Universidad
Carlos III de Madrid

TESIS DOCTORAL

Bio-inspired decision making system for an autonomous social robot. The role of fear.

Autor:
Álvaro Castro González
Directores:
Miguel Ángel Salichs Sánchez-Caballero
María Malfaz Vázquez

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

Leganés, December 2012

TESIS DOCTORAL (DOCTORAL THESIS)

BIO-INSPIRED DECISION MAKING SYSTEM FOR AN AUTONOMOUS SOCIAL ROBOT. THE ROLE OF FEAR.

Autor (Candidate): Álvaro Castro González

Director (Adviser): Miguel Ángel Salichs Sánchez-Caballero
Directora (Adviser): María Malfaz Vázquez

Tribunal (Review Committee)

Firma (Signature)

Presidente (Chair): ________________

Vocal (Member): ________________

Secretario (Secretary): ________________

Título (Degree): Doctorado en Ingeniería Eléctrica, Electrónica y Automática
Calificación (Grade): ____________________

Leganés, de de

A mis padres

"En la vida hay que valer para todo, para trabajar y para ir de fiesta"
— Diómedes Castro Rodríguez

Agradecimientos

Son muchas las veces que, a lo largo de estos años, he pensado en este momento: en cómo sería escribir las últimas líneas de mi tesis, desde dónde lo haría, de quién me iba a olvidar, o qué palabras usar. Llegado este momento, no se me ocurre mejor lugar que donde me encuentro: sentado en la mesa donde tantas horas he pasado delante del ordenador que me ha tenido que soportar.

Son muchas las personas que se amontonan en mi cabeza esperando para ser agradecidas, así que vamos por orden. No podía empezar por otro que no fuera Miguel Ángel, mi director, mi mentor y mi "padre" científico. Siempre dispuesto, a cualquier hora y desde cualquier sitio, para lo que se necesite. Él fue el que un día me dio la oportunidad para entrar en este mundo de la robótica. Si antes he hablado de mi "padre" científico, María es mi "hermana mayor" científica. Trabajar codo con codo con ella ha sido una suerte. Minuciosa, detallista, rigurosa, y también comprensiva e inteligente. Sus consejos, en todos los ámbitos, han sido una ayuda inestimable. Ambos han conseguido que estos años de trabajo e investigación sean apasionantes y por ello les estoy muy agradecido.

Mis padres, su ejemplo, su educación, nuestra forma de vida, mi familia, ha constituido la base para que yo hoy defienda una tesis doctoral. Nunca podré expresarles lo agradecido que estoy por todos los esfuerzos que hacen y todo lo que me han enseñado: la capacidad de esfuerzo y sacrificio, el trabajo constante, cómo poner un vivero de chopos, a cultivar remolacha o menta o maíz, cómo regar una tierra, a recoger la hierba, a ir a *hacendera*, a cuidar la huerta, a quitar malas hierbas, la matanza, a ir a por leña y picarla, andar en bici, y tantas otras cosas que, todas en conjunto, han logrado que yo hoy sea lo que soy, y lo que no soy. Por supuesto, el elemento clave ha sido los *tuppers* de mi madre que tanta envidia causan a cualquiera que los ve. Gracias por todo.

Desde pequeño he tenido un modelo, alguien a quien siempre he imitado en las cosas que hacía, en cómo vestía, y hasta en cómo se ¡peinaba! Mi hermano Raúl, con su insaciable curiosidad, siempre ha puesto un punto crítico y ha ido abriendo *puertas* difíciles de abrir que otros ya nos las hemos encontrado entreabiertas cuando llegamos a ellas. Ahora todos disfrutamos de su gran familia, con Caro, Sergio, y el último “fichaje” Irene. Gracias.

Qué decir de Tere que no se me vea en la cara cuando estamos juntos. Ella ha conseguido hacerme feliz a su lado, siempre me ha apoyado y, lo que es más importante, sé que siempre lo va a seguir haciendo. A pesar de las múltiples veces que yo me he equivocado, ella siempre estaba allí para ayudarme a solucionarlo. En ella he descubierto a la mejor persona que he conocido. Le ha tocado desempeñar muchos papeles: pareja, compañera, asesora (personal y profesional), pintora, etc., y todos ellos los ha desempeñado a la perfección. Gracias por estar en mi vida.

Por supuesto mucho tengo que agradecer a mis compañeros y amigos del departamento. En especial a Alberto, Paolo y Fer que siempre han estado cerca y con los que he pasado tantas *vivencias*. Espero que sigamos teniendo nuevas *vivencias*. También a los compañeros de despacho, Javi, Carla, Fer y Ana, con los que he compartido muchas horas de discusiones, música, risas, consultas y hasta problemas personales. Especial cariño tengo a todos los “sociales” que han hecho realidad a Maggie: Javi, Fernando Alonso, David, Rafa, Arnaud, Ana, Alberto, Kike, Luis, y tantos otros que han pasado por el labo y con los que he compartido tantas alegrías y problemas en el día a día. Obligado agradecimiento se merecen Sonia y Edu que, con su eterna paciencia y sus soluciones rápidas y eficaces, logran que todo trámite sea más fácil. También me gustaría agradecer al resto de miembros del Robotics Lab y del departamento que han logrado que ir a trabajar cada día sea muy fácil. A todos gracias.

A pesar de estar lejos de ellos, mis amigos de León siempre han logrado que tenga esa sensación de “estar en casa” cuando vuelvo. Vity, Alberto, Moral, Dani, Estrada, Patri, Arola, Emma y un largo etcétera son los responsables de que nunca me canse de volver a mi tierra. Gracias *cazurros*.

No quiero olvidarme de los investigadores con los que he trabajado durante mis estancias en Japón, Takayuki Kanda y Masahiro Shiomi, y en Inglaterra, Farshid Amirabdollahian y Daniel Polani; así como los compañeros de estos laboratorios lejos de casa y que me aceptaron rápidamente como uno más. Ellos lograron que estas experiencias sean inolvidables.

Por último agradecer a la Universidad Carlos III de Madrid y a los distintos ministerios del Gobierno de España la financiación que durante estos años ha permitido que desarrolle esta tesis.

A todos gracias.

Abstract

Robotics is an emergent field which is currently in vogue. In the near future, many researchers anticipate the spread of robots coexisting with humans in the real world. This requires a considerable level of autonomy in robots. Moreover, in order to provide a proper interaction between robots and humans without technical knowledge, these robots must behave according to the social and cultural norms. This results in social robots with cognitive capabilities inspired by biological organisms such as humans or animals.

The work presented in this dissertation tries to extend the autonomy of a social robot by implementing a biologically inspired decision making system which allows the robot to make its own decisions. Considering this kind of decision making system, the robot will not be considered as a slave any more, but as a partner.

The decision making system is based on drives, motivations, emotions, and self-learning. According to psychological theories, drives are deficits of internal variables or needs (e.g. energy) and the urge to correct these deficits are the motivations (e.g. survival). Following a homeostatic approach, the goal of the robot is to satisfy its drives maintaining its necessities within an acceptable range, i.e. to keep the robot's wellbeing as high as possible. The learning process provides the robot with the proper behaviors to cope with each motivation in order to achieve the goal.

In this dissertation, emotions are individually treated following a functional approach. This means that, considering some of the different functions of emotions in animals or humans, each artificial emotion plays a different role. Happiness and sadness are employed during learning as the reward or punishment respectively, so they evaluate the performance of the robot. On the other hand, fear plays a motivational role, that is, it is considered as a motivation which impels the robot to avoid dangerous situations. The benefits of these emotions in a real robot are detailed and empirically tested.

The robot decides its future actions based on what it has learned from previous experiences. Although the current context of this robot is limited to a laboratory, the social robot cohabits with humans in a potentially non-deterministic environment. The robot is endowed with a repertory of actions but, initially, it does not know what action to execute either when to do it. Actually, it has to learn the policy of behavior, i.e. what action to execute in different world configuration, that is, in every state, in order to satisfy the drive related to the highest motivation. Since the robot will be learning in a real environment interacting with several objects, it is desired to achieve the policy of behavior in an acceptable range of time.

The learning process is performed using a variation of the well-known Q-Learning algorithm, the Object Q-Learning. By using this algorithm, the robot learns the value of every state-action pair through its interaction with the environment. This means, it learns the value that every action has in every possible state; the higher the value, the better the action is in that state. At the beginning of the learning process these values, called the Q values, can all be set to the same value, or some of them can be fixed to another value. In the first case, this implies that the robot will learn from scratch; in the second case, the robot has some kind of previous information about the action selection. These values are updated during the learning process.

The emotion of fear is particularly studied. The generation process of this emotion (the appraisal) and the reactions to fear are really useful to endow the robot with an adaptive reliable mechanism of "survival". This dissertation presents a social robot which benefits from a particular learning process of new releasers of fear, i.e. the capacity to identify new dangerous situations. In addition, by means of the decision making system, the robot learns different reactions to prevent danger according to different unpredictable events. In fact, these reactions to fear are quite similar to the fear reactions observed in nature.

Another challenge is to design a solution for the decision making system in such a way that it is flexible enough to easily change the configuration or even apply it to different robots.

Considering the bio-inspiration of this work, this research (and other related works) was born as a try to better understand the brain processes. It is the author's hope that it sheds some light in the study of mental processes, in particular those which may lead to mental or cognitive disorders.

Resumen

La robótica es un área emergente que actualmente se encuentra en boga. Muchos científicos pronostican que, en un futuro próximo, los robot cohabitarán con las personas en el mundo real. Para que esto llegue a suceder, se necesita que los robots tengan un nivel de autonomía considerable. Además, para que exista una interacción entre robots y personas sin conocimientos técnicos, estos robots deben comportarse de acuerdo a las normas sociales y culturales. Esto nos lleva a robots sociales con capacidades cognitivas inspiradas en organismos biológicos, como los humanos o los animales.

El trabajo que se presenta en esta tesis pretende aumentar la autonomía de un robot social mediante la implementación de un sistema de toma de decisiones bioinspirado que permita a un robot tomar sus propias decisiones. Desde este punto de vista, el robot no se considerará más como un esclavo, sino como un compañero.

El sistema de toma de decisiones está basado en necesidades (*drives*), motivaciones, emociones y auto-aprendizaje. De acuerdo a diversas teorías psicológicas, las necesidades son carencias o déficits de variables internas (por ejemplo, la energía) y el impulso para corregir estas necesidades son las motivaciones (como por ejemplo la supervivencia). Considerando un enfoque homeostático, el objetivo del robot es satisfacer sus carencias manteniéndolas en un nivel aceptable. Esto quiere decir que el bienestar del robot debe ser lo más alto posible. El proceso de aprendizaje permite al robot desarrollar el comportamiento necesario según las distintas motivaciones para lograr su objetivo.

En esta tesis, las emociones son consideradas de forma individual desde un punto de vista funcional. Esto significa que, considerando las diferentes funciones de las emociones en animales y humanos, cada una de las emociones artificiales juega un papel diferente. Por un lado, la felicidad y la tristeza se usan durante el aprendizaje como refuerzo o castigo respectivamente y, por tanto, evaluan el comportamiento del robot. Por otro lado, el miedo juega un papel motivacional, es decir, es considerado como una motivación la cual

“empuja” el robot a evitar las situaciones peligrosas. Los detalles y las ventajas de estas emociones en un robot real se muestran empíricamente a lo largo de este libro.

El robot decide sus acciones futuras en base a lo que ha aprendido en experiencias pasadas. A pesar de que el contexto actual del robot está limitado a un laboratorio, el robot social cohabita con personas en un entorno potencialmente no-determinístico. El robot está equipado con un repertorio de acciones pero, inicialmente, no sabe qué acción ejecutar ni cuando hacerlo. De echo, tiene que aprender la política de comportamiento, esto es, qué acción ejecutar en diferentes configuraciones del mundo (en cada estado) para satisfacer la necesidad relacionada con la motivación más alta. Puesto que el robot aprende en un entorno real interaccionando con distintos objetos, es necesario que este aprendizaje se realice en un tiempo aceptable.

El algoritmo de aprendizaje que se utiliza es una variación del conocido Q-Learning, el Object Q-Learning. Mediante este algoritmo el robot aprende el valor de cada par estado-acción a través de interacción con el entorno. Esto significa, que aprende el valor de cada acción in cada posible estado. Cuanto más alto sea el valor, mejor es la acción en ese estado. Al inicio del proceso de aprendizaje, estos valores, llamados valores Q, pueden tener todos el mismo valor o pueden pueden tener asignados distintos valores. En el primer caso, el robot no dispone de conocimientos previos; en el segundo, el robot dispone de cierta información sobre la acción a elegir. Estos valores serán actualizados durante el aprendizaje.

La emoción de miedo es especialmente estudiada en esta tesis. La forma de generarse esta emoción (el *appraisal*) y las reacciones al miedo resultan realmente útiles a la hora de dotar al robot con un mecanismo de supervivencia adaptable y fiable. Esta tesis presenta un robot social que utiliza un proceso particular para el aprendizaje de nuevos “liberadores” del miedo, es decir, dispone de la capacidad de identificar nuevas situaciones peligrosas. Además, mediante el sistema de toma de decisiones, el robot aprende diferente reacciones para protegerse ante posibles daños causados por diversos eventos impredecibles. De echo, estas reacciones al miedo son bastante similares a las reacciones al miedo que se pueden observar en la naturaleza.

Otro reto importante es el diseño de la solución: el sistema de toma de decisiones tiene que diseñarse de forma que sea suficientemente flexible para permitir cambiar fácilmente la configuración o incluso para aplicarse a distintos robots.

Teniendo en cuenta el enfoque bioinspirado de este trabajo, esta investigación (y muchos otros trabajos relacionados) surge como un intento de entender un poco más lo que sucede en el cerebro. El autor espera que esta tesis pueda ayudar en el estudio de los procesos mentales, en particular aquellos que pueden llevar a desórdenes mentales o cognitivos.

Contents

List of Tables

List of Figures

List of Algorithms

Code listings

CHAPTER 1

Introduction

1.1 Motivation

The current society is aging. According to the data obtained from the Spanish National Statistics Institute, the Spanish population is getting older and this tendency will remain, at least, for the next forty years. As shown in Figure 1.1 [1] , the Spanish population pyramid is expected to get wider in upper levels over the years. This corresponds to a constrictive pyramid, which means lower percentages of young people and, in general, an elder population. This is often a typical pattern of a developed country. It results in an increment of the percentage of the dependency ratio (everyone out of working age). In fact, it is expected that in 2049 the dependency ratio reaches the $89,6\%$, from the current $47,8\%$. These data correspond to Spain, however a similar tendency can be observed in most of the developed countries.

