?Using MUDs as an experimental platform for testing a decision making system for self-motivated autonomous agents Using MUDs as an experimental platform for testing a decision making system

RoboticsLab, Carlos III University of Madrid28911, Leganés, Madrid, Spainmmalfaz@ing.uc3m.esRoboticsLab, Carlos III University of Madrid28911, Leganés, Madrid, Spainsalichs@ing.uc3m.es

Abstract

In this paper a decision making system for autonomous and social agents who live in a virtual world is presented. This world was built using a text based multi-user game: a MUD (Multi User Domain). In this world the agents can interact with one other, allowing social interaction, as well as interaction with the other objects present in the world. In this paper, the usefulness of using this kind of text based multi-user games as test beds for designing decision making systems of artificial agents, is proved.

The proposed decision making system is composed of several subsystems: a motivational system, a drives system and an evaluation and behaviour selection system. The selection of behaviours is learned by the agent using reinforcement learning algorithms. The dominant motivation is considered as the inner state of the agent. In order to simplify the learning process, the states related to the objects are considered as independent from one another. The state of the agent is a combination between his inner state and his state in relation with the rest of agents and objects. This system uses happiness and sadness, defined as positive and negative variations of the wellbeing of the agent, as the reinforcement function.

1 Introduction

The final goal of the work presented in this paper is to design a decision making system for an autonomous and social agent with no *a priori* knowledge. This means that the agent is the one who decides its own actions, and it interacts with other agents. One important feature of the agent is that, using reinforcement learning, it learns the right behaviours to execute through its own experience. This decision making system could be implemented on virtual agents as well as on real robots. In fact, this research was originally oriented to the design of a decision making system for autonomous robots. However, before implementation of this system in a real robot, we used MUDs as an experimental platform for testing this decision making system.

The agent lives in a virtual world where objects, necessary for survival, and other agents exist. This agent must learn a policy of behaviour to survive, maintaining all his needs inside acceptable ranges. The policies establish a normative about what to do in each situation. This means that the agent must learn the proper relation between states and actions. In this system the agent knows the properties of every object, i.e. the agent knows which actions can be executed with each object. What the agent does not know is which action is appropriate in each situation. In order to carry out this learning process, the agent uses reinforcement learning algorithms. In order to create this virtual world a text based game, available on the net and called CoffeMud, gave us the perfect tool to carry out our objective.

Emotions, in general, are used for showing the emotional expression of the agents, as a way of communication among users and for making the agent more believable. Nevertheless, emotions have a fundamental role in human behaviour and social interaction. They also influence cognitive processes, particularly problem solving and decision making [Damasio, 1994]. Emotions can also act as control and learning mechanisms [Fong et al., 2002]. In this work, emotions are used to attempt to imitate their natural function in learning processes and decision making.

The remainder of the paper is organized as follows. In section 2 the concept of autonomy, and its meaning from several points of view, is introduced. This autonomy implies the introduction of new concepts: motivations and drives. Both concepts are explained in this section. Next, in section 3 the decision making system proposed in this work is presented, later the state of the agent is defined, as well as the reinforcement function used in the learning process. Section 4 presents the experimental procedure used in this work. First, the environment, the virtual world where the agents live, is presented and the experimental settings of the agent are described. Finally in this section, the indicators of performance of the agent are introduced. In section 5 and section 6 the experimental results, when the agent lives alone in the world and when he shares the environment with others, are presented and discussed. Finally, the main conclusions of this paper are summarized in section 7.

2 Autonomy

In order for agents to be truly autonomous, not only must they be capable of intelligent action, but they must also be self-sustained [Arkin, 1988]. In other words, autonomy implies a decision making process and this requires some knowledge about the current state of the agent and environment, including his objectives [Bellman, 2003].

According to Cañamero, autonomous agents are natural or artificial systems in constant interaction with dynamic and unpredictable environments, with limited resources. In general, these agents are sociable and they must satisfy a set of possible conflictive goals in order to survive [Cañamero, 2003].

From this same point of view, Gadanho defines an autonomous agent as an agent with goals and motivations. This agent has also some way to evaluate behaviours in terms of environment and his own motivations. The motivations are desires or preferences that can lead to the generation and adoption of objectives. The objectives are situations that must be reached. These final objectives of the autonomous agent, or motivations, must be oriented to maintaining the internal equilibrium of the agent [Gadanho, 1999].

In games, the autonomy of the agents that the user can find while playing is an essential issue for giving them a life-like appearance. For this reason the decision making system presented in this paper is designed for giving the agent the ability to select his own actions. In this work it will be considered that an autonomous agent is the one that is self-motivated, and decides which behaviours to select in order to maintain the internal equilibrium of the agent.

2.1 Homeostasis, drives and motivations

Homeostasis means maintaining a stable internal state [Berridge, 2004]. This internal state can be parameterized by several variables, which must be around an ideal level. When the value of these variables differs from the ideal one, an error signal occurs: drive. These drives constitute urges to action based on bodily needs related to self-sufficiency and survival [Cañamero, 1997].

One of the oldest theories about drives was proposed by Hull in 1943. Hull suggested that privation induces an aversion state in the organism, which was termed drive. According to his theory, drive increases the general excitation level of an animal. Drives were considered as properties of deficit states which motivate behaviour [Hull, 1943].

The word motivation derives from the Latin word *motus* and indicates the dynamic root of behaviour, which means those internal, rather than external factors, that urge the organism to action [Santa-Cruz et al., 1989].

There are several motivational theories that attempt to explain the human and animal behaviour. Nevertheless, there is not a unique classification of those theories. In this section some of those theories will be presented.

2.1.1 Homeostatic theories of motivation

According to these theories, human behaviour is oriented to the maintenance of the internal equilibrium. Among several homeostatic theories, one of them will be selected: The drive reduction theory.

The drive reduction theory

Many drive theories of motivation between 1930 and 1970 posited that drive reduction is the chief mechanism of reward. If motivation is due to drive, then, the reduction of deficit signals should satisfy this drive and essentially could be the goal of the entire motivation [Berridge, 2004].

Hull proposes the idea that motivation is determined by two factors. The first factor is drive. The second one is the incentive, that is the presence of an external stimuli that predicts the future reduction of the need. For example, the presentation of food constitutes an incentive for an hungry animal [Hull, 1943].

2.1.2 Incentive motivation theory

Incentive motivation concepts rose as drive concepts decrease, beginning in the 1960s. Several studies with animals suggested that motivation is more compatible with incentive concepts of taste reward than with earlier drive reduction concepts. It was proposed that individuals were motivated by incentive expectancies, not by drives or drive reduction [Berridge, 2004].

Nevertheless, clearly, a physiological drive state is important for motivation, even if drive is not equivalent to motivation. One does not seek out food when one is thirsty. Physiological deficits such as hunger or thirst depletion signals do modulate motivation for rewards such as food. To incorporate physiological drive/deficit states into incentive motivation, Toates suggested that physiological depletion states could enhance the incentive value of their goal stimuli. This was essentially a multiplicative interaction between physiological deficit and external stimulus, which determined the stimulus' incentive value [Toates, 1986].

