is significantly reduced and robot learns to deal with each object. Besides, the possible relationships between objects (collateral effects) are considered by the algorithm. The collateral effects are the effects over the rest of the objects when the robot is interacting with a certain object.

Since the variation of the robot's wellbeing has been used as the reinforcement during the learning, the robot has learnt to keep its wellbeing as high as possible. This is, it learns the best policy to satisfy its needs. In order to achieve it, robot's actions and the robot's state perception,which is formed by its inner state (the dominant motivation) and the external state (the state into its "world"), are accomplished by skills running in the AD architecture.

The experimental tests carried out with the robot showed that the learnt policy keeps the drives low and, therefore, a high wellbeing. During the learning phase, the average robot's wellbeing is lower than when the robot is exploiting the acquired knowledge.

Looking into the results, we can observe that the relationship between actions with different objects has been properly learnt as well. For example, when *fun* is the dominant motivation, the robot has learnt that it has to dance but, if it is not listening music, it has to play it with the player. Moreover, it also has learnt that if it wants to have fun but it is not listening music and it is far from the player, before anything else it must approach to the *player*.

The motivations, the drives, and all other required parameters define the personality for the robot. The value of all parameters are critical to get the desire behaviour and this has to be considered when tuning the parameters.

In summary, the proposed decision making system is able to learn by itself, without prior knowledge, the relationship between the behaviours and the states: it is able to satisfy the most urgent drive selecting the proper behaviour. The presented reinforcement learning process has exhibited great results considering the states of objects independent one from each. Besides, the relationship among several objects has been learnt as well by means of the Object Q-Learning algorithm which considers the collateral effects. All these concepts have been implemented on the social robot Maggie whose wellbeing remains high when Maggie exploits the learnt policy.

11 Future Works

The presented work is limited by the little set of objects the robot interacts with. The capacity of interaction with the environment is limited by the perception and action over the environment. New skills and technologies are being explored in order to interact with more complex objects.

So far, all the objects are static. However, active objects, i.e. objects which are able to act in the environment and change the robot's state, are being studied to incorporate them into the system. The most significant one is the human-being. For example, a person is able to play/stop music and consequently the robot's state will change without its consideration.

In spite of the improvement in the learning algorithm, new algorithms which fit the new requirements will be considered.

In future versions, emotions will take part of the system. We expect that emotions will add more natural behaviors and bring new approaches to the challenging decision making process.

Acknowledgements The authors gratefully acknowledge the funds provided by the Spanish Government through the project called "Peer to Peer Robot-Human Interaction" (R2H), of MEC (Ministry of Science and Education), the project "A new approach to social robotics" (AROS), of MICINN (Ministry of Science and Innovation), and the RoboCity2030-II-CM project (S2009/DPI-1559), funded by Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU.

References

1. Alami R, Chatila R, Fleury S, Ghallab M, Ingrand F (1998) An architecture for autonomy. Int J Robot Res 17(4):315–337. doi:10.1177/027836499801700402
2. Aldewereld H (2007) Autonomy vs. conformity an institutional perspective on norms and protocols. PhD thesis
3. Ávila García O, Cañamero, L (2004) Using hormonal feedback to modulate action selection in a competitive scenario. In: Proc 8th intl conference on simulation of adaptive behavior (SAB'04)
4. Bakker B, Zhumatiy V, Gruener G, Schmidhuber J (2003) A robot that reinforcement-learns to identify and memorize important previous observations. In: The IEEE/RSJ international conference on intelligent robots and systems, IROS2003
5. Barber K, Martin C (1999) Agent autonomy: specification, measurement, and dynamic adjustment. In: Proceedings of the autonomy control software workshop at autonomous agents, vol 1999, pp 8–15
6. Barber R, Salichs MA (2001) Mobile robot navigation based on event maps. In: International conference on field and service robotics, pp 61–66