The consequences of the aging of the population are that much more people will demand different services and, probably, the available labor force and the economic resources will not be enough for providing the required services. In this context, robots are a promising tool for increasing the labor capacity of a society, and their cost will be reduced once they are mass produced. The development of new robots, which will be able to perform tasks in the same manner (or at least close) as humans do, can be a solution to many services where, nowadays, humans cannot be replaced. Among other tasks, robots are already carrying out several works traditionally achieved by humans: performing as museum guides [9], handling explosives [10], delivering medicines in hospitals [11], assisting elders in

[1] This plot has been obtained from the web of the National Statistical Institute of Spain [8]

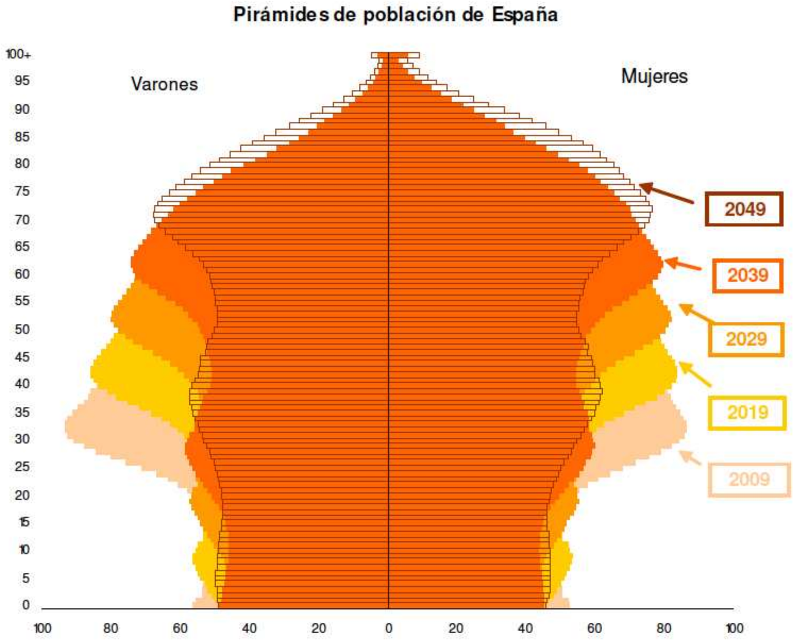


Figure 1.1: Simulated future evolution of Spain pyramid population

daily shopping [12], facilitating daily tasks to handicapped people [13], or transporting in industrial productions [14].

Most of these tasks implies that people directly interact with the robot. Apparently, human-robot interactions will spread fast. Bill Gates titled a paper at Scientific American magazine as “A robot in every house” (Jan. 2007). This is just the forecast of a relevant person, but this idea has been going round and round for years. The increase of robots foresees a widespread use of robots living with humans. It is expected that in a near future, personal robots will be endowed with enough autonomy to work and live in an individual’s home. For these reasons, social robots (those robots interacting with humans in natural ways) need to be able to decide their own actions (autonomy), to make deliberative plans (reasoning), and to have an emotional behavior in order to facilitate the human-robot interaction.

The expansion of social robots will bring people without any knowledge about robots trying to interact with them in a natural way, i.e. in the same manner they do with animals or other humans. Therefore, these robots must be endowed with the required abilities to provide a proper human-robot interaction and life-like appearance. In order to achieve these capabilities, robots must be endowed with capacities inspired by humans, or at least by animals. Thus, it is desired to equip robots with *cognitive* capacities which provide enough *autonomy* to develop their tasks. Accordingly, cognitive and physical human-robot interaction are nowadays among the most studied aspects of the robotics [15].

1.1.1 Cognitive robotics

During the last few years, the interest in robots which are integrated in our everyday environment, personal robots, has increased [16]. Human-robot interaction is one of the main characteristics of these robots. Therefore, many efforts have been put into human-robot interaction. In order to facilitate it, the robotic research is now centered on cognitive robotics which addresses the emerging field of autonomous systems with artificial reasoning skills.

In the Nineties, the term "cognitive robotics" was first introduced by Ray Reiter and his colleagues, who have a research group in this topic at the University of Toronto. According to them, cognitive robotics is concerned with endowing robotic or software agents with higher level cognitive functions that involve reasoning about goals, perception, actions, mental states of other agents, collaborative task execution, etc. In 1997, Brooks defined *cognitive robotics* [17] as the field aimed to give the robot cognitive abilities that make the robot forms and develops knowledge and skills independently and gradually through cognitive processes. The idea is to extend the robots' abilities in order to implement some of the high level cognitive functions. Some examples of high level cognitive functions already implemented in robots are surprise [18], developmental learning [19], and deception [20].

Moreover, at the beginning of the Sixties, the artificial intelligence precursor Herbert Simon was convinced that including emotions in the cognitive model to approximate the human mind was necessary [21]. Later, near the mid Nineties, Antonio Damasio published *Descartes's Error* [22]. His studies proved that damage to the brain's emotional system caused the patient to make poor judgments despite intact logical reasoning skills. As a consequence, the positive role of human emotions in cognition started to gain prominence among a group of researchers from the scientific community. Later, other studies showed that emotions have influence on many cognitive mechanisms, such as memory, attention, perception, and reasoning [23, 24, 25, 26]. Besides, emotions play a very important role in survival, social interaction and learning of new behaviors [27, 28, 29].

Therefore, in recent years, the role of emotional mechanisms in natural and artificial cognitive architectures, in particular in cognitive robotics, has become very popular. According to Ziemke [30], in relation to the main question: do robots need emotions? many researchers have answered positively, mainly considering the two aspects of emotion: the external (social) one and the internal (individual) one. It seems to be obvious that in human-robot social interaction, expression of emotions helps to make interaction more natural [31]. On the other hand, the internal aspects of emotion, i.e. its role in the behavioral organization of an individual cognitive agent, are essential for the autonomy issue. This is the main concern of this dissertation.

1.1.2 Autonomy

Autonomy is a term widely used in literature and its meaning ranges from very different levels. Is it possible to achieve a full autonomous robot? Is it desirable? Absolutely autonomous robots are impossible to build. Robots are designed for achieving duties and this implies some kind of interaction with the world. Even human beings do not have this level of autonomy, they depend on others and their environment. In particular, *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.

As a result of the previous ideas, several levels of autonomy in robots can be found. A brief classification, from low to high autonomy, is listed below.

1. **Teleoperated robots**: they just execute actions commanded by an operator. Decisions are made just by the operator, so the robot is externally controlled by a human. For example, a bomb disposal robot is remotely controlled by the police.

2. **Robots with a minimal autonomy**: still there is an operator commanding the robot but it can make low level decisions, generally related to security, e.g. avoiding obstacles or interrupting its working cycle when a person is detected nearby the robot. For instance, in some surgical robots, the surgeon teleoperates the movements but the robot filters the motions proposed by the surgeon to keep only those which are compatible with the surgical plan.

3. ***Slave* robots**: the robot receives high level commands such as *go to a point* or *perform certain task*. The robot behaves as a "slave". The robot's goal is decided by a human. Space robots are clear examples of this kind. They receive high level commands, such as *sample the surface*, from the earth base and perform the task according to the circumstances.

4. ***Repetitive* robots**: these robots are endowed with predefined behaviors. It does not receive orders from people but its actions are fixed and known. Traditionally, these robots are employed to repetitive tasks. For example, industrial robots in car manufacturing have very specific tasks assigned and they do not change.

5. ***Script-based* robots**: several scripts are in charge of define the robot's behavior. Each script is a fix sequence of actions and the decision of which script to use depends on internal and external events. For example, a guide-robot in a museum has different behaviors which are predefined and they depend on the people around the robot, the level of energy, the exhibition, etc.

6. ***Self-goal-directed* robots**: there is an internal state related to physical parameters (e.g. battery level) as well as other more "cognitive" and "abstract" aspects, such as *happiness*. The internal state is related to the purpose of the robot which is able to determine its own goals. This thesis is framed in this level.

Figure 1.2 depicts robots from all these levels placed according to their autonomy and the importance of humans in their control. As shown, the higher level of autonomy, the less important the role of the human is. This could clearly rises legal risks in case of malfunction of the robot: who is responsible for that? [32].

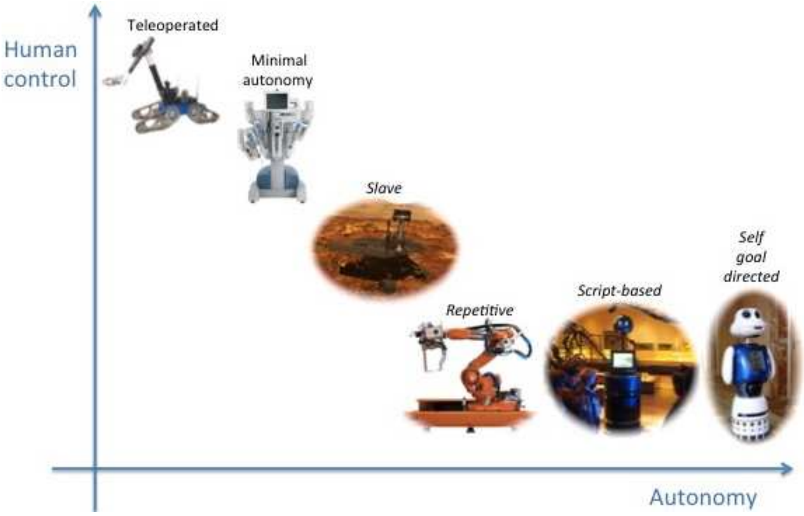


Figure 1.2: Levels of autonomy in robots in relation to the level of human control

Bellman [28] 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. In consequence, the level of autonomy relates to the amount of decisional mechanisms they are endowed with [33]. Moreover, several authors such as Arkin [34], Gadanho [24], Bellman [28], or Cañamero[29], in general, state that an autonomous agent must be self-sustained, which implies a decision making system. According to Hardy-Vallée [35], making choices is a reasoning process and rational decisions are made taking into account the probability and the outcomes of each action.

Moreover, some definitions of robots classify them as a special kind of agents and, being an agent entails making choices [35]. Consequently, robots have to be endowed with some kind of decision making mechanism. An autonomous robot acts on the basis of its own decisions [36] in order to fulfill its goals. Thus, it must know what action to execute in every situation. In the case that this robot does not have this knowledge, it must learn this relation between situations and actions. According to Mataric [37], 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 [24] states that learning is an important skill for an autonomous agent, since it gives the agent the plasticity needed for being independent.

As in other scientific fields, researchers try to imitate animals' mind and last investigations emulate animals' decision making. Accordingly, emotional and motivational models are suggested and some of them are oriented to maintain its internal equilibrium (homeostasis). As exposed in [38], humans' decision-making is not affected only by the possible outcomes, but also emotions play a main role. In view of it, several authors propose decision making systems based on motivations, drives, and emotions [39, 40, 41, 42, 43]. In fact, in recent years, several authors have argued that a truly biologically inspired and truly cognitive robotics would need to take into account homeostatic/emotional dynamics, i.e., the interplay between constitutive and interactive aspects of autonomy; for example, the need to keep essential system-internal variables within certain viability ranges [44]. In this work, this approach is followed, and the decision making system is based on drives, motivations, and emotions. This approach corresponds to the highest level of autonomy listed above.

This bio-inspired approach provides a mechanism to test and develop theories for understanding the underlying structures of the animal behaviors. Since even nowadays all *secrets* of the brain are still an ongoing problem, robots are an ideal platform for researching on different theories about minds, brains, or other areas, particularly when experimenting with living beings could be an ethical problem. Therefore, cognitive approaches in the development of robots can help to shed light on the ins and outs of the brain.

These ideas are not accepted by all researches. Bryson, on the contrary, defends that robots should be servants that people own [45]. She affirms that robots should be built, marketed and considered legally as slaves, not companion peers. This idea restricts the highest levels of autonomy to robots.

1.1.3 Learning

As said before, learning is a cognitive ability that provides the plasticity for adapting to new situations. Then, this is a key element for autonomy, mainly when dealing with high non-deterministic environments, like the real world.

Lorenz defined learning as the adaptive changes of behavior and this is, in fact, the reason why it exists in animals and humans [46]. Living beings react to sensory input coming from their environment. Some of these living beings change their reactions as time goes by: given the same input (sensorial perception), the reaction may be totally different. They are able to learn and update their reactions. Learning algorithms try to imitate this ability and to explain how and why the reactions change over time.

Most of the robots existing in unstructured environments require to be as autonomous as possible. This autonomy is related to the selection of actions during the robot's *live*. The robot self-governs its behavior through the policy that determines the next action to be executed at each moment. This policy can be acquired by two different manners:

1. The policy is assigned and the robot just follows this pre-designed policy.

2. The robot learns the best policy according to certain requisites.

In the first case, the policy is defined by others and it is imposed to the robot. In order to obtain an optimal policy, all situations and possibilities should be considered in the policy. However, in unpredictable environments, like real scenarios where the robots and people coexist, this is a tedious task. Sometimes it cannot be tackled. In this situations, the available decisions of the robot are pre-programmed and limited.

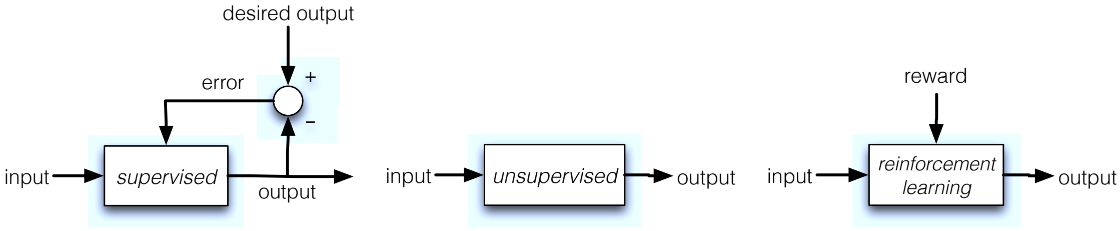
Learning does not restrict the possible decisions but provides a flexible mechanism to adapt the robot's behavior to new or unforeseeable events. Then, learning perfectly fits the needs of the exploration of uncharted "worlds", or situations.

Artificial Intelligence has explored many ways of learning since its beginning. Learning algorithms can be classified in three different main paradigms according to the kind of teaching signal, or feedback, received by the learner [47] (Figure 1.3). The first one is known as **supervised learning** where examples (input-output pairs) are provided by a well-informed external supervisor (Figure 1.3(a)); the problem is to obtain the function which links the provided inputs with the desired outputs (input-output mapping). For example, children at school learn the alphabet and they are told each time what sound correspond to what letter, so they can compare their responses to the correct one.

In contrast, **unsupervised learning** does not receive any feedback at all, so there is no way to evaluate a potential output (Figure 1.3(b)). It is based on the similarities and differences among inputs. Its goal is to fully categorize the input data. Typical unsupervised tasks are clustering where input data is classified. For example, our visual system is able to distinguish that humans are very different from elephants, which are very different from buildings; but these objects do not have to be labeled before they are clustered, even it is not necessary that our brain knows what a person, an elephant, and a house are in order to discriminate them.

The last kind of learning paradigm is the one called **reinforcement learning** (RL from now on). In this case, the teaching signal informs about the appropriateness of the response by means of the reward or reinforcement signal (Figure 1.3(c)). It looks for a state-action mapping which maximizes the reward. Unlike the supervised paradigm, the correct output is never presented in reinforcement learning. The reinforcement signal just informs about whether the output is correct or incorrect and how good or bad it is.

In relation to learning in robotics, Mataric in [37] 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 RL is one of the learning methods that has been most successfully implemented in robots. In fact, according to some authors, RL 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



(a) Supervised learning (b) Unsupervised learning (c) Reinforcement learning

Figure 1.3: Diagram of the main learning paradigms

obstacle avoidance task, the robot can obtain a positive reinforcement if it gets its goal, and negative if it crashes into some obstacle [48]. However, the reward can be oriented to more abstract events. This thesis presents an example of the latter.