3 Decision making system

In this section the decision making system proposed in this work is presented. This system has been developed based on motivation, drive, emotion and learning concepts. These concepts are essential for research in human and animal behaviour.

The objective of this work is to obtain a completely autonomous agent and therefore, an agent with the capacity of making his own decisions. This decision making process has to be learnt through his own experience: his successes and failures. During his experience, using reinforcement learning algorithms, the agent learns the right policy of behaviour in order to survive.

The proposed decision making system is composed of several subsystems: a motivational system, a drives system and an evaluation and behaviour selection system, see Figure 1. The reinforcement function is happiness or sadness that,

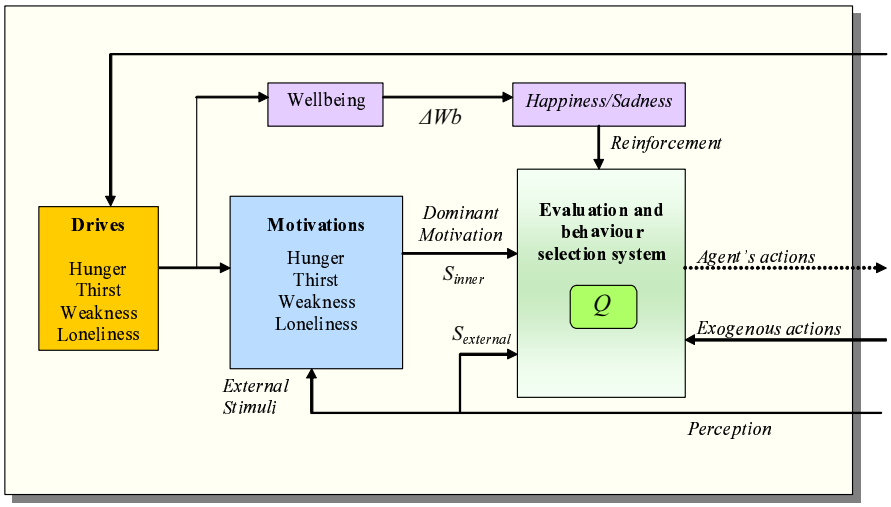

Figure 1: The proposed decision making system

as it will be shown later, are related to the variation of the wellbeing of the agent. How this system works is described next.

When the drives are satisfied their value is zero and as needs increase, the values of the drives increase in each simulation step, with each of them following a certain dynamics. These values are introduced, together with the external stimuli values, in the motivational system. In this system the intensity of the motivations related to each drive are calculated. The motivation with the highest intensity is the dominant motivation. This dominant motivation determines the inner state of the agent. This inner state and the external state define the state of the agent.

The evaluation and behaviour selection system chooses the behaviours according to a certain selection policy. The evaluation of behaviours, when the agent is in a certain state, is done using reinforcement learning algorithms, more specifically Q-leaning. Therefore, the agent learns which action to select in a particular state. The reinforcement used to evaluate the result of the execution of an action are the emotions happiness and sadness. These emotions are defined based on the variation of the wellbeing experimented by the agent: ΔWb. Wellbeing is a function of the needs of the agent. Therefore, this reinforcement measures the effect of the selected action on the needs of the agent. As will be described later in this paper, the positive and negative variations of wellbeing are directly related to happiness and sadness respectively. The agent will use these emotions to evaluate his own actions and to learn which of them are most suitable for each state.

3.1 Wellbeing

The wellbeing of the agent is defined as the degree of needs satisfaction. Therefore, when all the drives of the agent are satisfied, their values are zero and the wellbeing is maximum.

As is shown in equation (1), the wellbeing of the agent is a function of its drives values, D_i, and some personality factors, α_i. These personality factors weigh the importance of each drive in the wellbeing of the agent.

$$Wb = Wb_{ideal} - \sum_i \alpha_i \cdot D_i \tag{1}$$

Wb_{ideal} is the ideal value of the wellbeing of the agent. As the values of the drives of the agent increase as time goes on, or due to the effect of any other action, the wellbeing of the agent decreases. Depending on the values of the personality factors, the increase of the drives can affect, to a certain extent, in the wellbeing of the agent. Every time that a drive reduction exists, there is an increase in the wellbeing.

The wellbeing of the agent is calculated at every simulation step, as well as its variation ($\triangle Wb$). This wellbeing variation is calculated as the current value of the wellbeing minus the value in the previous step, as it is shown in the next equation:

$$\Delta Wb^{k+1} = Wb^{k+1} - Wb^k \tag{2}$$

The biggest positive variation of the wellbeing will be produced when the drive related to the dominant motivation is satisfied.

3.2 Reinforcement learning

The agent that uses reinforcement learning tries to learns, through interaction with the environment, how to behave in order to fulfil a certain goal. The agent and the environment are continuously interacting, the agent selecting actions and the environment responding to those actions and presenting new situations to the agent. The environment and the proper agent also give rise to rewards that the agent tries to maximize over time. This type of learning allows the agent to adapt to the environment through the development of a policy. This policy determines the most suitable action in each state in order to maximize the reinforcement. The goal of the agent is to maximize the total amount of reward he receives over the long run [Sutton and Barto, 1998].

Reinforcement learning has been successfully implemented in several virtual agents and robots [Isbell et al., 2001], [Martinson et al., 2002], [Bakker et al., 2003], [Ribeiro et al., 2002], [Bonarini et al., 2006], [Thomaz and Breazeal, 2006].

3.3 Q-learning

The goal of reinforcement learning is to learn a mapping from states and actions to a measure of the long term value of taking that action in that

state, known as the optimal value function [Smart and Kaelbling, 2002]. The Q-learning optimal value function is defined as:

$$Q^*(s,a) = E\left[R(s,a) + \gamma \max_{a'} Q^*(s',a')\right] \tag{3}$$

This represents the expected value of the rewards received by the agent for taking action a from state s, leading to the new state s', and then acting optimally from there. The parameter γ $(0 < \gamma < 1)$ is known as the discount factor, and is a measure of how much attention the agent pays to possible rewards that the agent might get in the future. In other words, it defines how much expected future rewards affect decisions now [Humphrys, 1997].

The Q-function is frequently stored in a table, indexed by state and action. Starting with arbitrary values, one can iteratively approximate the optimal Q-function based on the observations of the world. Every table entry $Q(s,a)$ is then updated according to [Smart and Kaelbling, 2002]:

$$Q(s,a) = (1-\alpha) \cdot Q(s,a) + \alpha \cdot (r + \gamma V(s')) \tag{4}$$

Where:

$$V(s') = \max_{a \in A} (Q(s',a)) \tag{5}$$

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

In other words, the Q value is the expected rewards for executing action a in state s and then following the optimal policy from there. The goal of the Q-learning algorithm is to estimate the Q values.

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 [Humphrys, 1997].

3.4 Behaviour selection

At the beginning of each experiment the initial values of all Q values are equal to zero. The agent, through his experience in the world, will explore all the possible actions and will update those values. Random exploration takes too long to focus on the best actions, so instead a method will be used that interleaves exploration and exploitation of the best learnt policy.