7. Barber R, Salichs M (2002) A new human based architecture for intelligent autonomous robots. In: Proceedings of the 4th IFAC symposium on intelligent autonomous vehicles. Elsevier, Amsterdam, pp 85–90
8. Bechara A, Damasio H, Damasio AR (2000) Emotion decision making and the orbitofrontal cortex. Cereb Cortex (NY 1991) 10(3):295–307
9. Bekey G (2005) Autonomous robots: from biological inspiration to implementation and control. MIT Press, Cambridge
10. Bellman KL (2003) Emotions in humans and artifacts, chap. Emotions: meaningful mappings between the individual and its world. MIT Press, Cambridge
11. Berridge KC (2004) Motivation concepts in behavioural neuroscience. Physiol Behav 81:179–209
12. Bonarini A, Lazaric A, Restelli M, Vitali P (2006) Self-development framework for reinforcement learning agents. In: The 5th international conference on developmental learning (ICDL)
13. Boutilier C, Dearden R, Goldszmidt M (2000) Stochastic dynamic programming with factored representation. Artif Intell 121(1–2):49–107
14. Boyan J, Moore A (1995) Generalization in reinforcement learning: Safely approximating the value function. In: Advances in neural information processing systems, vol 7. MIT Press, Cambridge, pp 369–376
15. Callum A (1995) Reinforcement learning with selective perception and hidden state. Ph.D. thesis, University of Rochester, Rochester, NY
16. Cañamero L (1997) Modeling motivations and emotions as a basis for intelligent behavior. In: First international symposium on autonomous agents (Agents'97). ACM Press, New York, pp 148–155
17. Cañamero L (2000) Designing emotions for activity selection. Tech. rep., Dept. of Computer Science Technical Report DAIMI PB 545, University of Aarhus, Denmark
18. Cañamero L (2003) In: Emotions in humans and artifacts, chap. Designing emotions for activity selection in autonomous agents. MIT Press, Cambridge
19. Estlin T, Volpe R, Nesnas I, Mutz D, Fisher F, Engelhardt B, Chien S (2001) Decision-making in a robotic architecture for autonomy. In: Proceedings of the international symposium on artificial intelligence, robotics, and automation in space
20. Gadanho S (1999) Reinforcement learning in autonomous robots: an empirical investigation of the role of emotions. PhD thesis, University of Edinburgh
21. Gadanho S (2003) Learning behavior-selection by emotions and cognition in a multi-goal robot task. J Mach Learn Res 4:385–412
22. Gadanho S, Custodio L (2002) Asynchronous learning by emotions and cognition. In: From animals to animats VII, proceedings of the seventh international conference on simulation of adaptive behavior (SAB'02), Edinburgh, UK
23. Gancet J, Lacroix S (2007) Embedding heterogeneous levels of decisional autonomy in multi-robot systems. In: Distributed autonomous robotic systems, vol 6. Springer, Berlin, pp 263–272. doi:10.1007/978-4-431-35873-2
24. Geerinck T, Colon E, Berrabah SA, Cauwerts K, Sahli H (2006) Tele-robot with shared autonomy: distributed navigation development framework. Integr Comput-Aided Eng 13:329–345
25. Givan R, Dean T, Greig M (2003) Equivalence notions and model minimization in Markov decision processes. Artif Intell 147(1–2), 163–223
26. Guestrin C, Koller D, Parr R, Venkataraman S (2003) Efficient solution algorithms for factored mdps. J Artif Intell Res 19:399–468
27. Hull CL (1943) Principles of behavior. Appleton Century Crofts, New York
28. Humphrys M (1997) Action selection methods using reinforcement learning. PhD thesis, Trinity Hall, Cambridge
29. Isbell C, Shelton CR, Kearns M, Singh S, Stone P (2001) A social reinforcement learning agent. In: the fifth international conference on autonomous agents, Montreal, Quebec, Canada
30. Kaelbling LP, Littman LM, Moore AW (1996) Reinforcement learning: a survey. J Artif Intell Res 4:237–285
31. Kaupp T, Makarenko A, Durrant-Whyte H (2010) Human-robot communication for collaborative decision making—a probabilistic approach. Robot Auton Syst 58(5):444–456. doi:10.1016/j.robot.2010.02.003
32. Li L, Walsh T, Littman M (2006) Towards a unified theory of state abstraction for MDP. In: Ninth international symposium on artificial intelligence and mathematics, pp 531–539
33. Lorenz K (1977) Behind the mirror. Methuen Young Books, London. ISBN 04 16942709
34. Lorenz K, Leyhausen P (1973) Motivation of human and animal behaviour; an ethological view, vol XIX. Van Nostrand-Reinhold, New York
35. Malfaz M (2007) Decision making system for autonomous social agents based on emotions and self-learning. PhD thesis, Carlos III University of Madrid (2007)
36. Malfaz M, Salichs M (2009) Learning to deal with objects. In: The 8th international conference on development and learning (ICDL 2009)
37. Malfaz M, Salichs M (2009) The use of emotions in an autonomous agent's decision making process. In: Ninth international conference on epigenetic robotics: modeling cognitive development in robotic systems (EpiRob09), Venice, Italy
38. Malfaz M, Salichs M (2010) Using muds as an experimental platform for testing a decision making system for self-motivated autonomous agents. AISB J 2(1):21–44
39. Malfaz M, Castro-Gonzalez A, Barber R, Salichs M (2011) A biologically inspired architecture for an autonomous and social robot. IEEE Trans Auton Ment Dev 3(2). doi:10.1109/TAMD.2011.2112766
40. Martinson E, Stoytchev A, Arkin R (2002) Robot behavioral selection using q-learning. In: The IEEE/RSJ international conference on intelligent robots and systems (IROS), EPFL, Switzerland
41. Mataric M (1998) Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior. Trends Cogn Sci 2(3):82–87
42. Michaud F, Ferland F, Létourneau D, Legault MA, Lauria M (2010) Toward autonomous, compliant, omnidirectional humanoid robots for natural interaction in real-life settings. Paladyn 1(1):57–65. doi:10.2478/s13230-010-0003-3
43. Ribeiro CHC, Pegoraro R, RealiCosta AH (2002) Experience generalization for concurrent reinforcement learners: the minimax-qs algorithm. In: AAMAS 2002
44. Rivas R, Corrales A, Barber R, Salichs MA (2007) Robot skill abstraction for ad architecture. In: 6th IFAC symposium on intelligent autonomous vehicles
45. Salichs MA, Barber R, Khamis MA, Malfaz M, Gorostiza FJ, Pacheco R, Rivas R, Corrales A, Delgado E (2006) Maggie: a robotic platform for human-robot social interaction. In: IEEE international conference on robotics, automation and mechatronics (RAM), Bangkok, Thailand
46. Salichs J, Castro-Gonzalez A, Salichs MA (2009) Infrared remote control with a social robot. In: FIRA RoboWorld congress 2009, Incheon, Korea. Springer, Berlin
47. Salichs MA, Malfaz M, Gorostiza JF (2010) Toma de decisiones en robotica. Rev Iberoam Autom Inf Ind 7(2):5–16
48. Santa-Cruz J, Tobal JM, Vindel AC, Fernández EG (1989) Introducción a la psicología. Facultad de Psicología. Universidad Complutense de Madrid