1.2 The problem

Taking everything in mind, society is demanding robots with enough autonomy and cognitive abilities to *live* with humans attending and assisting them. These robots must be able to decide its actions according to external circumstances as well as internal ones.

In this thesis, the problem to face is how to extend the autonomy of a social robot in such a way that it can decide its own behaviors. Therefore, the robot is not considered a slave any more, but a partner which is able to make its own decisions. Human-robot interactions are accomplished in a peer to peer manner.

Since the robot is intended for human-robot interaction, it has to behave in a certain manner that it does not cause rejection of its human counterparts.

According to the kind of robots considered in this thesis (social robots), the problem is tackled from a bio-inspired perspective. Therefore, concepts and ideas coming from biological fields are included in an attempt to obtain biomimetic solutions.

1.3 Objectives

This thesis is the continuation of a previous line of investigation of the same research group, the RoboticsLab. María Malfaz, in her thesis, designed a model of decision making system based on drives, motivations, emotions, and self-learning tested on agents living in a virtual world [49].

Therefore, the main goal of this thesis is the extension of the robot's control architecture with a decision making system where drives, motivations, emotions, and self-learning are the essential elements. By means of this decision making system, the

role of several emotions in robots is studied. The previous model has been adapted and applied to a real robotic platform.

In order to achieve the final goal, other sub-goals come up:

- The robot has to be able to learn by itself from scratch the right behavior in each situation. Then, there is not a supervisor providing input-output pairs to the robot, so supervised learning is not an option. Furthermore, learning must be achieved in a reasonable amount of time and interacting with the real world. Consequently, reinforcement learning perfectly fulfills all the requisites previously presented.

- The robot will have to interact with objects in its environment. Then, it has to be endowed with enough mechanisms to properly perceive them and act over them. Objects have to be modeled considering their potential states and available capacities.

- To adapt the decision making model proposed in [49] to a real physical robot. The decision making module has to be incorporated into the current robot's control architecture with minimum modifications to the rest of the elements. Actually, these other elements will be utilized without altering them.

- The decision making system has to be designed in such a way that it is flexible enough to be applied to heterogeneous robots. Therefore, the system must be designed as flexible as possible so, it is easy to adapt to new robots and to extend to new requirements and configurations with minimum effort.

- The selection about what drives, motivations, and emotions are considered depends on the purpose the robot is intended for. Besides, the parameters of each one of these elements determines the final behavior of the robot. Then, these details are chief variables that must be carefully assigned.

- Each artificial emotion has to be independently analyzed for defining its right function. Later, these emotional functions are applied to the robot. In particular, the fear emotion in animals provides a reliable adaptive mechanism to deal with dangerous situations which threaten the survival. This function of fear will be applied to a robot.

- Human-beings will be considered as other "object" the robot can interact with. Their relationship has to be carefully studied. Moreover, since humans cannot be controlled, the effects of their actions must be managed.

- Finally, it must be analyzed if the inclusion of emotion-based functionalities result on a better performance of the robot or, in contrast, the benefit is not relevant or, even, unfavorable.

1.4 Overview of the contents

This thesis starts with an introduction to several biological concepts which will be referred in the rest of the book. Then, a review of other related works is presented. Some of these works have inspired the decision making system introduced right after. Following, the robotic platform, the integration of the decision making system, and its configuration used for the experiments are presented. After, the used learning mechanism is largely exposed. Next, the technical design of the adopted solution is detailed. And last, the experiments and their results are detailed. This thesis ends with several conclusions, comments, and future works.

These contents are explained in chapters which cover the following topics.

Chapter 2. This chapter settles the basic concepts that inspired the rest of the text. Several concepts, such as drives, external stimuli, or emotions, are introduced and their role in living beings is explained. Moreover, how humans make decisions is commented. In the last part, special attention is given to the emotions and their roles in humans and animals.

Chapter 3. A review of the most relevant works is presented in this chapter. First, the most important social robots, according to different purposes, are listed. Then, the review is centered on social robots which are controlled by architectures where motivations and emotions are essential components. The works that have inspired this dissertation are particularly detailed. At the end, a comparative analysis considering the main characteristics is presented.

Chapter 4. In this chapter the decision making system proposed by Malfaz, and followed in this thesis, is presented. This chapter shows how bio-inspired concepts are translated to "synthetic life": what a drive is, how a motivation is computed, and what the wellbeing is. Moreover, how the reinforcement learning fits in the decision making process is commented. Probably the most well-known reinforcement learning algorithm, Q-Learning, is detailed since a variation of it is implemented in the robot. Finally, the role of the three implemented artificial emotions (happiness, sadness, and fear) and their generation processes are explained.

Chapter 5. In this chapter the robot Maggie is presented. This is the robotic platform where the ideas of this thesis are implemented. First, a general description of its hardware is presented. Then, its control architecture is described. Last, the particular implementation of the decision making system is featured. That is, customizing the drives and the motivations, defining how the robot interacts with several items, and the consequences of its actions.

Chapter 6. This chapter explains the learning process implemented in the robot. Initially, the frame of the problem is introduced and the use of the Object Q-Learning algorithm is justified. Two modification specially designed to solve problems that appear when it is executed in real environments have been added. In order to clarify the ideas of the algorithm, several iterations of the learning process in different scenarios are evaluated.

Chapter 7. This chapter presents the technical design of the decision making system. A data base has been designed to provide an easy and expandable mechanism for adding new elements or easily modify the existing ones. All the elements of the decision making system have been modeled following an object oriented approach. Its design is shown here. Besides, the skills implemented for interacting with the objects are explained too.

Chapter 8. In this chapter, several tests prove the correct setup of the whole system. Initially, general arrangements for the experiments are stated. After, the correct operating of the decision making system is carefully checked. It shows how the theoretical concepts are properly working in the robot. Then, the learning algorithm and several improvements are justified. The results are compared with other traditional learning algorithm.

Chapter 9. Here, several experiments show the performance of the whole system in this chapter. These tests are performed by the robot Maggie in a real environment. First, the use of *fear* in a robot is carefully evaluated from several perspectives: how fear is appraised, how to react to fear, and the convenience, or not, of it. And second, the policy of behaviors learned by the robot are studied. These behaviors are the result of the *happiness* and *sadness* emotions as the reward signal.

Chapter 10. In this last chapter, the results are commented and the conclusions of this thesis are compiled. Moreover, the main contributions are presented and several future works are listed. Finally, some author's personal comments are included.

CHAPTER 2

Biological foundations

2.1 Introduction

The decision making system proposed in this dissertation has been inspired by mechanisms observed in nature. All animals are endowed with systems which are in charge of selecting the behaviors or actions to execute at each instant according to specific reasons.

In this chapter, a general view of some of these biological mechanisms as well as the reasons to behave in a particular way are exposed and explained. Moreover, similarities with the implemented system are highlighted.

In animals, behavior is considered as a manner of acting due to certain circumstances in order to achieve certain goals. The brain is responsible for all kind of behaviors arrangement, from seeking for food to falling in love. Certain brain neurons (electrically excitable cells) communicate with hundreds of thousands of cells around the whole body to orchestrate their functions and, as a consequence, behaviors arise. Then, behaviors can involve many organs (the heart, the liver, lungs, kidneys, etc.), and without them all behaviors would fail.

2.2 The origin of behavior

2.2.1 Innate vs learned

When animals make decisions, these can be innate or learned. Innate decisions are inherited and are species dependent. Some authors [50] consider them as instincts which are fundamental for the development of the individual. For instance, a baby animal already knows that it has to suckle from its mother. On the other hand, learned decisions consider the past experiences. As a result, when a decision has been learned, the behavior is guided by past experiences [51].

Then, decisions result from a "combination" of both, innate and learned, and they exist side-be-side.

2.2.2 Unconscious involuntary vs conscious voluntary

Veldhuis affirms that, such as in other high cognitive processes, decision making has a dual-processing perspective: conscious and unconscious [51]. Therefore, there are two levels of decision making which, somehow, are related:

- System 1: unconscious, fast, automatic, and high capacity decision (e.g. intuitive decisions). Prior knowledge is used to form a response. In this level, decision are involuntary.

- System 2: the highly conscious, slow, and deliberative decision (e.g. reflective decisions). It could happen that this system does not do anything (it does not have any effect), so the unconscious responses keep on working, or it inhibits the unconscious responses for developing a more conscious strategic thinking. This kind of responses are voluntary.

According to Veldhuis, in general, decision are made unconsciously but, when a novel event happens, the deliberative, conscious system takes over. This assumption implies that any deficit in the System 1 greatly affects our decision making capacity. Without the unconscious decision making system, all information has to be processed by the deliberative system. Due to its low capacity and slow processes, it results on very slow decision making and potentially loss of information.

Automatic processes are also referred as reactive processes by some researchers. Both terms, without distinction, can be used but author prefers the automatic term.

2.2.3 Homeostasis

Animals have to carefully control some internal conditions. For example, mammals live under tight conditions of body temperature and blood pressure, volume and composition. These variables must keep their values in a narrow range. The hypothalamus adjusts these levels in response to changes coming from the external environment. This regulatory process is called **homeostasis**: the maintenance of the body's internal environment within a narrow physiological range [52].

Homeostasis was discovered by Claude Bernard in the middle XIX century when he observed that the body variations had as an objective to give the stability back to the body.

According to the homeostatic approach, the human behavior is oriented to the maintenance of the internal equilibrium [53].

An example of this tendency towards internal stability can be easily observed on temperature regulation. Cells properly work at 37°C and variations of more than a few degrees are catastrophic. Precise cells belonging to the whole body perceive modifications on body temperature and response to this situation. On a extremely cold situation (e.g. you are naked on the North Pole), the brain sends commands to generate heat in the muscles (you shiver), to increase tissue metabolism, and to keep blood as far as possible of external cold surfaces of our skin in order to maintain internal warm (you turn blue). In contrast, if you are in a sauna, the brain activates cooling mechanisms: blood is moved to the external tissues where heat is radiated away (you turn red) and the skin is cooled by evaporation (you sweat).

In order to maintain the homeostatic balance, the whole body responds with voluntary and involuntary behaviors. All behaviors are orchestrated by the brain which reaches organs by means of the nervous system. The combination of the nervous system and the somatic motor system originates different behaviors.

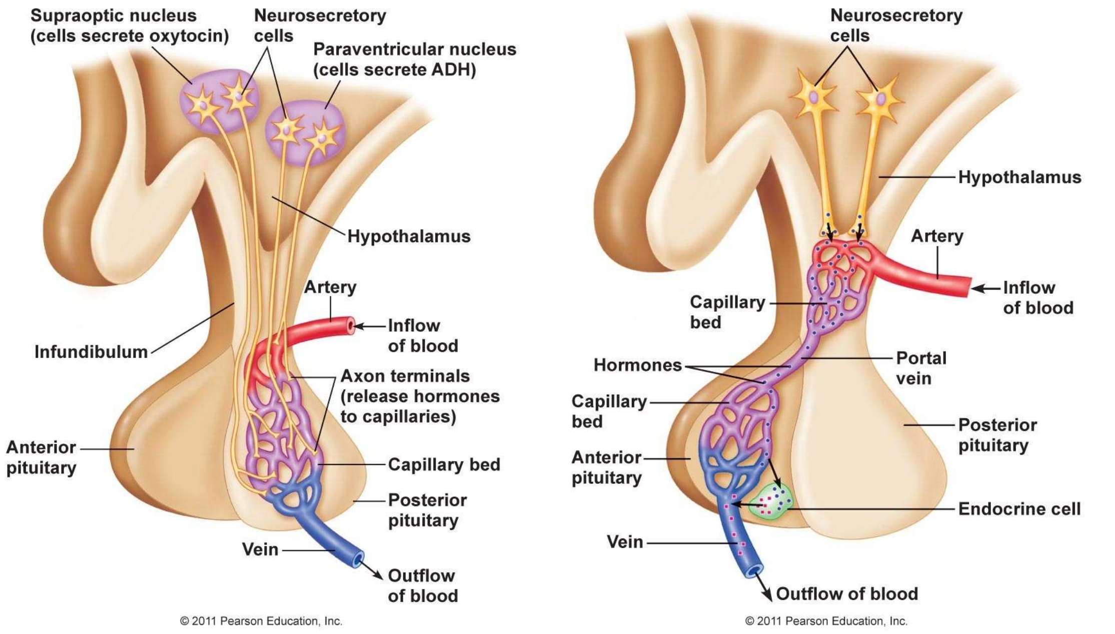
Involuntary behaviors

The behaviors which are not intentional, i.e. they are not voluntary, are based on the Veldhuis' System 1 and they depend on the nervous system. Three of its components are characterized by their great influence on the whole body: the secretory hypothalamus, the autonomic nervous system, and the diffuse modulatory systems. They differ on the areas they affect (from the brain to all over the body), the duration of their effects (from minutes to hours), and how they exert their influence.

The secretory **hypothalamus** is a structure of the brain and a component of the nervous system. It secrets chemical elements straight into the bloodstream and they alter activities on both body and brain parts. Figure 2.1 [2] presents the hypothalamus in the brain and how it is connected to the pituitary gland (where the hypothalamus exercises its influence). Despite of its insignificant mass (less than 1% of brain's mass), the hypothalamus' influence over the rest of the body is enormous.

Hypothalamus integrates bodily and emotional responses in accordance with the needs of the brain. Lesions in this part can result on disruptions of widely dispersed bodily function. Furthermore, the hypothalamus intervenes on common reflex where neural inputs and neural outputs are involved. Then, it is seen as the head ganglion of the autonomous nervous system which unconsciously controls internal organs. Experiments where certain areas of the hypothalamus are excited result on variations on heart rate, blood pressure, erection of hairs, and so forth.

[2] This figure appears in [52]



© 2011 Pearson Education, Inc.

© 2011 Pearson Education, Inc.

(a) The hypothalamus and its influence over the Posterior Pituitary

(b) The hypothalamus and its influence over the Anterior Pituitary

Figure 2.1: The Hypothalamus and the Pituitary gland

Some cells of the hypothalamus, the *neurosecretory neuros*, extend their axons to the pituitary (it is located just below the brain, Figure 2.1). The pituitary acts as the “speaker” the hypothalamus uses to communicate with the body. Neurosecretory neurons release substances (neurohormones) into capillaries running throw the pituitary (Figure 2.1(a)) or stimulate/inhibit the secretion of pituitary hormones (Figure 2.1(b)). These released hormones into the bloodstream reach organs whose functions are altered. The above reactions mentioned in the example of the homeostatic temperature regulation are provoked due to the activity of the hypothalamus.

The hormones secretion can be stimulated or inhibited due to several reasons. For example, the *oxytocin* hormone stimulates the ejection of milk from the mammary glands. A suckling baby stimulates the secretion of this hormone, even the cry or sight of a baby does. Sensory stimulus (somatic, auditory or visual) trigger the oxytocin release. This can be seen as **external stimuli** affecting the bodily reactions. Additionally, letdown of milk can be suppressed due to anxiety or other circumstances.

Another component of the nervous system responsible of involuntary behaviors is the **Autonomic Nervous System** (ANS), which commands the rest of the tissues and organs in the body. The ANS controls the physiological systems which are autonomous from the voluntary control [54]. For example, the smooth muscle system of digestion and blood flow. Therefore, ANS is constituted by a network of neurons covering the whole body which automatically acts, i.e. without voluntary control. The influence of the ANS over the whole body can be observed in Figure 2.2 [3]. It is anatomically separated from the voluntary motor system.