The specific control policy used is a standard one already implemented, obtaining good results, in [Watkins, 1989]. The agent tries out actions probabilistically based on their Q values using a Boltzmann distribution. Given a state s, it tries out action a with probability:

$$P_s(a) = \frac{e^{\frac{Q(s,a)}{T}}}{\sum_{b \in A} e^{\frac{Q(s,b)}{T}}} \tag{6}$$

Temperature T controls the amount of exploration, i.e. the probability of executing actions other than the one with the highest Q value. If T is high, or if Q values are all the same, this will pick a random action. If T is low and the Q values are different, it will tend to pick the action with the highest Q value.

In order to select the value of T that, as has been shown, will determine the randomness in the action selection, several experiments were carried out. Those experiments showed that this T value is dependent on the Q values. Therefore, in this work it is proposed that, in order to maintain a fixed randomness, T must be defined as a function of the average value of the Q values:

$$T = \delta * average\ value\ of\ Q \tag{7}$$

In the experiments the parameter δ must be tuned in order to determine the exploration/exploitation level. As is shown in (7), a high value of δ will favor the exploration of all the possible actions. On the contrary, a low value of δ will favor the exploitation of the most suitable actions.

3.5 State of the agent

As has been stated for the decision making process, it is necessary to know the state of the agent. In this system the state is the agent is the combination of his inner state, S_{inner}, and his external state, $S_{external}$.

$$S = S_{inner} \times S_{external} \tag{8}$$

Next, both states, inner and external, will be defined.

3.5.1 Inner state

The inner state depends on the motivations that are related to the needs of the agent i.e. the drives. In this system other factors, that may affect the human inner state such as psychological factors, will not be considered.

Motivational states represent tendencies to behave in particular ways as a consequence of internal (drives) and external (incentive stimuli) factors [Ávila García and Cañamero, In other words, the motivational state is a tendency to correct the error, the drive, through the execution of behaviours.

In order to model the motivations of the agent we used the Lorentz's hydraulic model of motivation as an inspiration [Lorenz and Leyhausen, 1973]. Lorenz's hydraulic model is essentially a metaphor that suggests that motivational drive grows internally and operates a bit like pressure from a fluid reservoir which grows until it bursts through an outlet. Motivational stimuli in the external world (food, water, sexual and social stimuli, etc.) act to open an outflow valve, releasing drive to be expressed in behavior. In Lorenz's model, internal drive strength interacts with external stimulus strength. If drive is low, then, a strong stimulus is needed to trigger motivated behaviour. If the drive is high, then a mild stimulus is sufficient [Berridge, 2004].

Have been also introduced activation levels (L_d) for motivations. Therefore the intensity of the motivations, whose related drive is higher than this level is calculated, following the idea of the Lorenz's hydraulic model, as the sum of the intensity of the related drive (D_i) and the related external stimuli (w_i). In other case, the intensity of the related motivation is zero, as is reflected in the following equation:

$$\begin{array}{l} \text{If } D_i < L_d \text{ then } M_i = 0 \\ \text{If } D_i \geq L_d \text{ then } M_i = D_i + w_i \end{array} \tag{9}$$

The external or incentive stimuli are the different objects that the player can find in the virtual world. These incentive stimuli are the same used by Cañamero [Cañamero, 1997]. Therefore, certain behaviours, consummatory ones, can only be executed when these stimuli are present. According to (9), the intensity of a motivation can be high due to two reasons:

1. The value of the correspondent drive is high.

2. The related motivational stimulus is present.

This model can explain the fact that due to the availability of food in front of us, we sometimes eat although we are not hungry.

In this decision making system, as is proposed in [Balkenius, 1993] and [Balkenius, 1995], once all the intensities of the motivations are calculated, these compete one another. The motivation with the highest intensity is the dominant motivation and it is the one that determines the inner state, as shown in equation (10). It can happen that none of the drives of the agent has a value higher than that limit. In that case, there is not any dominant motivation and it can be considered that the agent has no needs, he is "OK".

$$S_{inner} = \begin{cases} \arg\max_i M_i \rightarrow & If\ \max_i M_i \neq 0 \\ OK \rightarrow & In\ other\ case \end{cases} \tag{10}$$

3.5.2 External state

The external state is the state of the agent in relation to all the objects, passive and active, that the agent can interact with:

$$S_{external} = S_{obj_1} \times S_{obj_2} ... \tag{11}$$

Since this definition implies a huge number of states, in this system is considered that the states related to the objects are independent from one another. This means that the agent, in each moment, considers that his state in relation to the food is independent from his state in relation to water, medicine, etc. This simplification reduces the number of states that must be considered during the learning process of the agent.

Without this simplification the number of states in relation to all the objects would be huge. For example, if there were 10 objects present in the world and

it was assumed that for each object there exist 3 logical variables: having the object, being next to the object and knowing where the object is, we would have $2^3 = 8$ states related to every object. If the external state of the agent is his relation to all the objects, $8^{10} = 1.073.741.824$ states will exist, as was previously stated, a huge number of states. Nevertheless, using the simplification, it is considered that the external state is the state of the agent in relation to each object separately, therefore for the 10 objects present in the world we would obtain $10 \times 8 = 80$ states, which is a great reduction in the number of states.

3.6 Modification of Q-learning

The simplification made on the states in relation to the objects causes, for example, the agent to learn, when he is hungry, what to do with the food ($s \epsilon S_{hunger} \times S_{food}$) without considering his relation to the rest of objects. Therefore the total state of the agent in relation to each object is defined as follows:

$$s \epsilon S_{inner} \times S_{obj_i} \tag{12}$$

This definition implies that the value of the actions executed in relation to a certain object are independent of his relation with the rest of objects present in his environment. This is not really true, if for example, the agent is beside the object water and executes the action "go for food", and at the end of this action the agent is next to food. Therefore, the agent is no longer beside the object water, so his state in relation to water has changed although the action executed was related to food.

Therefore, in order to take into account these "collateral effects", a modification of the Q-learning algorithm is proposed:

$$Q^{obj_i}(s,a) = (1-\alpha) \cdot Q^{obj_i}(s,a) + \alpha \cdot \left(r + \gamma \cdot V^{obj_i}(s')\right) \tag{13}$$

Where:

$$V^{obj_i}(s') = \max_{a \in A_{obj_i}} \left(Q^{obj_i}(s',a)\right) + \sum_m \Delta Q^{obj_m}_{\max} \tag{14}$$

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 equation (5).

These increments are calculated as follows:

$$\Delta Q^{obj_m}_{\max} = \max_{a \in A_{obj_m}} \left(Q^{obj_m}(s',a)\right) - \max_{a \in A_{obj_m}} \left(Q^{obj_m}(s,a)\right) \tag{15}$$

Each of these increments measures, for every object, the difference between the best the agent can do in the new state, and the best the agent could do in the previous state.

3.7 Reinforcement function: Happiness and sadness

Considering the definition of emotion given by Ortony [Ortony et al., 1988], it is considered that the emotion occurs due to an appraised reaction (positive or negative) to events. According to this point of view, in [Ortony, 2003], Ortony proposes that happiness occurs because something good happened to the agent. On the contrary, sadness appears when something bad happened. In our system, this can be translated into the fact that happiness and sadness are related to the positive and negative variations of the wellbeing of the agent:

$$\begin{array}{l} \text{if } \Delta Wb > L_h \Rightarrow \text{Happiness} \\ \text{if } \Delta Wb < L_s \Rightarrow \text{Sadness} \end{array} \tag{16}$$

Where ΔWb is the variation of the wellbeing, defined in equation (2), and $L_h \geq 0$ and $L_s \leq 0$ are the minimum variations of the wellbeing of the agent that cause happiness or sadness.