49. Schermerhorn P, Scheutz M (2009) Dynamic robot autonomy: investigating the effects of robot decision-making in a human-robot team task. In: Procceding of ICMI-MLMI 2009, Cambridge, MA, USA. doi:10.1145/1647314.1647328
50. Scheutz M, Schermerhorn P (2009) Affective goal and task selection for social robots. Handbook of research on synthetic emotions and sociable robotics: new applications in affective computing and artificial intelligence, p 74
51. Smart WD, Kaelbling LP (2002) Effective reinforcement learning for mobile robots. In: International conference on robotics and automation (ICRA2002)
52. Smith EB (2009) The motion control of a mobile robot using multiobjective decision making. In: ACM-SE 47: proceedings of the 47th annual southeast regional conference. ACM Press, New York, pp 1–6
53. Sprague N, Ballard D (2003) Multiple-goal reinforcement learning with modular sarsa(0). In: The 18th international joint conference on artificial intelligence (IJCAI-03), Acapulco, Mexico
54. Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT Press/Bradford Book, Cambridge
55. Thomaz AL, Breazeal C (2006) Transparency and socially guided machine learning. In: The 5th international conference on developmental learning (ICDL)
56. Touzet C (2003) Q-learning for robots. In: The handbook of brain theory and neural networks. MIT Press, Cambridge, pp 934–937
57. Velásquez J (1997) Modeling emotions and other motivations in synthetic agents. In: Fourteenth national conf artificial intelligence
58. Velásquez J (1998) Modelling emotion-based decision-making. In: 1998 AAAI fall symposium emotional and intelligent: the tangled knot of cognition
59. Velásquez J (1998) When robots weep: emotional memories and decision making. In: Proceedings of AAAI-98
60. Verhagen H (2000) Norm autonomous agents. PhD thesis, The Royal Institute of Technology and Stockholm University
61. Vigorito C, Barto A (2010) Intrinsically motivated hierarchical skill learning in structured environment. IEEE Trans Auton Ment Dev 2(2):132–143. Special Issue on Active Learning and Intrinsically Motivated Exploration in Robots
62. Watkins CJ (1989) Models of delayed reinforcement learning. PhD thesis, Cambridge University, Cambridge, UK

Álvaro Castro-González obtained his degree as Computer Engineer from the University of León, Spain, in 2005. After he became a M.Sc. in Robotics and Automation in 2008 at the University Carlos III of Madrid, Spain, where he is currently a Ph.D. candidate and a member of the RoboticsLab research group. Álvaro has been a teaching assistant for several classes at the Department of Systems Engineering and Automation of the Carlos III University of Madrid. His research interests include personal and social robots, human-robot interaction, decision making systems, emotions and motivations on robots.

María Malfaz received her degree in Physics Science at La Laguna University in 1999. In October 2001, she became a M.Sc. in Control Systems at Imperial College of London and since then she is an Assistant Professor at the Department of System Engineering and Automation of the University Carlos III of Madrid. She received the Ph.D. degree in Industrial Engineering in 2007 and the topic was "Decision Making System for Autonomous Social Agents Based on Emotions and Self-learning". Her current research area follows the line carried out in her thesis.

Miguel A. Salichs is a full professor of the Systems Engineering and Automation Department at the Carlos III University of Madrid. He received the Electrical Engineering and Ph.D. degrees from Polytechnic University of Madrid. His research interests include autonomous social robots, multimodal human-robot interaction, mind models and cognitive architectures. He was member of the Policy Committee of the International Federation of Automatic Control (IFAC), chairman of the Technical Committee on Intelligent Autonomous Vehicles of IFAC, the responsible of the Spanish National Research Program on Industrial Design and Production, the President of the Spanish Society on Automation and Control (CEA) and the Spanish representative at the European Robotics Research Network (EURON).