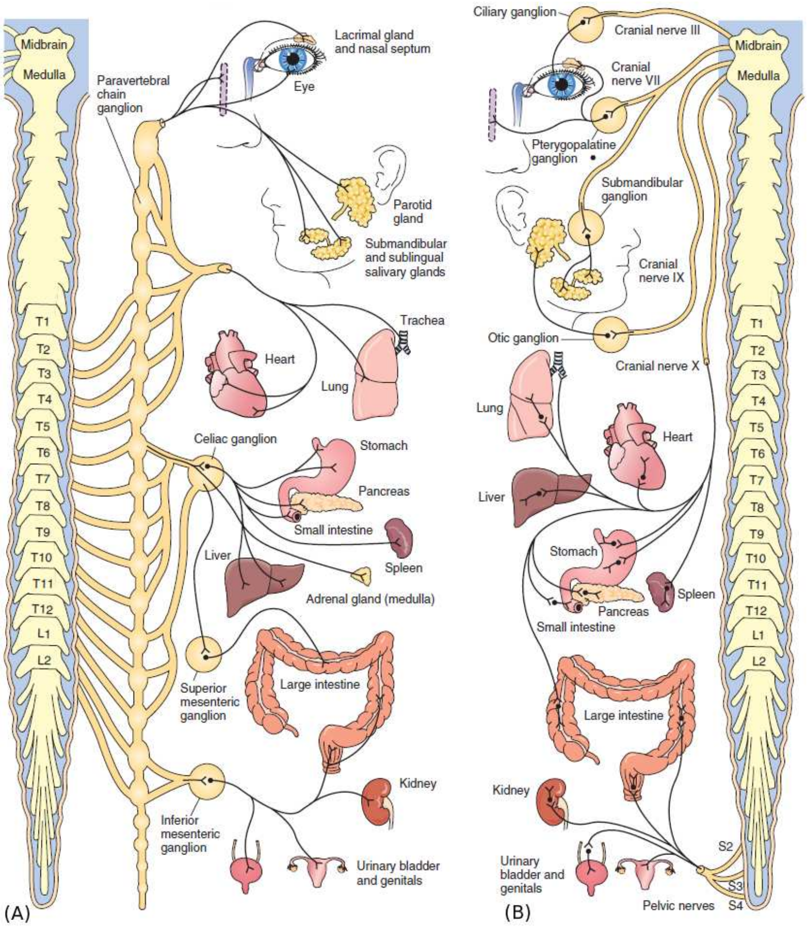


Figure 2.2: The sympathetic (A) and parasympathetic (B) divisions of the Autonomic Nervous System

The last component to mention is the **Diffuse Modulatory System**. It is entirely part of the Central Nervous System and it comprises several cell groups which extend their spatial reach to the entire brain and prolong their actions. These groups perform regulation functions that modulate the activity of a huge amount of neurons (each neuron may contact other 100000 neurons). Regulated neurons become more or less excitable, more or less

[3] This figure has been modified from its original version obtained from `http://pharmacology-notes-free.blogspot.com`

synchronously active, and so on. It is believed that they regulate the level of **arousal** and **mood**. Since this is a cutting-edge research field, the exact functions of this system on behavior are not absolutely clear and some ideas may be fuzzy [52].

Voluntary behaviors

Thus far, it has been mentioned how the brain influences on involuntary reactions. Nevertheless, the brain also generates intentional reactions. These are exhibited through the Somatic Motor System (SMS). The SMS is formed by the skeletal muscles and the nervous system that controls them. Its task is to innervate and command skeletal muscle fibers under voluntary control. Figure 2.3 [4] shows how the SMS is able to command a human arm: the muscles, which are activated by the signals coming from the CNS through axons, are in charge of moving the skeletal and, then, the behavior is generated.

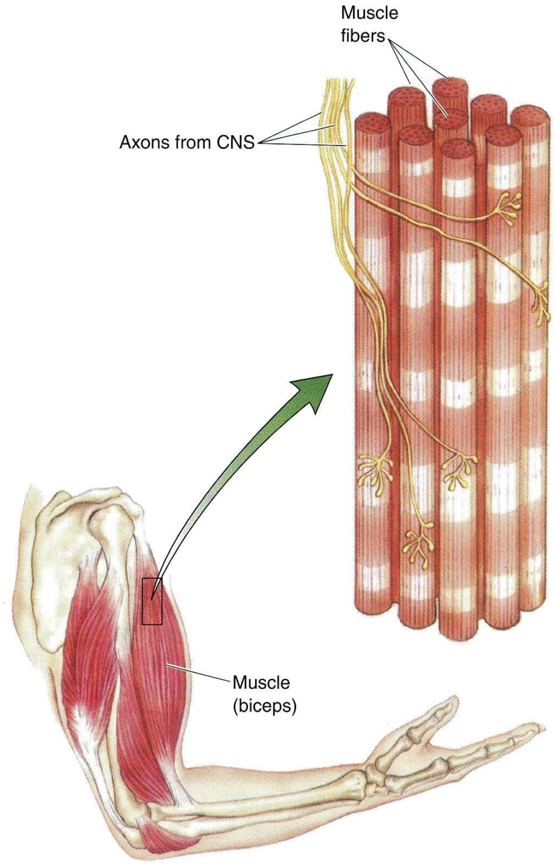


Figure 2.3: Part of the Somatic Motor System involved in the movement of a human arm

[4] This figure appears in [52]

All the organs and tissues are highly coordinated by neurons in the brain. Both systems, SMS and ANS, have upper motor neurons in the brain that send commands to lower motor neurons which, actually, act over the target structures.

In short, each of these elements has different specific functions but, generally speaking, it can be said that they all maintain brain homeostasis: they regulate different processes within a certain physiological range [52].

2.3 Motivated behavior

The key question at this point is why behavior occurs. As previously presented, behaviors involve motor responses and these can be unconscious reflexes (e.g. secretion of gastric juices before you eat) or self-conscious movements (e.g. approaching the fridge because you are starving). Intentional movements are originated (motivated) to satisfy some sort of need [52]. The motivation to satisfy a need can be abstract (to be happy) or totally real (to drink water because you are thirsty after running a marathon). These needs are due to a deviation in an homeostatic variable and they are referred as **drives**.

2.3.1 The Hull's drive-reduction theory

Clark Hull postulated in 1943 his drive-reduction theory [55]. This is one of the oldest theories about drives. Hull suggested that privation induces an aversion state in the organism, which is termed drive. According to his theory, the drives increase the general excitation level of an animal and they are considered as properties of deficit states which motivate behavior.

He stated that all the behaviors happen as the result of physiological needs, the drives. According to his theory, the reduction of drives is the primary force behind motivation [56]. He based his theory around the concept of homeostasis, i.e. the body tends to maintain certain internal balance and actively works for it. Behavior is one of the resources the body has for achieving it. Considering this approach, Hull postulated that all motivations come up due to biological needs, which Hull referred as drives (thirsty, hunger, warmth,etc.). Thus, a drive produces an unpleasant state that has to be reduced by means of the corresponding behavior (e.g. drink when we are thirsty or close the windows when we are cold).

This reduction of drives serves as a reinforcement for that behavior. In the future, when the same need arises, the reinforced behavior will be more likely repeated. In other words, when a stimulus and a response provoke a reduction in the need, the probability that the same stimulus causes the same response increases [57].

However, many years later the Hull's theory started to fall out of fashion due to many criticisms [58]. First, Hull's theory does not consider secondary reinforces. Primary reinforcers satisfy survival needs such as food, shelter, or safety. Secondary reinforcers are those that can be used to obtained primary reinforces. Some examples could be money, praises, or grades in school. Moreover, this drive-reduction theory does not explain the behaviors that are not related with biological needs and therefore do not reduce drives.Why do people eat when they are not hungry? Why do people sky dive? This theory does not answer these questions.

2.3.2 Motivations

Later, other researchers started to tackle the not explained questions in the Hull's drive-reduction theory. In [50], motivation is presented as an inferred internal state postulated to explain variability on behavioral responses. Motivational states represent urges or impulses that impel animals into action. Initially, motivations were linked to bodily needs such as energy or temperature regulation (classical homeostatic drives). But other non-physiological needs are well-accepted as motivations too, e.g. curiosity or sex. However, all these needs are referred as drives because they involve arousal and satiation. The concept of *drive* is postulated in order to explain why observable stimuli in external environment are not sufficient to predict behaviors. For example, sometimes food can stimulate feeding, but others, it results on indifference or even rejection. E.g. when you walk a street and see chocolate, it can provoke the "need" to eat chocolate. In contrast, after a big meal, the perception of more food activates a denial reaction.

Many drive theories 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 [53]. In other words, the motivational state is a tendency to correct the error (the drive) through the execution of behaviors.

The motivations can be seen as a driving force on behaviors. However, just motivation does not guarantee a behavior but it modulates the behavior and affects its probability to happen. Beside, several motivations may interfere each other, for example the need of food versus the need of sleeping.

The word motivation derives from the Latin word *motus* and indicates the dynamic root of the behavior, that means those internal, more than external, factors that urge to action [59]. Sometimes, motivational states can be explained as a compendium of internal and external stimuli. Hence, **motivation** can be presented as a complex reflex under the control of multiple stimuli, some of them internal [50]. Hull [60] also proposed the idea that motivation is determined by two factors. The first factor is the drive. The second one is the incentive, that is, the presence of an **external stimulus** that predicts the future reduction of the need. For example, the presence of food constitutes an incentive for a hungry animal.

The already presented hypothalamus is involved on the homeostasis process and motivated behavior. Recalling, homeostasis refers to the processes that maintain the internal variables of the body (temperature, fluid balance, energy balance, ...) within a narrow physiological range. The hypothalamic regulation of homeostasis starts when a regulated parameter has gone out of the desired range. Sensory neurons watch the parameter and communicate with hypothalamic neurons which detect the deviations from the optimal range. Then, these neurons orchestrate an integrated response to bring the variable back to the normal values. Generally speaking, this responses have three components [52]:

- Humoral response: the release of pituitary hormones are stimulated/inhibited by hypothalamus neurons.
- Visceromotor response: hypothalamic neurons act over the ANS and the corresponding tissues and organs accurately respond.
- Somatic motor response: hypothalamic neurons, acting on the somatic motor system, provoke a somatic motor behavior.

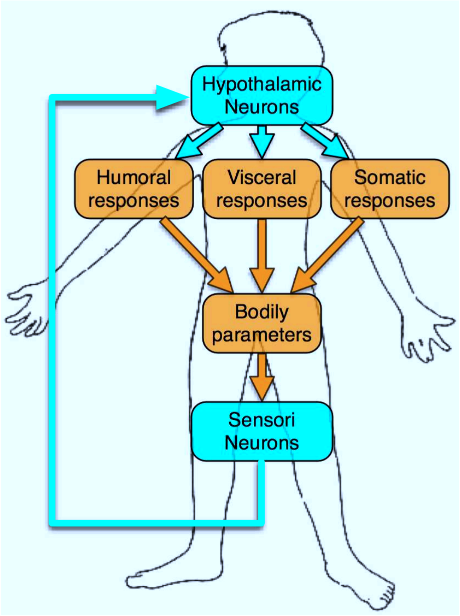


Figure 2.4: Hypothalamus responses to homeostatic body control

The following example will clarify the ideas previously introduced. When a person is cold, dehydrated, and depleted of energy, the proper responses automatically come through. This person shivers, his blood is moved away from the body surface, urine production is inhibited, body fat reserves are mobilized, and so on. However, the most effective and fastest way to correct the disruptions is to look for a warm place, to drink water and to eat. These are **motivated behaviors** generated by the SMS and incited to occur by the hypothalamus [52].

Hypothalamus and related structures received information from the internal environment and they directly act over the internal environment (if you are cold, your body temperature is directly kept constant by peripheral vasoconstriction). Other hypothalamic neurons are in charge of operating indirectly over the internal environment, by means of the SMS acting in the external environment (if you are cold, you can turn the heat on). Both indirect and direct homeostasis can work in parallel.

Besides, Veldhuis's systems (Section 2.1) can be observed in the previous example; vasoconstriction is a unconscious and involuntary reaction which can be placed at System 1; turning the heat on corresponds to the System 2 where conscious, voluntary actions are made.

The intensity of a motivation depends on several factors. Considering hunger the motivation to eat, it depends on how much you ate the last time, what kind of food, and how long it has been since then. Moreover, the motivation to keep on eating counts on how much and what kind of food has already been ingested. After we eat and the digestive process has begun, the need of energy is inhibited due to satiety signals. These satiety signals slowly dissipate until the need to eat again takes over. This interrelationship can be observed in Figure 2.5 [52]: just after eating, satiety signal soars; then, it slowly vanishes until the next ingestion of food when it rises again. In general terms, drives, in the sense of needs or deficiencies, lead the regulatory process of motivations. Drives vary according to several signals and parameters. However, the presence of incentives, external stimuli, can alter the course of motivations and/or drives.

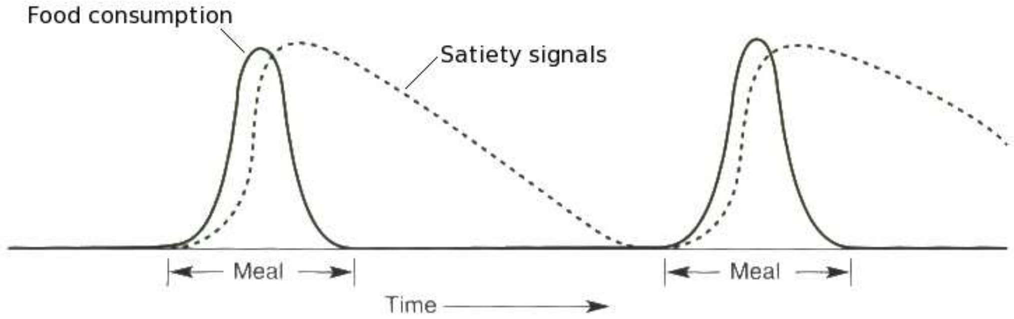


Figure 2.5: Feeding behavior and satiety signal

Cognitive aspects of motivations

After understanding the physiological aspects of the motivation of behaviors (especially those that are basic to survival), it seems that humans are ruled by hormones which secretion is activated by neurons all over the body. However, researchers clarify that one of the main advantages of human evolution is our capacity to exert cognitive control over our more primitive instincts.

Motivational behaviors are not only attached to physiological needs. For example, curiosity does not appear to be commanded by any physiological shortage. Particularly in humans, learned behaviors and pleasant feelings can prevail against bodily signals. This is the case when a person feels the need of going to the toilet but he is attending an important meeting and he cannot leave the room.

In psychological terms, according to [52], there are two points of view about motivated behaviors:

Hedonic People exhibit a behavior because they *like* it, it feels good so people do it (e.g. the smell, taste, and sight of food, and the act of eating itself are pleasant). Pleasure serves as an hedonic reward.

Drive reduction People *need* to behave in such a certain way in order to satisfy a drive (e.g. animals eat because they are hungry and *want* food).