Rolls [Rolls, 2003] proposes that emotions are states elicited by reinforcements (rewards or punishments), so our actions are oriented to obtaining rewards and to avoiding punishments. Following this point of view, in this proposed decision making system, happiness and sadness will be used as the positive and negative reinforcement function respectively, during the learning process.

The use of happiness and sadness, as the reinforcement function in the learning process, is also related to the drive reduction theory. This relationship is based on the definitions of happiness and sadness as the positive and negative variations of wellbeing, respectively. The positive variations, according to (1) and (2), are related to the reduction of drives, while the negative variations are related to their increase.

4 Experimental procedure

As has been already said, the final goal of this work is the design of a decision making system for autonomous and social agents. This system has been tested using a virtual agent who lives in a virtual world where some objects and other agents exist.

4.1 Description of the virtual environment

Bellman in [Bellman, 2003] proposes the use of virtual worlds as test beds for experiments with artificial agents. One of the most important things that is needed is an environment in which one can explore very difficult mappings between goals, agent capabilities, agent behaviours, and interaction with the environment, and consequences or results in that environment. Using virtual worlds this can be reached, but the disadvantages of course, are that these worlds are not nearly as rich as real worlds.

Virtual worlds rose from three mayor lines of development and experience: (1) Role-playing, multi-used Internet games called MUVEs (multi-user virtual

environments); (2) Virtual reality environment and advanced distributed simulation, especially those used in military training exercises; and (3) Distributed computing environments, including Internet.

The use of these virtual worlds as experimental platforms is being extended among the robotic and artificial intelligence community. For example, in [Isbell et al., 2001] the research on reinforcement learning of an artificial agent that lives in a multi-user environment called LambdaMOO, is presented. This environment is one of the oldest text-based multi-user role playing games, and it is formed by interconnected rooms, with users and objects that can move from one room to another. The social interaction mechanisms are designed to reinforce the illusion that the user is present in the virtual space. Another example of the use of computer games as experimental platforms is the work presented in [Thomaz and Breazeal, 2006]. In this work the players interactively train a virtual robot to do a task. As in LabdaMOO, it is an external player who gives the reinforcement to the agent during the learning process.

4.2 The virtual world: Coffeemud

MUD stands for "Multi-User Dungeon", originally developed in 1979, and refers to a text-based multi-user game based on the fantasy adventure genre such as Dungeons and Dragons. The choice of this text-based game instead of using a modern with 3D graphics one arises from the need to simulate a robot with sensors and actuators living in a real world. We wanted a virtual world with a very easy way to send and acquire information. Using this text based game, for our virtual agent, acquiring information is equivalent to reading a text and acting (move, take, etc) is equivalent to sending a text. For our purpose, a MUD offered the perfect way to create a virtual environment containing all the necessary objects (food, water, medicine) to interact with.

Among quite a lot of different MUD codebases, the java-based CoffeeMud [Zimmerman, 2007] has been chosen due to its available documentations and clear explanations.

In a typical MUD, a person would connect to a MUD Server using a Telnet client, and play. Since we want our agents to play, we have created several programs, in C language, that connect to our mud server through the Telnet interface, simulating different players. These agents will behave according to the proposed decision making system.

In order to set up our experiments, we decided to create the area *Passage*. *Passage* was designed in a way similar to the System and Automation Engineering Department plant of the Carlos III University. This means that is formed by a long corridor with rooms situated on both sides since one of the future applications is implementing this decision making system in our robot which will be moving around that scenario.

4.3 Agents at the *Passage* area

This area is formed by 20 rooms, 8 of them forming a corridor and the rest of the rooms are offices distributed at both sides of the corridor. In this area, as has been previously stated, the player can find different objects. These objects can be passive, which are not capable of executing actions, or active, which can execute actions.

The objects that are present in this world are the following:

- Food (passive)
- Water (passive)
- Medicine (passive)
- World (passive)
- Another Agent (active)

Except for the agents, which are moving around autonomously, the rest of the objects are distributed in rooms in such a way that there is a room with food, another with medicine and another with water. The amount of objects present in those rooms are huge, and therefore, it is considered that the agent has unlimited resources. The agent at the beginning of the game does not know where to find those objects. Throughout his time life, the agent finds the objects and remembers their position so if the agent needs some object, he will know where to find it.

There are no doors in this area and the way the agent moves in the world is giving commands of direction: north, south, east and west. With one movement command, the agent passes from one room to another. The commands used for interacting with the passive and active objects will be described later. It is worthy of mention that there are two movement behaviours: "explore" and "go to" which use two kinds of mathematical algorithms. In the case of the "explore" behaviour is the DFS (Depth First Search) algorithm, which gives a route to explore all the rooms of the area. For going to a certain room, the Dijkstra algorithm solves the shortest path problem between the current and the final rooms.

4.4 Graphic interface

Due to the nature of a MUD, as we have stated, the interaction between a player and the game is text based. Although it is quite easy to detect all the objects in the area, it is quite difficult to have a global view of the game since one can only "see" the room where one is placed. Furthermore, when there are several players connected at the same time, the only way to watch the player's evolution is by using a graphic interface developed by the authors for this purpose, see Figure 2.

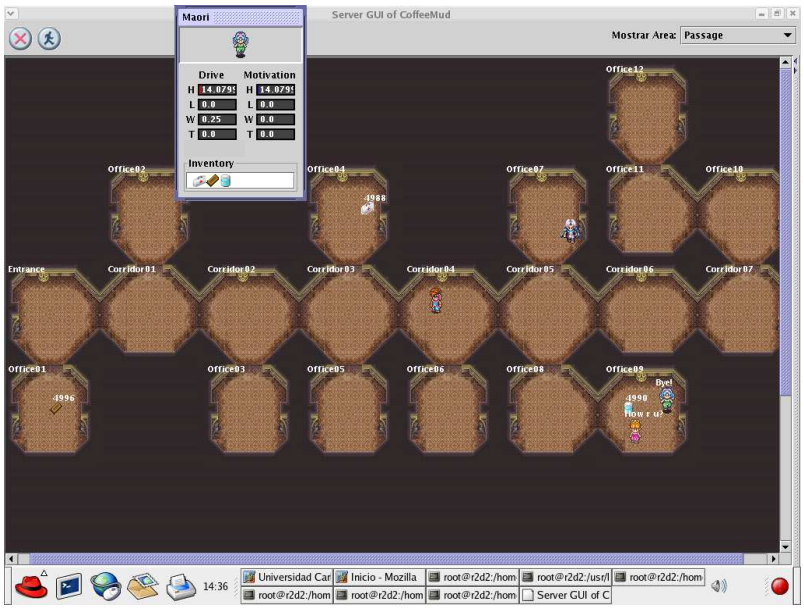

Figure 2: The graphic interface

Figure 3: The agent's sub-window

Using this interface all the player's actions can be followed, as well as their drives and motivations values. Moreover, all the items that the player is carrying are showed on the little sub-windows developed for each player as can be observed in Figure 3.