Both approaches seem to be complementary (we drink what we like) but, apparently, “liking” and “wanting” are controlled by different circuits in the brain [52].

Other researchers [50] identify three factors as motivated behaviors regulators: ecological requirements of the organism, anticipatory mechanisms, and hedonic factors (pleasure).

Ecological constraints Behavior patterns have been shaped by evolutionary selection. Ecological context is analyzed by cost-benefit functions. Feeding behavior includes the cost of searching and procuring food, and the benefits of the energy obtained from the nutrient intake.

Anticipatory mechanisms Clock mechanisms activate physiological behavioral responses before the need or the deficit in the tissues occurs. Therefore homeostasis often anticipates deficits.

Hedonic factors Pleasure is an undoubtedly factor in the control of motivated behavior of animals. Frequently, humans give up some need in order to obtain pleasure by satisfying others. For example, people go on a diet because they want to look more attractive. It gives the idea that pleasure mechanisms are concerned with reward and reinforcement on learned behavior.

The ecological constrains and the anticipatory mechanisms [50], and the drive reduction [52], somehow, all are related to physiological needs. The hedonic factor in motivated behavior is clearly identified in both approaches.

Since pleasure is an evident element on motivated behavior, researchers have studied how it is evoked. Olds [61] discovered pleasure areas on animal's brain. Later, Deutsch and Howard [62] found that stimuli of pleasure areas on the brain originate reinforcement independently of the drive state of the animal. In contrast, regular stimuli just function as reward in particular states (food is considered as reward just in hungry animals). Successive studies have shown that pleasure areas in the brain are involved on initiating some complex behaviors such as feeding and drinking. Apparently hypothalamus is one of the areas that produces reward and several transmitters seem to take part.

2.4 Emotions

Thus far, emotions have not been mentioned. However, emotions play a key role in the behavior exhibited by people and animals. Emotions are essential in our daily live. They make us going high and low in all our experiences. Emotions are not easy to study and just their behavioral manifestation can be certainly observed. Besides, emotions are not exclusive from humans [63], it is proved that animals also are endowed with emotional states [7]. Actually, Charles Darwin (1809-1882) studied emotions in humans and animals just by observation of the emotional expressions during his trips to exotic places. His evolutionary theories suggested that emotions have evolved due to their efficacy for adaptation and for communicating the behavioral intention, and, also, their role in social interaction [64]. Currently, it is widely accepted that humans and mammals share some emotional brain regions [54].

2.4.1 The role of emotions

Emotions are versatile mechanisms which are involved in many functions. According to Rumbell [65], emotions influence the attention, alter the likelihood of behavioral responses, activate associative memories, arrange rapid responses, influence learning, aid social behavior, and improve communication.

In this thesis, the attention is directed to the influence of emotions on the decision making process and the learning process. Cañamero states that emotions and motivations play a main role in autonomy and adaptation in biological systems [66].

According to the popular belief, *emotional* reactions are undesired and not appropriated. In contrast, rational reaction are more appreciated. However, both reactions are required and have different functions. Reason and emotions separately can make wrong decisions. A reasonable action can be rated as inadequate (to kill a person in order to save many), as well as a decision made considering excessive reasonable arguments (rejection to travel because of the afraid of flying). Therefore, people require a correct balance between reason and emotions in order to make right decisions. This process is shaped along the individual's life [67]. According to Gordillo [67], the right decision is "the most beneficial one for the individual".

Emotions influence decision making in two ways: expected emotions and immediate emotions. When an individual makes a decision, he likely expects certain outcome which can be related to an emotion (the expected emotion). Usually positive emotional results are preferred to negative ones. Besides, the emotional state when the individual makes a decision (the immediate emotion) influences this decision. In general, happy people overestimate their decisions and depressed people underestimate the outcomes. Considering this, emotions can lead to wrong decisions. Many other aspects influence decision making: gender, development, culture, etc.

As said, an individual can seek for a particular emotion through its actions. Emotions therefore can work as important reinforcements for certain behaviors or actions [4]. For this reason, emotions play an important role in learning. Then, a behavior can be pursued, among other reasons, in order to experience the emotions associated with the outcome of that behavior [68].

According to Castelfranchi [68], emotions activate goals and plans that are functional for re-establishing or preserving the well-being, considering the events that produce them. Consequently, emotions have a conative component, that is a tendency towards action. This is referred as the **motivational component of emotions**. It is worth mentioning that emotions cannot be reduced to motivations, or vice verse. Emotions have more functions than the motivational and, on the other hand, there are motivations which are not related to emotions. In short, to feel or not to feel an emotion can, by itself, become a goal to the individual [69].

LeDoux claims that emotional behaviors represents different functions for solving problems and with different brain mechanism. Therefore, he proposes to study emotions as independent functional units [63].

2.4.2 What is an emotion?

Although *emotion* is a word commonly used, there is not a clear definition of what an emotion is. For example, Ortony defines emotions as *"valence reactions to events, agents, or objects, with their particular nature being determined by the way in which the eliciting situation is construed"* [70]. Moreover, for Frijda, emotions are *"responses to events that are important to the individual, and these responses follow certain general rules or laws"* [71]. He strengths the conative role of emotions affirming that emotions are *"motivational states that underlie emotional behaviors"* [72]. In a definition given by Castelfranchi, he adds a cognitive component to emotions, so he states that human emotions are complex and rich mental states, not simple reactive mechanisms [68]. Rosis remarks the adaptive role of emotions: *"Emotions are biologically adaptive mechanisms that result from evaluation of one's own relationship with the environment"* [69]. Despite the wide variety of definitions, it seems that all of them, in one way or another, consider that emotion involves certain *conditions* that elicit emotions and, as a result, some reaction is provoked. Further on in this chapter, these two aspects are covered.

In an effort to not use definitions literally, in this thesis, emotions follow a more practical approach. Then, emotions have a dual nature: one is the external expression of emotions and the other is the internal experience of emotions. Both must be differentiated. When a stimuli causes the emergence of an emotional response, its effect is twofold: first, it provokes non-conscious internal reactions, so the internal state is altered and the organism is ready to fight, fly, sex or other adaptive behaviors; second, the behavior is modulated by cerebral structures during interaction with the external environment [50]. The external environment is richer on stimuli than the internal one, so it is much complex.

The behavioral signs of emotion are controlled by the somatic motor system, the autonomous nervous system, and the secretory hypothalamus. The hypothalamus also orchestrates the internal responses. The clue is how sensory input or internal signals lead to a particular emotion.

Hypothalamus can be interpreted as a coordinating center that integrates various inputs to ensure a well-organized, coherent, and appropriated set of autonomic and somatic responses. These responses were observed as similar to emotional behavior, then, it is suggested that the hypothalamus manages the emotional expression. Moreover, it seems that hypothalamus articulates motor and endocrine responses which produce emotional behavior [50].

2.4.3 Theories about emotions

About emotions, there are still not a unique theory about them. One of the first well-formed theories is the James-Lange Theory (1884). It states that emotions are experienced as a consequence of physiological changes in the body. The sensory system reacts to the changes evoked by the brain, and it is this sensation that constitute the emotion. That is, the physiological changes are the emotion. Therefore, if the changes are removed, so the emotion does [52].

In 1927, the Cannon-Bard Theory proposed that emotional experience can be independent of emotional expression. That is to say, emotions can be experienced even if physiological changes cannot be sensed. This theory states that the thalamus plays a special role on emotions which are produced when signals reach the thalamus [52].

An example will clarify the differences between both theories. According to James and Lange, when you see a rattlesnake you express fear (you shiver, your heart rate speeds up, and so on), and, as a consequence, you are terrified (you experience fear). In Cannon's theory, first, your thalamus is properly activated to experience fear and then the physiological signs of fear occur.

Several works after these theories have demonstrated that both theories have strengths and weaknesses. For example, perhaps some emotions depend on behavioral manifestations: such as smiling (expression of happiness) in order to feel happy; and experiencing other does not: hope does not have to be linked to any expression. About James-Lange theory, even if emotion is closely related to physiological states, emotions can be felt in the absence of evidence physiological signs. E.g. a person suffering a high-level paralysis can experience happiness. But some strong emotions are close related to physiological states and it is not clear what causes what [52].

2.4.4 Emotion systems

Currently, it seems that different emotions involve different brain circuits, despite of same brain areas could be common. In fact, Dalgleish [73] explains how other theorists, inspired by the prototypical work of Darwin, have proposed that a small set of discrete emotions are underpinned by relatively separable neural system in the brain [74, 75]. Then, as Kandel et all states in [50], distinct emotions are located at different parts of the brain. Then, when stimulating these areas in human's and experimental animals' brains, different emotions elicits.

These emotional theories do not mention about the possibility of experience several emotions concurrently. Actually, Rosis [69] states that several emotions can be experienced by the same individual at the same time, but with different intensity each one due to many circumstances (the kind of goal, the novelty, etc.).

Therefore, as reported by Bear et all [52], the definition of an emotion system is controversial. Due to the broad spectrum of emotions, it is no clear that only one system is controlling emotions, rather than several systems. Moreover, some elements involved in emotion also take part on other functions; then, there is not one-to-one relationship between structure and function. This fact reflects that researchers are just beginning to understand how emotions are experienced and expressed. Therefore, some questions are still pending.

Then, as already mentioned, instead of thinking of one emotion system, some authors convey the impression that several separated emotional systems exist, each one relates to an emotion and considers its stimuli and reactions.

However, from a physiological point of view, there seems to be several common brain parts that support the emotional life [54]. This is referred as the **limbic system** (Figure 2.6) and its elements are hypothetically responsible for the sensation and expression of emotions. The concept of *limbic system* is controversial because there is not an universal agreement about its components. Even some scientists defend the suppression of the term [50].

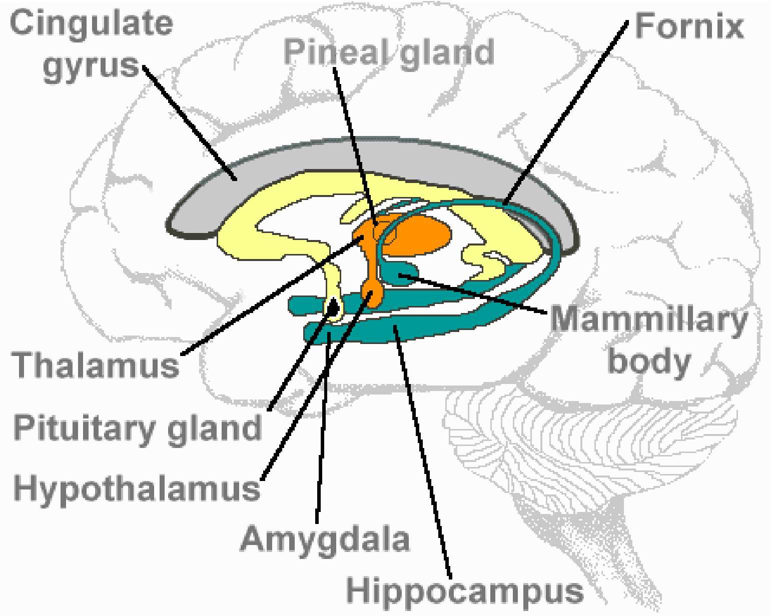


Figure 2.6: The Limbic System

The limbic system is mainly composed of the cortex (the outer layer of the brain), the hypothalamus, and the amygdala. The **cortex** is the main actor in the experience of emotions and the **hypothalamus** rules the expression of emotions. Both structures influence each other so they are bidirectionally communicated. This bidirectional link makes James-Lange and Cannon-Board theories compatible. Then, this limbic system enables animals to express and experience emotions.

The **amygdala** is other element of the limbic system which plays an important role on emotions. It conveys high cognitive information to the hypothalamic structures. The amygdala receives inputs from cortical structures and the thalamus. LeDoux [76] suggested that this direct thalamic input mediates on short-latency primitive emotional responses and prepare the amygdala for the reception of more sophisticated information from higher centers, such as the prefrontal cortex. The output of the amygdala is connected to cortical structures and results in a conscious emotional experience. More details about the amygdala are presented in Section 2.4.7 in relation to the fear emotion (it seems that the amygdala is closely linked to this emotion).

One of the most famous studies about the influence of the amygdala and cortical structures on emotions dates from 19th century. On 1848 Phineas Gage suffered a terrible industrial accident: an iron rod was sent into Phineas' head due to an unfortunately explosion. The rod destroyed much of his brain's left frontal lobe and a considerable portion of his skull. Miraculously, just one month after the accident, Phineas was walking again and returned to his job. He looked like before the accident but something has changed:

his personality was totally altered. Before the accident, Phineas was considered as an efficient, capable, well-balanced mind, shrewd, smart business, and persistent man. After the accident he was described as a fitful, irreverent, blasphemous, impatient, obstinate, capricious and vacillating. Despite of the lack of psychological test, it appears that Phineas' personality was dramatically changed far more than his intelligence.

In 1994, Hanna and Damasio [77] made new studies on Gage's skull using modern technics. Figure 2.7 shows the trajectory of the rod into Phineas' head. The iron rod severely damaged the cerebral cortex in both hemispheres, particularly the frontal lobes. As result of the damages, Phineas became to act as an irritable kid suffering of strong emotions. The significant increase on emotional behavior proposes that cerebral cortex plays a key role in regulating emotions.

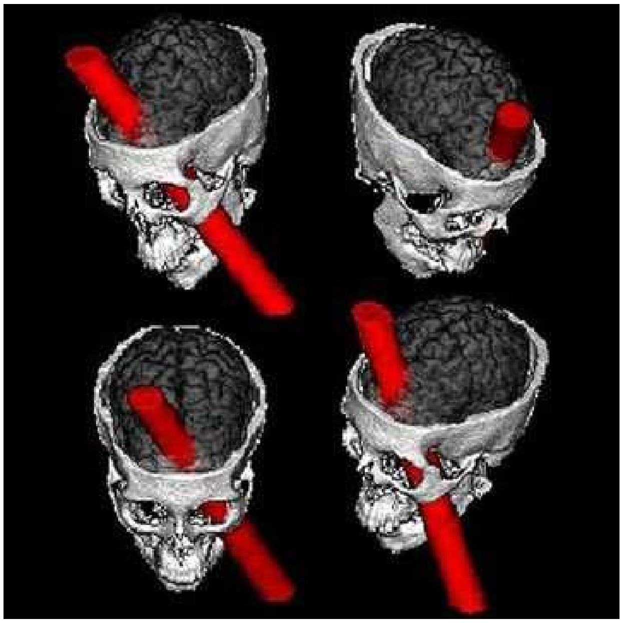


Figure 2.7: Reconstruction of Phineas' skull and the iron rod

2.4.5 The Appraisal Theory

Especially interesting is the study of the processes involved in the generation of emotions. The mechanism in charge of evaluating the current internal and external situations in terms of affective state or emotion are referred as appraisal [78]. Appraisal theories seek for explanations of these evaluations that lead to evoke one emotion over another.

The term *appraisal* was coined in 1960 by Magda Arnold [79] who stated that the appraisal starts the emotional responses. The appraisal theory is the claim that emotions are elicited by evaluations of situations [80]. According to this theory, it is the interpretations of situations, rather than the situations themselves, that cause emotions. Consequently, emotions are differentiated by appraisals, i.e. each emotion is elicited by a unique pattern of appraisal.

Most of the researches link the appraisal of certain event or situation to the motivation of the individual [81]. Thus, the emotion resulting from an appraisal depends of the relevance of the event to a motivation [82]. For instance, a bear can elicit fear if we are picnicking; however the same bear can result on excitement or happiness if we are hunting.

Because appraisals intervene between situations and emotions, different individuals who appraise the same situation in significantly different ways will experience different emotions; and even a given individual who appraises the same situation in significantly different ways at different times will experience different emotions. A good example would be a football match, the same situation, the result of the game, will produce different emotions depending on your team [83]. Another example can be observed on a student doing an exam: if he has studied hard all the semester, he feels confident and relaxed; in contrast, if he has not studied enough, he experiences fear and gets nervous.

At some point, this theory could seem controversial. A polemic example could be the dead of a person: a priori, the passing of a person can be clearly tagged as a sad event. However, sometimes, this is not true. Considering the relatives of people murdered by a psychopath, sometimes, they experience happiness or relief when the psycho is dead. The different outcomes of the appraisal result from the individual conditions considered during the personal appraisal process.

Moreover, Scherer [82] affirms that appraisals are part of a circular process where they are "cause of emotions, components of emotions, and consequence of emotions". Therefore, it seems that appraisals are more complex than simple linear relations between appraisals and emotions.

Following this theory, a situation cannot be tagged with an emotional value in advance, it is the interpretation each person makes of that situation which gives that individual evaluation.

In order to understand the appraisal mechanisms of emotions, that is, how they emerge in our brain, it is proposed that, as LeDoux in [63], since emotions are produced by different brain networks, they must be studied one by one.

The appraisal of emotional events (the releasers) can be classified considering how they are acquired or its origin. Some releasers are innate or inherited. For instance, the presence of a cat is evaluated as dangerous by mice, and this "knowledge" has been inherited. Consequently, cats are the releasers of fear in mice. This implies that they are more species specific than those acquired during life by experience.

Other emotional releasers are learned. For example, cats do not like to visit the veterinarian because they usually hurt them. As a result, they associate the presence of the veterinarian with a harmful situation. Then, the perception of the veterinarian becomes a releaser of fear in cats.

As seen in Section 2.1, dual-process theories distinguish between reacting (fast and intuitive) and reasoning (slow and controlled) as a basis for human decision-making. This dual-approach is observed also in relation to the generation of emotions. Based on how the appraisal is performed, both approaches, automatic and deliberative, are considered. Castelfranchi [68] proposes to distinguish two kinds of "evaluation": cognitive evaluation (or just evaluation) and appraisal.

- *Appraisal*: a non-rational appraisal based on associative learning and memory, but it is not based on justifiable reasons. It is automatic, implicit, and intuitive orientation towards what is good and bad for the organism. Then, appraisal is an automatic association of an affective internal state (emotion) to the appraised stimulus or representation. This involves System 1.

- *Evaluation*: a reason-based evaluation that can be discussed, explained, and argumented. It is a the cognitive judgments relative to what is good or bad for someone (and why). It is related to System 2.

In fact, LeDoux relates this unconscious appraisal to emotion, and conscious evaluation to *feelings* [63]. On the other hand, Sloman [84] and Bechara [85] differentiate between primary emotions which have a reactive or automatic basis and secondary emotions that require a deliberative process to initiate them.

Taking again the fear emotion as an example, this dualistic approach is easily observed. In some cases, fear is automatically elicited (mice are afraid of cats), but in others fear emerges due to a reasoning process (e.g. *due to the actual economic circumstances, I am afraid of loosing my job*). Moreover, this deliberative process affecting fear works as feedback to the intensity of fear (e.g. *if I loose my job, I will not get money, and then I will not be able to feed my family, and finally we will all die*).

These two classifications of the generation of emotions and the related examples about *fear* are exposed in Table 2.1. Each cell contains an example considering how the process has been acquired and how it is performed. Yellow cells correspond with the kind of fear implemented in this thesis. Red cells are those combinations which are impossible: something innate has been inherited so it is a species feature; in contrast, deliberation is a particular process of each individual; in consequence, deliberative-innate processes are not possible.

2.4.6 Emotional reactions

Often, the emotional behavior is considered as a consequence of emotions, rather than a part of them, because other factors than emotions contribute to their generation [82]. However, without getting into the details of this discussion, each emotion alters the behavior and causes an emotional reaction.

Table 2.1: Examples of different generations of fear

		How is it acquired?	
		Innate	**Learned**
How is it performed?	**Appraisal (automatic)**	Mice experience fear when they perceive the presence of a cat	Cats have fear when they perceive the presence of the veterinarian
	Evaluation (deliberative)	🟥	The global economic crisis

These reactions can be classified in a similar approach to the appraisal: considering how these emotional reactions were acquired (innate vs learned) and how the reactions are performed (automatic vs deliberative).

In order to clarify these ideas, several examples are presented in Table 2.2. This table presents examples of different reactions to fear based on the examples of appraisal aforementioned in Table 2.1. Both tables follow the same arrangement.

Mice inherently know that, when they experience fear in front of a cat, they must escape from the cat. However, cats have learned, through several experiences, that they must run away when a veterinarian is present. These two examples are automatically executed, so these reactions are not the result of a deliberative process. However, when a person is frighten because of the uncertain stability of his job, this person performs a reasoning process (e.g. *if I don't want to loose my job, I have to increase my production, so I have to work more hours*) where all possibilities are considered and, as a result, it decides to work harder.

Table 2.2: Examples of different reactions to fear

		How is it acquired?	
		Innate	**Learned**
How is it performed?	**Automatic**	Mice escape from cats	Cats run away when they see the veterinarian
	Deliberative	🟥	I must work harder in order to keep my job position

Analogously to the appraisal, innate-deliberative reactions are not possible (red cells): innate reactions have a species component and deliberative reactions are the result of an individual cognitive process.

Automatic processes, both for appraisal as well as for reactions, can be observed in animals. These are required for survival purposes. Innate fears are considered by some researches as instincts which provide a key survival mechanism. Actually, animals without these instincts should have difficulties to reach adulthood. However, deliberative processes are specific of humans beings and this is one of our main characteristics.

2.4.7 Fear, anxiety and the amygdala

Fear is an emotion particularly studied in this dissertation (mainly the automatic aspects of fear). Therefore, this section presents a deep analysis of fear.

In animals, fear is associated to anxiety as a response of threatening situations [50]. Whenever an individual is afraid, he becomes anxious. Its symptoms are: arousal, restlessness, overreaction, dry mouth, desire to escape, avoidance behavior, sweat, heart-racing, high blood pressure, etc. About the utility of fear, it contributes to behave in a proper way when a difficult situation is being faced. However, it also can be detrimental as it is presented at the end of this section.

LeDoux [21] states that the function of the emotion of fear is to detect danger and to produce reactions which increase the probabilities of survival in a dangerous situation. In other words, it is a defense mechanism. Therefore, the appraisal mechanism of fear is related to the evaluation of situations as dangerous. Then, the fear emotion is involved with natural enemy avoidance behaviors and areas where previously suffered fear experience [86]. Accordingly to Darwin's theory of evolution [64], fear has evolved as a mechanism that enhanced chances of survival.

Most of the previous listed symptoms of fear are physiological reactions provoked by fear. They are orchestrated by the hypothalamus even before a behavioral reaction appears. Anxiety reactions are controlled by the ANS and they virtually affect all parts in the body. Moreover, the level of anxiety and the intensity of bodily responses are proportional to the amount of perceived danger [52]: the more danger, the more anxiety, and, then, the more fear.

From a neurological perspective, it has to be explained how the incoming information into the brain causes behavioral and physiological reactions related to fear and anxiety. Several studies propose that the **amygdala** of cerebral limbic system processes fear emotion and plays a key role for survival of animals [86].

The amygdala is a structure placed at the pole of the temporal lobe just below the cortex (Figure 2.8). The amygdala is one of the most important brain regions for emotions, with a key role in processing social signals of emotions (particularly involving fear), in emotional conditioning and in the consolidation of emotional memories [73].

Information from all of the sensory systems feeds into the amygdala where the information is integrated. It is connected to the hypothalamus. The amygdala alters the ANS through the hypothalamus and evokes behavioral reactions via the SMS [52].

It is assumed that amygdala is the brain structure in charge of fear regulation and responses: projections from the amygdala to the brainstem contribute to the expression of fear, and the experience of fear, and other cognitive aspects of emotional processing, involve projections from the amygdala to the cortex [87].

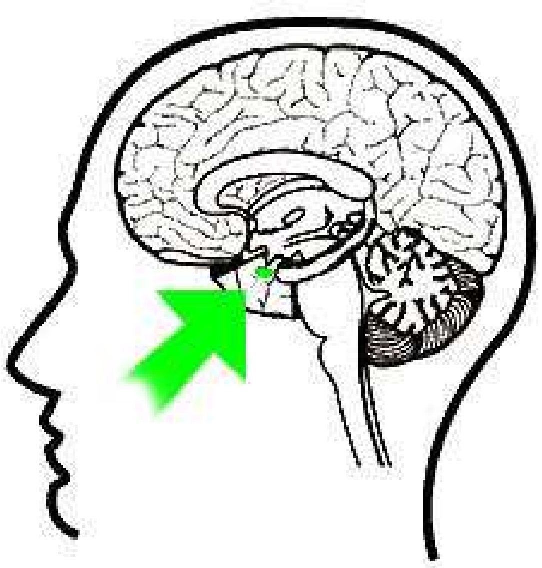


Figure 2.8: The amygdala in the brain

As exposed in [87], animals exhibit fear responses when return to a situation where they previously experienced fear. For example, an animal, that has experienced fear in a chamber due to footshocks, experiences fear when it returns to the same chamber and the footshocks are not present. This is called **contextual fear conditioning** and it depends on the amygdala [73].

According to [73], the study of fear conditioning has shown two afferent routes involving the amygdala (Figure 2.9): the first route (thalamo-amygdala) processes crude sensory aspects of incoming stimuli and directly transfers this information to the amygdala, allowing an early conditioned fear response if any of these crude sensory elements are signals of threat. This enables automatic (or reactive), unconscious emotion activation before we have time to think about our responses [54], that is, without cognition. The second route (thalamo-cortico-amygdala) implies a more complex analysis of the incoming stimulus and results on a slower, conscious, conditioned emotions response. In this case, a cognitive or deliberative process could be involved. This longer pathway is more influenced by social and personal decision making processes and thus can reflect culture-specific emotional responses [54].

Experiments have found that temporal lobotomy (suppression of the area of the brain where the amygdala is located) in animals results in fearless behaviors. Considering experiments achieved by Klüver and Bucy [88], monkeys' behavior were studied in relation with fear. Normal wild monkeys, which has been captured, are afraid of people: when a person tries to approach a monkey in a cage, it escapes running to other corner and remain there. In contrast, monkeys with bilateral temporal lobectomies experienced some kind of

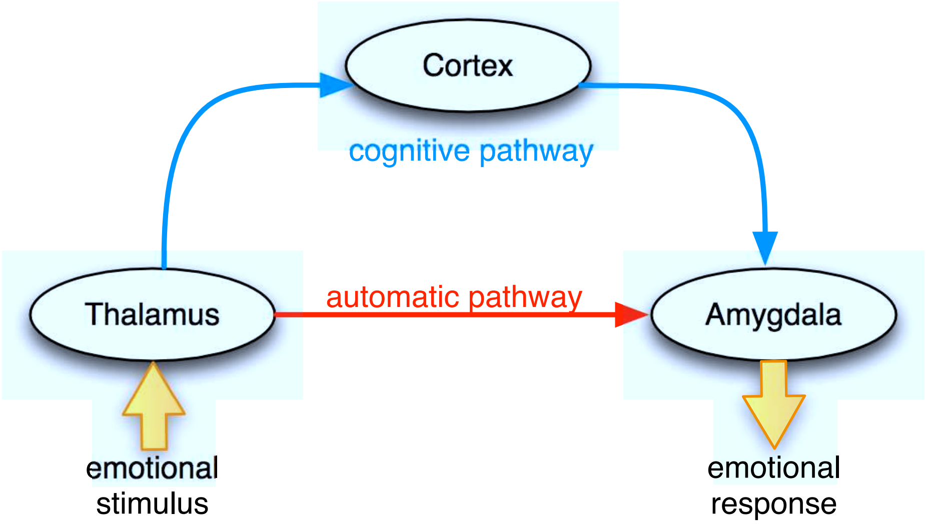


Figure 2.9: Fear pathways involving the amygdala

fearlessness: people approached them, touched them, and even stroke them, and picked them up. Amygdalectomies in rats and lynxes reflect the similar results and fearless behaviors. Damage to the amygdala also affects to fear in humans [87, 89]. Thus, fear provides animals with the escaping behaviors required at certain situations to survive (humans can be dangerous for monkeys). Then, fear is an adaptive response to dangerous situations.

Destruction of the amygdala affects to all emotions as well as showing a reduction on emotionality: expression and experience emotions are considerable flatten when amygdala is removed. In contrast, intelligence appear to be normal. These symptoms are also shown on humans with temporal lobe lesions.

As stated, the amygdala is also involved in the modulation of memory consolidation. By means of painful experiences, animals learn to avoid various behaviors because they are afraid of been hurt. Those hurting experiences are quickly and long-lasting memorized [52] due to the emotional content given by the amygdala. Therefore, amygdala and emotions are involved in the consolidation of long-term emotional memories too. Moreover, the amygdala has been associated with the modulation of other cognitive processes, such as visual perception [73].

Previously, the usefulness of fear has already been mentioned. In contrast, fear can be disadvantageous if anxiety is excessive, persistent, or if threatening situations are not well recognized. Inadequate anxiety might result in anxiety disorders typical from humans. From a psychological perspective, these anxiety disorders can be seen as an inappropriate experience and expression of fear.

Following, few anxiety disorders are briefly commented to give an idea about the incorrect use of fear. For example, General Anxiety Disorder corresponds to a long-lasting, unrealistic or excessive worry [50]. Approximately, Post-traumatic Stress Disorder is related to intense or unrealistic worried suffered when stimuli related to a past trauma are present. Also, phobia is an intensive anxiety due to an exposure to situations leading to avoidance behaviors. In particular, a social phobia causes the avoidance of any social interaction.

2.5 Summary

This chapter has introduced the general concepts involved in the generation of behaviors in animals. Several concepts, such as homeostasis, drives, motivations, external stimuli, and emotions, will be re-used in the following sections to design and implement a decision making system of a robot. These concepts have been biologically and psychology justified in this chapter.

As a general idea, animals exhibit specific behaviors due to the processes occurred in the brain. Physiological needs (food, warm, drink,...) and non physiological needs (curiosity, sex, happiness,...) will guide the behavior through various responses. The observed behaviors are based on the need to satisfy a drive or on the hedonic reward.

In this dissertation, the robot has certain needs (drives), that need to be satisfied, and motivations. Following the homeostatic approach, the decision making system will be oriented to maintain those needs within an acceptable range. These needs will not be just limited to physical ones (as it is stated in the classical point of view of the homeostasis), but psychological and social necessities too. Throughout this thesis, *drive* and *need* will be used as synonyms and they are totally interchangeable.