4.5 Agent's description

4.5.1 *Drives*

The considered drives and motivations are the following:

- Hunger
- Thirst
- Weakness
- Loneliness

These drives have been selected taking into account the needs the agents in the virtual world. A typical player who lives in CoffeMud needs to eat and drink in order to survive. The Weakness and Loneliness drives have been added to make the working frame more complete. Since our final goal is to develop a decision making system for an autonomous and social agent, the need for social interaction is included as one of the agent's needs.

The values of the Hunger and Thirst drives increment a certain amount at every step simulation. These drives do not grow at the same rate. Physiological studies determine that in most human beings the necessity of water, thirst, appears before the need for food, hunger. In [Gautier and Boeree, 2005] it is presented how Maslow discovered that certain needs prevail over others. For example, if one is hungry and thirsty, one will tend to relieve thirst before hunger. After all, one can survive several days without food, but one can only live a couple of days without water. As a conclusion, thirst is a stronger need that hunger.

Some drives, or needs of the agent, after being satisfied do not start to increase their values immediately, but after a certain time, which we term "satisfaction time". This happens in the same way that when after eating, one is not hungry again until some hours later.

In this system, some of the drives of the agent follow this pattern, previously described. Thus, some drives have these satisfaction times whose values depend on the urgency of each of them. These drives are Hunger, Thirst and Loneliness. The Weakness drive follows a different pattern, as will be described later. In the next equation, the satisfaction times corresponding to these drives are shown:

$$\begin{aligned} T_{thirst} &= 50\ steps \\ T_{hunger} &= 100\ steps \\ T_{loneliness} &= 150\ steps \end{aligned} \tag{17}$$

According to these values, the Thirst drive is the most urgent one, since it takes less time to increase its value again. In general, one is thirsty more frequently than one is hungry. The social need, the Loneliness drive, takes much more time in increasing its value again since it is not a very urgent need. Once the satisfaction time passes the drives grow as follows:

$$
\begin{aligned}
D_{thirst}^{k+1} &= D_{thirst}^{k} + 0.1 \\
D_{hunger}^{k+1} &= D_{hunger}^{k} + 0.08 \\
D_{loneliness}^{k+1} &= D_{loneliness}^{k} + 0.06
\end{aligned}
\tag{18}
$$

As is shown, the growing rate of the Thirst drive is higher than that of the Hunger drive, and this drive in turn increases its value faster than the Loneliness drive.

The variation of the Weakness drive depends on the movement of the agent. Therefore, if the agent is still this drive does not suffer any variation, but if the agent moves the value of the drive increases at every step, as is shown next:

$$
D_{weakness}^{k+1} = D_{weakness}^{k} + 0.05 \tag{19}
$$

Moreover, while the agent is interacting with another agent some drives can be affected by the actions executed by the other agent. In fact, when the agent is robbed:

$$
D_{loneliness}^{k+1} = D_{loneliness}^{k} + 1 \tag{20}
$$

and, when the agent is kicked:

$$
\begin{aligned}
D_{loneliness}^{k+1} &= D_{loneliness}^{k} + 1 \\
D_{weakness}^{k+1} &= D_{weakness}^{k} + 3
\end{aligned}
\tag{21}
$$

4.5.2 Motivations of the agent

According to the equation (9), motivations are defined as the sum of the value of the drives and the external stimuli. These external or motivational stimuli, w_i, are the different objects that the agent can find in the world during the game, so:

$$
\begin{aligned}
&\text{If the stimuli is present then } w_i \neq 0 \\
&\text{If the stimuli is not present then } w_i = 0
\end{aligned}
\tag{22}
$$

Table 1 shows the motivations, drives and their related motivational stimuli.

Equation (9) shows the application of the activation levels L_d in order to calculate the value of the intensity of motivations. In the experiments:

$$
L_d = 2 \tag{23}
$$

Table 1: Motivations, Drives and motivational stimuli

Motivation/Drive	Motivational stimuli
Hunger	Food
Thirst	Water
Weakness	Medicine
Loneliness	Another agent

4.5.3 Wellbeing

In relation to the wellbeing of the agent, as has been shown, said wellbeing is a function of the drives. Therefore, adapting equation (1) to the agent's design:

$$Wb = Wb_{ideal} - (\alpha_1 D_{hunger} + \alpha_2 D_{thirst} + \alpha_3 D_{weakness} + \alpha_4 D_{loneliness}) \tag{24}$$

Where: $Wb_{ideal} = 100$.

The personality factors, α_i, weigh the importance of each drive on the wellbeing of the agent. In the experiments all the drives will have the same importance, therefore, all the personality factors are equal to one other:

$$\alpha_i = 1 \tag{25}$$

By varying these personality factors we could design different kinds of agents. For example, if we increase the value of α_4, the personality factor related to the Loneliness drive, the lack of social interaction will imply a big decrease of the wellbeing of the agent (sadness). Therefore, since the reinforcement function is related to the variation of the wellbeing (emotions), this agent will be very sociable.

4.5.4 State of the agent

According to section 3.5.1, in this scenario the inner state of the agent is defined as follows:

$$S_{inner} = \{Hungry,\ Thirsty,\ Weak,\ Alone,\ OK\} \tag{26}$$

In relation to the external state, the state related to every passive object, except for the object world, are the combination of three binary variables:

$$S_{obj} = Being\_in\_posession\_of \times Being\_next\_to \times Knowing\_where\_to\_find \tag{27}$$

In relation to the object world, at the moment, the state of the agent in relation to the world is unique, the agent is always in the world:

$$S_{world} = Being\_at \qquad (Always\ True) \tag{28}$$

Finally, in relation to another agent:

$$S_{agent} = Being\_next\_to \tag{29}$$

Every variable is evaluated as $= \{true, false\}$.

Therefore, according to the definition of the state given by the equation (3.6), the agent could be, for example, in the following state in relation to food: " hungry, not having food, not being next to food and knowing where to find food".

4.5.5 Actions of the agent

The sets of actions that the agent can execute, depending on his state in relation to the objects, are the following:

$$A_{food} = \{Eat,\ Get,\ Go\ to\} \tag{30}$$

$$A_{water} = \{Drink\ water,\ Get,\ Go\ to\} \tag{31}$$

$$A_{medicine} = \{Drink\ medicine,\ Get,\ Go\ to\} \tag{32}$$

$$A_{another\ agent} = \begin{cases} Steal\ food/water/medicine \\ Give\ food/water/medicine \\ Greet \\ To\ do\ nothing \\ Kick \end{cases} \tag{33}$$

$$A_{world} = \{Keep\ still,\ Explore\} \tag{34}$$

Among all these behaviours there are some of which cause an increase or decrease of some drives, as is shown in table 2, leading to a variation of the wellbeing of the agent:

The "do nothing" action has no effect on the drives of the agent.