Emotions influence several aspects of our daily life, e.g. the decision making or the learning. The eliciting of emotions (the appraisal) and the emotional reactions can be inherited or learned, and they can be automatically or deliberatively performed. In this work, artificial emotions exploits their learning and automatic aspects. Artificial motions are involved in the decision making process and the learning process.

In particular, the emotion of fear is carefully studied. Fear provides animals with a self-defense mechanism which helps them to avoid dangerous situations. This mechanism has inspired the implementation of fear in a robot and animal-like behaviors based in this emotion have been observed in the robot.

CHAPTER 3

State of the Art

3.1 Introduction

This chapter presents the current state of the art related to the main theme of this thesis. Since this work covers several fields, they are reviewed and the most relevant works are commented here.

This dissertation introduces a decision making system applied to a social robot where motivations and emotions play a key role. Therefore, initially, the most famous social robots are presented. Then, the attention is centered on control architectures where motivations and emotions work as an adaptive mechanism shaping the robot's behavior. After, the most relevant works are compared considering several aspects and the differences with the work developed in this dissertation are established.

3.2 Social Robots

Before talking about social robots, the question *What is a robot?* must be answered. Many researchers can answer this question with different definitions but this the author of this thesis likes the definition given by Maja J. Matarić in [36]:

> *A robot is an autonomous system which exists in the physical world, can sense its environment, and can act on it to achieve some goals.*

According to this definition, in the late Forties, Grey Walter built the first robots that were named as tortoises (Figure 3.1). They were three-wheeled robots with light sensors and bumpers connected to the drive wheels. With these simple mechanisms, Walter's tortoises shown biomimetic behaviors such us find the light, back away from the light, or head towards the light [36].

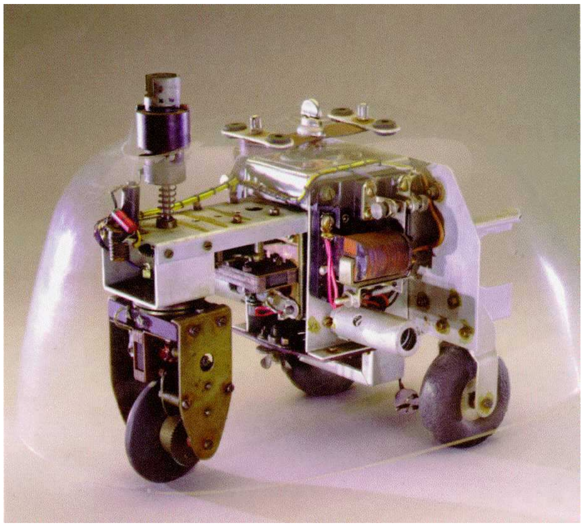


Figure 3.1: Gray Walter's tortoise

From Grey Walter's tortoises, robots have largely evolved and their capacities have been extended. Recently robots are moving from factories and specialized environments to homes. Even for the general public, robots seem to be something relatively common. For instance, many science fiction films have deal with robots (*Metropolis*, *Blade Runner*, *Star Wars*, *AI*, *I, Robot*, *Wall-E*, etc). Even though these fiction-robots are rather far from real robots, all of them are easily understood by people watching them. Therefore, they are endowed with some kind of social ability.

Considering the definition given by Bartneck and Forlizzi [90], *"a social robot is an autonomous or semi-autonomous robot that interacts and communicates with humans by following the behavioral norms expected by the people with whom the robot is intended to interact"*. This definition is the first one that emphasizes the human-robot interaction and communication. Following these ideas, many robots are claimed to be social. In this section, some of the most remarkable social robots are reviewed according to the different purposes they were designed for: research, entertainment, therapeutic, or assistance. This review does not pretend to be an exhaustive survey but an overview of the most important social robots.

3.2.1 Social robots for research

There are many research centers where human-robot interaction is one of the main investigation areas. Two of the most relevant centers, the IRC and the MIT, are analyzed considering their social robots. These centers have been working in this field many years and different robotic platforms have been developed and tested. Following, a brief review about their famous social robots is presented

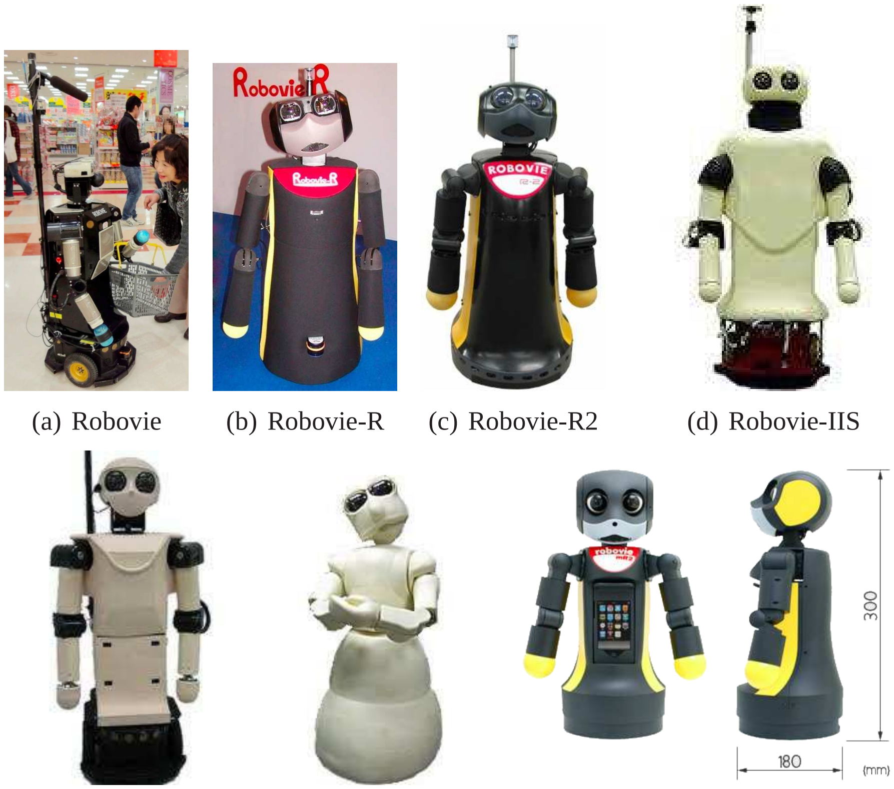
▶ Intelligent Robotics and Communication Laboratories (IRC), from Advanced Telecommunications Research Institute International (ATR) in Japan, is a research group with a long tradition of social robots. **Robovie** (Figure 3.2) is a humanlike appearance robot which is designed for communication with humans [91, 92]. In order to achieve it, it is endowed with the same kind of sensors humans have: vision, a sense of touch, audition,etc. It is 120 centimeters high and its weight is around 40 kg. Many sensors and actuators are spread over the entire robot. It is worth mentioning the omnidirectional vision sensor on top of the head which provides a 360 degree visual field. Apparently, the robot's behaviors are totally predefined by the developer by coupling modules. Different evolutions of this robot are shown in Figure 3.2.

In 2009, **Robovie-mR2** (Figure 3.2(g)) was presented [93]. It is the Robovies' little brother (it is 30 centimeters high). Its creators state that it is a communication robot which is connected to the world through an iPod Touch placed at its tummy. It communicates by means of gestures made by its arrangement of degrees of freedom: four in each arm, three in its neck, two in each eye, one in each eyelid, and one for its waist.

▶ Massachusetts Institute of Technology (MIT) has "produced" several relevant social robots (Figure 3.3). To the best of the author's knowledge, the first social robot is **Kismet** (late 1990s) developed by Cynthia Breazeal. The robot is an expressive anthropomorphic robot head that engages people in natural and expressive face-to-face interaction (Figure 3.3(a)). It perceives a variety of natural social cues from visual and auditory channels, and delivers social signals to the human caregiver through gaze direction, facial expression, body posture, and vocal babbles. Kismet is endowed with a motivational system which has drives, motivations, and emotions (it is analyzed in the next section).

In 2004, Breazeal presented **Leonardo** (Figure 3.3(b)). It quickly and effectively learns from natural human interactions using gestures and dialogues, and then cooperate o performed a learned task jointly with a person [94].

The last robot from MIT, **Nexi** (Figure 3.3(c)), is a small mobile humanoid robot (the size of a 3 year old child)) that possesses a novel combination of mobility, moderate dexterity, and human-centric communication and interaction abilities. This kind of robots are referred as ”MDS” for Mobile-Dexterous-Social. The purpose of this platform is to support research and education goals in human-robot interaction, teaming, and social learning [95]. This robot detects the emotions of humans and acts accordingly.



(a) Robovie (b) Robovie-R (c) Robovie-R2 (d) Robovie-IIS

(e) Robovie-IIF (f) Robovie-IV (g) Robovie mini-R2

Figure 3.2: Several version of robot Robovie from ATR-IRC

3.2.2 Social robots for entertainment

In late Nineties, Sony (Japan) presented the first commercial robot-dog called **Aibo** (1999) (Figure 3.4(a)). Aibo is a well-known pet-style robot which was designed to maintain a lifelike appearance [96]. According to its specifications, it is able to express emotions through LEDs placed at its head, recognizes speech and faces, it has a wide repertory of predefined actions, and learns from the user's preferences and the environment.

Other robot developed by Sony is **Qrio** (2004), the biped humanoid robot (Figure 3.4(b)). It is also a small size robot (58 cm and 7 kg) intended for entertaining people by interacting with them through movements and speech [97]. It understands many spoken commands, says thousands words, and even learns new ones. Flashing colored lights around its eyes are used to express emotions. Its most important quality is its motion-control system that maintains its balance as it walks, runs, hops and dances. Sony had created the world's first running humanoid robot [98]. Initially, Qrio was planned to be marketed too, as Aibo; however, it was ever a prototype, and was not launched commercially. After many years and different generations, in 2006, Sony stopped both robots developments.

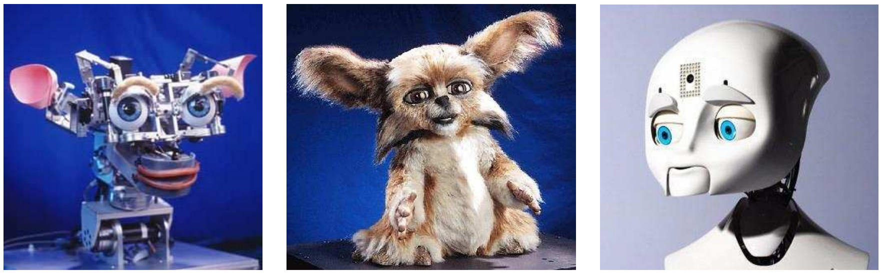

(a) Kismet (b) Leonardo (c) Nexi

Figure 3.3: Social robots from MIT

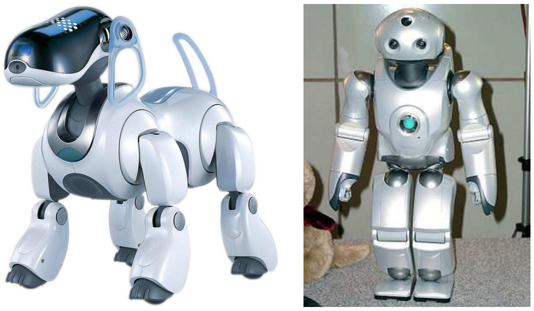

(a) Aibo (b) Qrio

Figure 3.4: Social robots developed by Sony

3.2.3 Therapeutic social robots

Many social robots have been applied to therapeutic purposes; some of them are shown in Figure 3.5. Omron Corporation (Japan) developed **NeCoRo** (2001) (Figure 3.5(a)): a robotic cat that can be perceived as human companions and used as diagnostic and therapeutic tools in psychological and clinical practice [99]. Its real-life-looking creates a playful, natural communication with humans by mimicking a real cat's reactions. Its feelings are generated according to recognition feedback, which is dependent on configurations based on psychological concepts, leading to cognitive decisions and actions determined by these feelings. Desires to sleep or be cuddled are generated according to physiological rhythms. Via a learning function, personality traits, such as selfishness and the need for attention, will change in response to the owner [100].

Other famous therapeutic robot, **Paro** (Figure 3.5(b)), was first exhibited to the public in 2001. This is an advanced interactive baby harp seal robot developed by the National Institute of Advanced Industrial Science and Technology (Japan). They tried to apply it in treatments similar to animal therapy, a special type of therapy that helps to heal people through contact with animals. Many successful experiments have been achieved with patients in environments such as hospitals and residencies for elderly [101, 102, 103].

Robota (Figure 3.5(c))was developed by Billard as a mini-humanoid doll-shaped robot [104]. Its main goal is to investigate the use of toy robots for normal children and for children with disabilities. Soon after, Robins et al.[105] presented the first results of long-term experiments applying this robot to autism children. Some of these researchers stayed in this research line and they developed **Kaspar** (2005) (Figure 3.5(d)). It is a friendly robot which helps children with autism to understand to read emotions and to engage with the people around them [106]. It has simplified human-like features and a minimally-expressive design that invite children with autism to explore the robot. Several body gestures allow social interaction and collaborative games. Kaspar is remote-controlled by the therapists or even the children themselves. Some encouraging results have shown how some of the children learn about social communication skills in repeated, long-term interactions with Kaspar [107].

Other robots have also been applied to children with developmental disorders. **Keepon** (2004, National Institute of Information and Communications Technology, Japan) is a small creature-like robot designed for simple, natural, nonverbal interaction with children suffering autism (Figure 3.5(e)) [108]. Its design is effective in eliciting a motivation to share mental states. It uses simple bodily movements (rocking, bobbing up and down, and vibrating) to express pleasure, excitement, and fear. Moreover, it has been observed an important role of rhythm in establishing engagement between people and robots [109, 110]. In this case, this robot is also used for entertainment purposes [111].

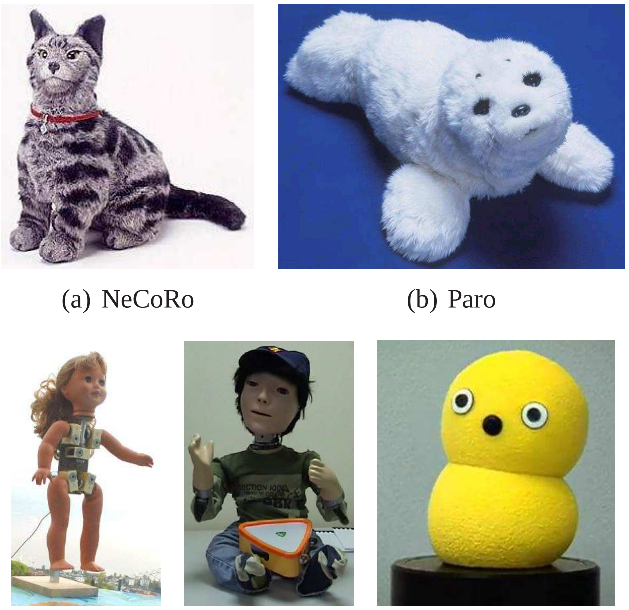
3.2.4 Social robots for assistance

Other robotic platforms support people in different duties, such as manipulating objects, performing daily tasks, or increasing the capacity of people with special necessities. These social robots work as assistants and some of them can be observed in Figure 3.6.

Phillips Corporation developed its social robot: **iCat** (2005). This is a desktop user-interface robot (Figure 3.6(a)) with mechanically rendered facial expressions [112]. It is able to recognize users, build profiles of them, and handle user requests. These profiles are used to personalize domestic functions performed by the robot, such as lighting and music conditions.