4.6 Indicators of performance of the agent

Other authors, [Ávila García and Cañamero, 2002], defined some viability indicators to compare the performance of several agents (robots) that use different decision making architectures. Taking those indicators as a reference, in this work two different indicators are defined for the analysis of the obtained results. For this reason it has to be taken into account that during the experiments carried out, the agents do not die. The agents have a fixed time life. The performance of the agent is determined by the analysis of the wellbeing

Table 2: Effects of the actions over drives

Action	Drive	Effect
Eat	Hunger	Reduce to zero (drive satisfaction)
Drink water	Thirst	Reduce to zero (drive satisfaction)
Drink medicine	Weakness	Reduce to zero (drive satisfaction)
To be greeted	Loneliness	Reduce to zero (drive satisfaction)
To be stolen	Loneliness	Increase certain amount
To be given	Loneliness	Reduce to zero (drive satisfaction)
To be kicked	Loneliness	Increase certain amount
To be kicked	Weakness	Increase certain amount
Explore/ go to	Weakness	Increase certain amount

of the agent, since this information gives an accurate idea as to how well the experiment went.

First, an important concept needs to be introduced: Security Zone. The Security Zone is defined as an interval of wellbeing values, in such a way that it can be considered that if the wellbeing of the agent is inside this interval, then the agent is "doing good". The interval of the Security Zone is defined as $SZ = [100, 92]$

In order to analyze the performance of the agent for every experiment, two indicators have been defined:

1. The average value of wellbeing: This indicator gives a general idea about the performance of the agent, but it does not gives a clear idea about quality of life. This average value of wellbeing can be high due to a good general performance, as well as due to very good and very bad moments.

2. Percentage of permanence inside the Security Zone: This indicator gives a more obvious idea about the quality of life of the agent during the experiment. What is important for the experiment is not only that the agent has a good average value of wellbeing, but also that the agent has a good life.

5 Experimental results: Solitary Agent

In this section, the behaviour of an agent living on his own in the previously described world, is presented. Moreover, we have decided to do the learning parameters adjustment in this environment. These parameters are the following:

- The parameter δ that defines the relation between exploration and exploitation of the actions.

- The learning rate α which controls how much weight is given to the reward just experienced.

- The discounted factor γ that defines how much expected future rewards affect decisions now.

In order to learn a good policy of behaviour all the available actions in every state have to be executed. Therefore, the agent decides which action is the most suitable for every state. In each experience, the agent updates the Q value of every state-action pair. Finally, the most suitable actions for every state will have a high Q value. In order to guarantee that all the actions are explored, the parameter δ must be high. This parameter has already been introduced in section 3.4 and determines the randomness when selecting an action. When its value is high, all the actions have the same probability of being selected, with no preference among them. Since all the possible actions are executed, it causes some of them not to be the most suitable ones. As a consequence, the wellbeing of the agent decreases.

On the other hand, the learning rate α was introduced in equation (4), where the updating of the Q values of the reinforcement learning was defined. The effect of this parameter is to give more or less importance to the learnt values than to the new experiences. A high value of this parameter causes very sudden changes in the learnt Q values. On the other hand, a very low value of this parameter causes the learning process to be slow. This is because the agent is very conservative and he gives little importance to the new experiences. The best option would be an intermediate value of this learning rate. Based on several experiments it was proved that an intermediate value guarantees a good relation between the variability of the Q values and the importance of the new experiences.

There is another parameter related to the Q-learning process (13), which defines how much expected future rewards affect present decisions. This is the parameter γ, called the discount factor. As was explained in section 3.3, 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. The updating of the Q value, for every state-action pair, is formed by two contributions: the reward received in that moment (r) and the importance of the best that can happen from the new state ($\gamma \cdot V^{obj_i}(s')$).

The experiments showed that in order to get the agent to learn a proper sequence of actions, the discount factor γ must be high. In Figure 4 the Q values of the actions related to the food when Hunger is the dominant motivation, and the value of the discount factor γ is high, are shown.

When the agent is hungry, next to food, has food and then eats, the Q value is high due to the received reward (see the graph on the bottom right corner of Figure 4). The next time the agent is beside food and gets the food, the Q value of this action is updated as has been previously explained. The new state is "to have food", and the best thing that the agent can do is to eat it, therefore the value of this new state is high. As a consequence, although the agent did

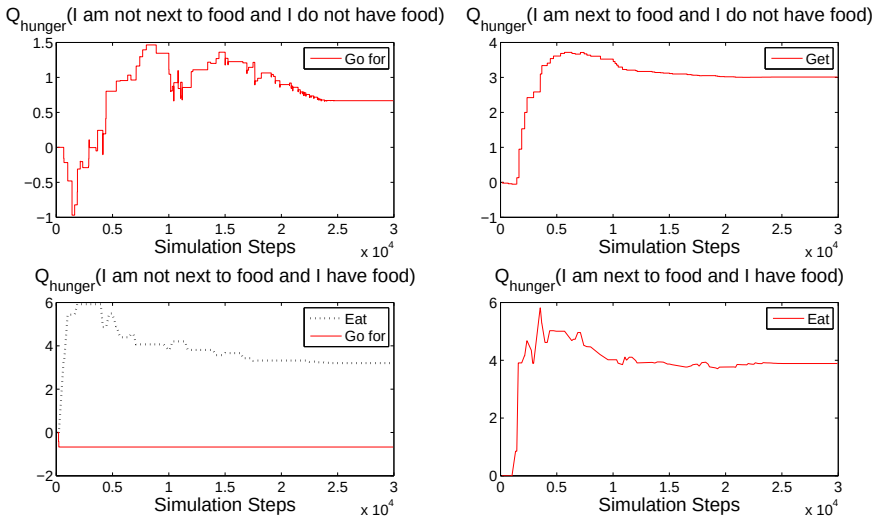

Figure 4: Q values of the actions related to food when the agent is hungry, with $\gamma = 0.8$

not receive an immediate reward for getting the food, the fact that the value of the new state is multiplied by a high value of γ will cause the Q value of "get food" to be high (see the graph on the top right corner of Figure 4). The same will happen when the agent executes the action "go for food" and the new state will be " to be next to food" since its value, as has just been shown, is high. As a conclusion, when the agent uses a high value of γ, he learns the sequence of behaviours that leads him to satisfy the Hunger drive correctly.

Each experiment consists of two phases: the *learning phase* and the *steady phase*. During the learning phase, the agent starts with all the initial Q values equal to zero. The agent, through his experience in the world, learns and updates his Q values. Once the learning phase has finished, the steady phase starts. In this last phase the agent "lives" according to the learnt Q values.

During the learning phase the values of the parameters δ, that determines the exploration level, and the learning rate α decrease gradually. In the steady phase, the agent exploits the learnt policy of behaviour and stops learning. Therefore, in this steady phase the actions executed are the ones whose Q values are the highest and this implies that δ is very low. Since the agent stops learning, the learning rate is equal to zero $\alpha = 0$.

In Figure 5, the wellbeing of the agent along both phases is shown. As can be observed, during the learning phase the wellbeing of the agent increases gradually, being higher than 95 at the end of this phase. In the steady phase wellbeing maintains its high value, in fact the average value is 98.51 and the percentage of permanence in the Security Zone is 100%. Therefore, it can be considered that the agent learned a good policy of behaviour.

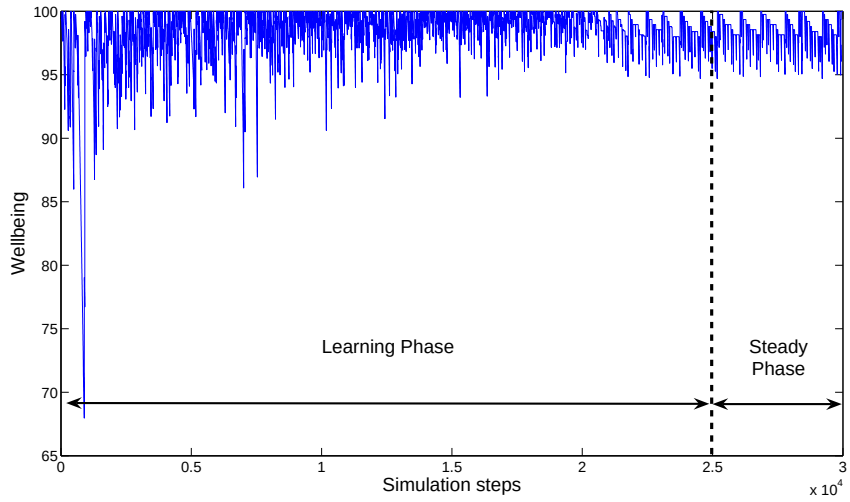

Figure 5: Wellbeing of the agent during the learning and steady phases

In the table 3, the most suitable values of all the parameters involved in the agent's design are shown:

Table 3: Parameters of the agent

	Learning Phase	*Steady Phase*
δ	$1.8 \rightarrow 0.1$	0.1
α	$0.3 \rightarrow 0$	0
γ	0.8	0.8

6 Experimental results: Accompanied Agent

In the case that the agent lives with another agents, the Loneliness drive appears in order to cause social interaction. In the experiments the agent has to live with different kinds of opponents who have fixed policies of behaviour: one of them is a good one, another a bad one and finally, the neutral one. Depending on the kinds of opponents that are living with the agent, different types of worlds are described: the good world, the bad world, the neutral world and the mixed world.

When social interaction exists, the rewards received by the agent depend not only on his own actions but on the action executed by the other agent. Therefore, several multi-agent reinforcement learning algorithms were tested in every world, such as the Friend or Foe algorithm. This algorithm was developed for general-sum games [Littman, 2001]. Nevertheless, it was proved that those algorithms do not give any significant advantages in comparison with the Q-learning algorithm. This is because the multi-agent learning algorithms are developed based on game theory, and this implies that both agents are adaptable, which means that both of them are learning. This is not the case for the proposed environments where the opponents have fixed policies of behaviour and therefore, it seems to make sense that the best results are obtained when the agent uses the new Q-learning algorithm proposed in this work.

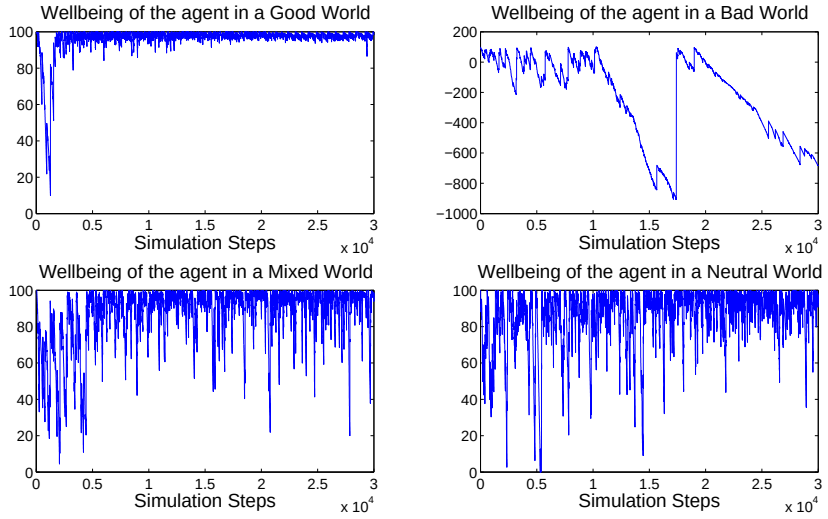

Figure 6: The wellbeing of the agent in every world when using the new Q-learning

In Figure 6, the wellbeing of the agent during the learning and the steady phase when the agent uses during the social interaction the new Q-learning algorithm, is shown for every different world. As can be seen, when the agent lives in a good world, i.e. all his opponents are good, wellbeing is almost near the ideal value during half of the learning phase and during the entire steady phase. In fact, the values of the indicators of performance of the agent, during the steady phase, in this world are very high as is shown in table 4.

On the contrary, when the agent shares the environment with three bad opponents, this is a bad world, then as is shown, the wellbeing of the agent is quite negative at the end of the steady phase. Therefore, in this world the indicators of the performance of the agent are not calculated. The wellbeing is so negative because the opponents kick him or to steal objects and therefore,

the agent will not be able to satisfy his Loneliness drive that, as a consequence, increases its value indefinitely. This means that the agent learns to avoid the social interaction in a bad world, since most of those interactions have very negative results.

In the case that the agent lives in a neutral or mixed world, the results obtained seem to be similar to one other. In these worlds the results of social interaction can be positive or negative and therefore, when the agent tries to satisfy his Loneliness drives sometimes he is able to do so but not always. This is the reason why in Figure 6 the wellbeing of the agent in these worlds has so many dips along both phases. In spite of the existence of all those drops, the indicators, as can be observed in table 4, are quite acceptable.

Table 4: Indicators of performance

World	*Average Value*	*% of permanence inside SZ*
Good	99.2	100
Neutral	90.68	58.7
Mixed	91.81	71
Bad	NO	NO

7 Conclusions

As was shown in the introduction, the final objective of this work is to design a decision making system, using unsupervised learning, for an autonomous and social agent. In order to carry out this objective, this decision making system was implemented in a virtual agent who lives in a MUD called CoffeeMud.

The agent lives in the "Passage" area where he can find different environments. In this paper the performance of the agent living in different kinds of environment has been presented. Looking at these results, we tested the developed decision making system for the agent. The experiments can be separated in two main parts: First, the agent living alone in "Passage", this means with no any other agent, and secondly, the agent living accompanied by another agents with different personalities.

In the first set of experiments it has been proved that, in order to learn a good policy, the agent has to first explore all possible actions. In relation to the learning rate, it was proved that the agent learns correctly with an intermediate-low value. In order to take advantage of the knowledge acquired, we decided to separate the life of the agent into two phases: the learning phase and the steady phase.

During the learning phase the exploration level and the learning rate decrease gradually. This implies that the agent, at the beginning of this phase, explores all the actions and learns from his experience. As the agent lives, he starts to exploit the actions that led to good results, as well as to give less importance to new experiences, i.e. the agent begins to be more conservative. During the steady phase the agent stops learning and lives according to the policy learned. Moreover, it has also been proved that, in order to learn a correct policy, the discounted factor must be high. It is necessary, when evaluating an action taken in a certain state, to consider future rewards.

As a conclusion, when the agent lives alone in the MUD he is able to learn an appropriate policy of behaviour by himself. The agent uses a modification of the Q-learning algorithm to learn the correct function between states and actions. Using the variation of the wellbeing of the agent, happiness and sadness, as the reinforcement function, the agent learns to survive by maintaining all his drives in acceptable ranges. Therefore, emotions are used not for external communication of the agent but for controlling the learning process.

When the agent lives with other agents he uses the values of the parameters previously tuned. In relation to the need for social interaction, a new drive was implemented: Loneliness. This drive was implemented so that the agent satisfied it by interacting with another agent; therefore, the agent needs to interact with others in order to survive.

It has been proved that the agent is also able to learn appropriate policies of behaviour when he shares his environment with other agents. The best learning algorithm to deal with the social interaction is the new algorithm based on Q-learning. Using this algorithm, the agent was able to survive in a complex world maintaining his drives with low values.

As the main conclusion of this paper, we proved the usefulness of using a MUD for developing and testing a decision making system. This text based game gave us the possibility of creating simple as well as complex environments in a very direct way. As the experimental results showed, the proposed decision making produce very natural results giving the agent a life-like appearance, which is very useful for the design of games.

Acknowledgements

The authors gratefully acknowledge the funds provided by the Spanish Government through the projects called “Peer to Peer Robot-Human Interaction” (R2H), of MEC (Ministry of Science and Education) and the project “A new approach to social robotics” (AROS), of MICINN (Ministry of Science and Innovation). Moreover, the research leading to these results has received funding from 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

[Arkin, 1988] Arkin, R. C. (1988). Homeostatic control for a mobile robot: Dynamic replanning in hazardous environments. In *SPIE Conference on Mobile Robots, Cambrige, MAA*.

[Bakker et al., 2003] Bakker, B., Zhumatiy, V., Gruener, G., and Schmidhuber, J. (2003). A robot that reinforcement-learns to identify and memorize important previous observations. In *the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS2003*.

[Balkenius, 1993] Balkenius, C. (1993). Motivation and attention in an autonomous agent. In *Workshop on Architectures Underlying Motvation and Emotion WAUME 93, University of Birmingham*.

[Balkenius, 1995] Balkenius, C. (1995). *Natural Intelligence in Artificial Creatures*. PhD thesis, Lund University Cognitive Studies 37.

[Bellman, 2003] Bellman, K. L. (2003). *Emotions in Humans and Artifacts*, chapter Emotions: Meaningful mappings between the individual and its world. MIT Press.

[Berridge, 2004] Berridge, K. C. (2004). Motivation concepts in behav-ioural neuroscience. *Physiology and Behaviour*, (81):179–209.

[Bonarini et al., 2006] Bonarini, A., Lazaric, A., Restelli, M., and Vitali, P. (2006). Self-development framework for reinforcement learning agents. In *the 5th International Conference on Developmental Learning (ICDL)*.

[Cañamero, 1997] Cañamero, L. (1997). Modeling motivations and emotions as a basis for intelligent behavior. In *First International Symposium on Autonomous Agents (Agents'97), 148-155. New York, NY: The ACM Press.*

[Cañamero, 2003] Cañamero, L. (2003). *Emotions in Humans and Artifacts*, chapter Designing emotions for activity selection in autonomous agents. MIT Press.

[Damasio, 1994] Damasio, A. (1994). *Descartes' Error - Emotion, reason and human brain*. Picador, London.

[Fong et al., 2002] Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2002). A survey of socially interactive robots: Concepts, design, and applications. Technical report, CMU-RI-TR-02-29.

[Gadanho, 1999] Gadanho, S. (1999). *Reinforcement Learning in Autonomous Robots: An Empirical Investigation of the Role of Emotions*. PhD thesis, UNiversity of Edinburgh.

[Gautier and Boeree, 2005] Gautier, R. and Boeree, G. (2005). *Teorías de la Personalidad: una selección de los mejores autores del S. XX*.

[Hull, 1943] Hull, C. L. (1943). *Principles of Behavior*. New York: Appleton Century Crofts.

[Humphrys, 1997] Humphrys, M. (1997). *Action Selection methods using Reinforcement Learning*. PhD thesis, Trinity Hall, Cambridge.

[Isbell et al., 2001] Isbell, C., Shelton, C. R., Kearns, M., Singh, S., and Stone, P. (2001). A social reinforcement learning agent. In *the fifth international conference on Autonomous agents, Montreal, Quebec, Canada*.

[Littman, 2001] Littman, M. (2001). Friend-or-foe q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learn-ing*, pages 322–328.

[Lorenz and Leyhausen, 1973] Lorenz, K. and Leyhausen, P. (1973). *Motivation of human and animal behaviour; an ethological view*, volume xix. New York: Van Nostrand-Reinhold.

[Martinson et al., 2002] Martinson, E., Stoytchev, A., and Arkin, R. (2002). Robot behavioral selection using q-learning. In *of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), EPFL, Switzerland*.

[Ortony, 2003] Ortony, A. (2003). *Emotions in Humans and Artifacts*, chapter On making Believable Emotional Agents Believable, pages 188–211. MIT Press.

[Ortony et al., 1988] Ortony, A., Clore, G. L., and Collins, A. (1988). *The Cognitive Structure of Emotions*. Cambridge University Press. Cambridge, UK.

[Ribeiro et al., 2002] Ribeiro, C. H. C., Pegoraro, R., and RealiCosta, A. H. (2002). Experience generalization for concurrent reinforcement learners: the minimax-qs algorithm. In *AAMAS 2002*.

[Rolls, 2003] Rolls, E. (2003). *Emotions in Humans and Artifacts*, chapter Theory of emotion, its functions, and its adaptive value. MIT Press.

[Santa-Cruz et al., 1989] Santa-Cruz, J., Tobal, J. M., Vindel, A. C., and Fernández, E. G. (1989). Introducción a la psicología. Facultad de Psicología. Universidad Complutense de Madrid.

[Smart and Kaelbling, 2002] Smart, W. D. and Kaelbling, L. P. (2002). Effective reinforcement learning for mobile robots. In *International Conference on Robotics and Automation (ICRA2002)*.

[Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, A Bradford Book.

[Thomaz and Breazeal, 2006] Thomaz, A. L. and Breazeal, C. (2006). Transparency and socially guided machine learning. In *the 5th International Conference on Developmental Learning (ICDL)*.

[Toates, 1986] Toates, F. (1986). *Motivational systems*. Cambridge (MA): Cambridge Univ. Press.

[Watkins, 1989] Watkins, C. J. (1989). *Models of Delayed Reinforcement Learning*. PhD thesis, Cambridge University, Cambridge, UK.

[Zimmerman, 2007] Zimmerman, B. (2007). http://www.coffeemud.org/.

[Ávila García and Cañamero, 2002] Ávila García, O. and Cañamero, L. (2002). A comparison of behavior selection architectures using viability indicators. In *Proc. International Workshop Biologically-Inspired Robotics: The Legacy of W. Grey Walter(WGW'02)*.

[Ávila García and Cañamero, 2004] Ávila García, O. and 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)*.