The Japanese company NEC also developed its research prototype communication robot called **PaPeRo** (first version on 2001) which is intended to live with people and serving as companion, in particular to children and elderlies (Figure 3.6(b)). It is endowed with autonomous behaviors (walking about, self-recharging,etc.), can play games and can be remotely operated. A visual friendly development environment can be used for creating new actions or functions.

Olivia (Figure 3.6(c)) is a receptionist robot created in the A*Star Social Robotics lab, Singapore. Using its 17 degrees of freedom, Olivia is a social robot designed mainly for


(a) NeCoRo (b) Paro

(c) Robota (d) Kaspar (e) Keepon

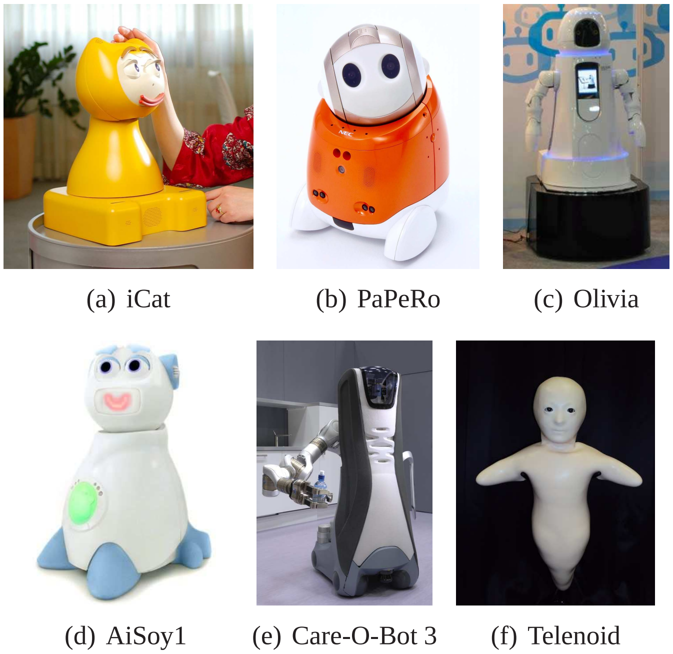
Figure 3.5: Therapeutic social robots

human-robot interactions and communication using speech, vision, and gestures [113].

The Spanish company AISoy Robotics is marketing the robot **AiSoy1** (Figure 3.6(d)) which has its own personality. It is used for entertainment and educational purposes. It expresses emotions by means of its face, voice, and lights [114].

In Fraunhor IPS, Germany, researchers have been working for more than ten years on a mobile service robot that performs supporting tasks in home environments. The last version of their robot, called **Care-O-Bot 3** (Figure 3.6(e)) was presented in [115] (2008). Also, it is meant to be applied in an eldercare facility in order to support the personnel in their daily tasks.

Finally, Figure 3.6(f) shows **Telenoid**, an android with a minimal human appearance for transferring different people's presence to distant places regardless of their personal features [116]. Its covering skin is made of high quality silicon so that it feels as pleasant and soft as human skin when touched. The remote person operates the android by an intuitive tele-operation system. The operator's face directions, mouth movements, and facial expressions are sent to the Telenoid. Also the operator's voice is outputted from a loud speaker embedded inside the Telenoid.



(a) iCat (b) PaPeRo (c) Olivia

(d) AiSoy1 (e) Care-O-Bot 3 (f) Telenoid

Figure 3.6: Assistant social robots

3.3 Control Architectures based on motivations and emotions

Traditionally, decision making systems in robots depend on the control architecture and its characteristics, and vice-versa. In this dissertation, the control architecture running in the robot Maggie (Chapter 5) is extended by the addition of a decision making system with emotions and motivations. Then, since emotions and motivations are one of the main issues in this work, the main control architectures working in real robots with emotions and motivations are studied.

Recently, some authors have implemented cognitive-related concepts in their control architectures, such as motivations, emotions, learning, etc. In this section, a review of these works is presented and a special interest is put on those that have inspired this research.

Several architectures for robots use motivations and emotions. Redko affirms that motivations ensure fine adaptation of agents to external environment variations [117]. Most of the robotic studies regarding emotions employ them mainly for expressing the affective state when the robots interact with humans. For example, Hirth et al. [118] have developed the UKL Emotion-based Control Architecture who claim to implement five emotional functions (regulative, selective, expressive, motivational, and rating). However, after a deep reading and analyzing the experiments presented, apparently due to different understandings of these emotional functions, just few of them are covered and, essentially, the expressive one. In contrast, fewer works have studied how emotions influence other cognitive states, such as motivations or decision making [119]. Matsuda [120] considers that emotions should be incorporated into decision making of robots in order to endow them with sociability.

In spite of the interest focused towards works implemented on real platforms, few relevant works implemented just in virtual agents (at least in a first stage) have also been included in this compilation. This compilation is chronologically sorted for a more comprehensive reading.

3.3.1 The Cathexis architecture (Velásquez, 1997)

To the best of the author's knowledge, one of the first works that considers emotions as an integral part of the decision making process was developed by Velásquez [121, 1, 122, 123]. He shows how drives, emotions, and behaviors can be integrated into a robust agent architecture named *Cathexis*. This architecture models some of the aspects of emotions as fundamental components within the process of decision making. It has a distributed model for the generation of emotions and their influence in the behavior of autonomous agents. The Cathexis architecture is formed by three main modules: the Drive System, the Emotion Generation System, and the Behavior System (Figure 3.7). In this model, the emotional system is the main motivation of the agent.

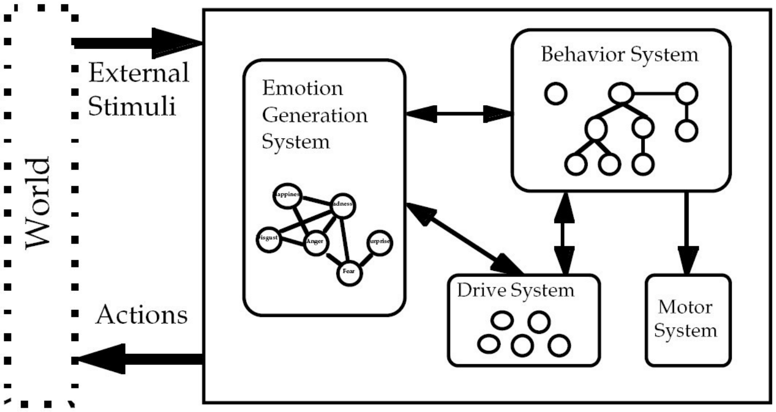


Figure 3.7: General view of the Cathexis architecture by Velásquez [1]

Drives represent needs that motivate the agent into action, so they work as internal stimuli. Each drive uses releasers to identify special conditions which either increase or decrease the value of the associated drive. A releaser regulates a variable within a certain range. When the variable is not inside this range, an error signal is produced and fed to the appropriate drive. Therefore, several error signals can be combined into the same drive [1].

In this work several basic emotions are modelled. Each one is not a single affective state but a family of related affective states (e.g. fear, fright, terror, panic, etc.) which shares certain characteristics (antecedents, expression, reactions, etc.). These characteristics distinguish one emotion family from other. For simplicity, an emotion family is just referred as emotion.

Emotions can be elicited by internal (e.g. drivers, sensorimotor processes) and external (events in the environment) stimuli which are constantly monitored by emotional releasers. Emotional releasers constantly check fo the appropriate conditions that would elicit the emotion they correspond to. Velásquez takes into account the existence of cognitive and non-cognitive releasers, which are classified as neural, sensorimotor, motivational, and cognitive. At first, these emotional releasers were pre-wired [121, 123]. In later works [1], the system learns them through emotional experiences associating the emotions with different stimuli. These new emotion-stimuli pairs will influence in future selection of actions when the same stimuli is present again.

Each emotion has an activation threshold (over it, the emotion influences other emotions and the behavior system) and a saturation threshold (maximum arousal for an emotion). Despite of the discrete approach to emotions, each emotion has an intensity value which is affected by its previous level, the emotional elicitors, and the interaction with other emotions (inhibitories and excitatories). Moreover, each emotion has a particular decay function which controls the duration of the emotion once it has become active. Emotions also interact with drives, and vice versa. For instance, the *hunger* drive might increase the *distress* or *anger* emotions, or high levels of *sadness* might decrease *hunger* [122].

All emotion processes run in parallel and constantly update their intensities. Actually, more than one emotion may be active at the same time. Once an emotion is active, it can excite or inhibit other emotions (e.g. *fear* inhibits *happiness*). The co-occurrence of two or more basic emotions at a time results on secondary emotions, such as *grief* is a mix of *sadness*, *anger*, *fear*, and even *surprise* [123].

In this architecture, emotions are differentiated from mood and temperament. Mood is explained as low tonic levels of arousal within emotions. Temperaments are associated to different activation and saturation thresholds for the emotions (e.g. a fearful individual has low level activation for the emotion of fear).

The behavior System selects the most appropriate behavior according to the emotional state at some point in time. It is also a distributed system composed of several self-interested behaviors (e.g. "approach human" or "play") competing for the control. Each behavior, when become active, has an expressive component and influences the motivational system, i.e. it affects the levels of drives, the emotions, moods. and other behaviors. behaviors can also mutually inhibit or excite each other ("wag the tail" might inhibit "running"). The competition for the control is based on the values of each behavior which are determined every cycle by the *behavior releasers*, such as emotions, moods, drives, pain, and external stimuli. Initially, the selection of behavior worked in a winner-take-all manner [123], but later a more elaborated combinatorial mechanism was proposed [1]; in this work, the active, non-conflicting behaviors (e.g. “walk” and “cry”) can issue commands simultaneously. In short, the selection loop initially reads the internal variables and the environment which are used to update the motivation (both emotions and drives). Then, considering the motivations and the external stimuli, the behaviors’ values are computed. With these values, the resulting behavior is obtained.

Velásquez created models for six different emotions (*anger*, *fear*, *distress/sadness*, *enjoyment/happiness*, *disgust*, and *surprise*) that were used in synthetic agents as well as in a pet robot.

3.3.2 Cañamero’s approach (1997)

The work developed by Lola Cañamero is other of the first researches done in this area [124, 40, 29]. In Cañamero’s works, the original idea was that the behaviors 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. Hence, the motivational state constitute urges to action based on internal bodily needs related to self-sufficiency and survival, e.g. the motivation of *cold* is related to the drive *increase temperature*. The intensity of the motivation is a function of its related drive and a certain external stimulus, also referred as environmental stimuli or incentive cues [125]. Once the highest motivation is obtained, the intensity of every behavior linked to this motivation is calculated and the one with the highest intensity is executed. For some behaviors, the intensity determines the strength of the motor actions or the duration of the behavior. Therefore, the motivation with the highest value organizes the behavior of the agent in order to satisfy its drive.

The implemented artificial emotions (*anger*, *boredom*, *fear*, *happiness*, *interest*, and *sadness*) follow a discrete approach and work as monitoring mechanisms to cope with important situations related to survival. Emotions are activated as a result of the interactions of the robot with the world, depending on different events. For example, *anger* becomes active when the goal of the agent is not finished, or *boredom* is activated when the agent is enroll in a repetitive activity. Emotions in this approach work as second-order modifiers or amplifiers of motivations. More precisely, emotions influence, proportionally to their intensities, the decision making process by releasing “hormones” in two ways. First, they can modify the intensity of the current motivation and, as a consequence, the intensity of the related behaviors. In fact, in extreme cases, they can avoid the execution of the behavior. Second, they can modify the reading of the sensors that monitors the variables affected by emotions. Therefore, they can alter the perception of the state of the body, as well as the external world. Moreover, the hormonal release can affect the way behaviors are executed. For example, the sad emotion provokes that behaviors are executed slower. Then, emotions are characterized by a triggering event, an intensity, an activation threshold, a list of hormones which are released when it is activated, a list of physiological manifestations, and a list of physiological variables it can affect. In this work, several emotions can be simultaneously activated, all of which contribute to the behavior by releasing hormones or adopt a winner-take-all strategy [126].

The action selection loop starts by computing the effects of the emotional state and the motivations are assessed. Then, the highest motivation and the behaviors that can best contribute to its satisfaction (those whose effects alleviate the drive) are selected. If none is found, other behaviors that contribute to it to a lesser extent are selected. Finally, when a behavior is executed, it has an associated intensity (the urge) and both the world and the body state change.

Later, Avila-García and Cañamero applied a “hormone-like” mechanism to adapt the actions selection process to dynamic and changing environmental circumstances [127, 2]. Such mechanism modulates the perception of external stimuli in order to adapt the same architecture to new environmental circumstances where the robot competes with others for the same resources. Moreover, this modulation also acts over a drive making the action selection process more sensitive to it. Figure 3.8 shows how the hormones influence motivations and behaviors.

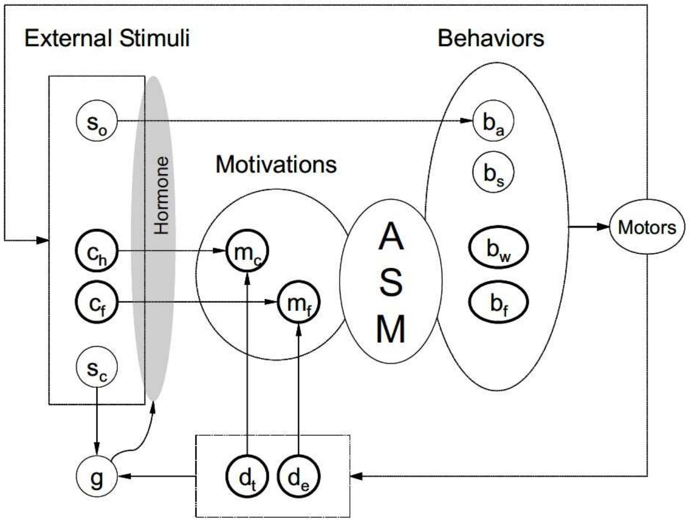


Figure 3.8: Hormone-like modulation for the action selection process proposed by Avila-García and Cañamero [2]

3.3.3 The ALEC architecture (Gadanho, 1998)

Another relevant work is the one presented by Gadanho [128, 24, 129]. In this work, the research is focused on how artificial emotions can improve the behavior of an autonomous robot. In her approach, the robot adapts to its environment using an adaptive controller adjusted by using reinforcement learning. Emotions are used to influence perception, as Cañamero does, and to provide a reinforcement function. In these works, emotions (happiness, sadness, fear, and anger) are determined by internal feelings (hunger, pain, restlessness, temperature, eating, smell, warmth, and proximity), and the relations between each emotions and the feelings are predefined.

In later works, Gadanho presented the ALEC (Asynchronous Learning by Emotion and Cognition) architecture where decision making is approached from two perspectives: emotive and cognitive [130, 3]. Then, the ALEC architecture (Figure 3.9) is mainly composed by the emotion and the cognitive systems. In this architecture, emotions take the form of evaluations or predictions of the internal state and the goals are explicitly associated to a set of homeostatic variables [131]. These homeostatic variables allow to learn the utility of each behavior and make decisions considering that. In addition, a cognitive system provides an alternative decision making process which can correct the emotion system's decision.

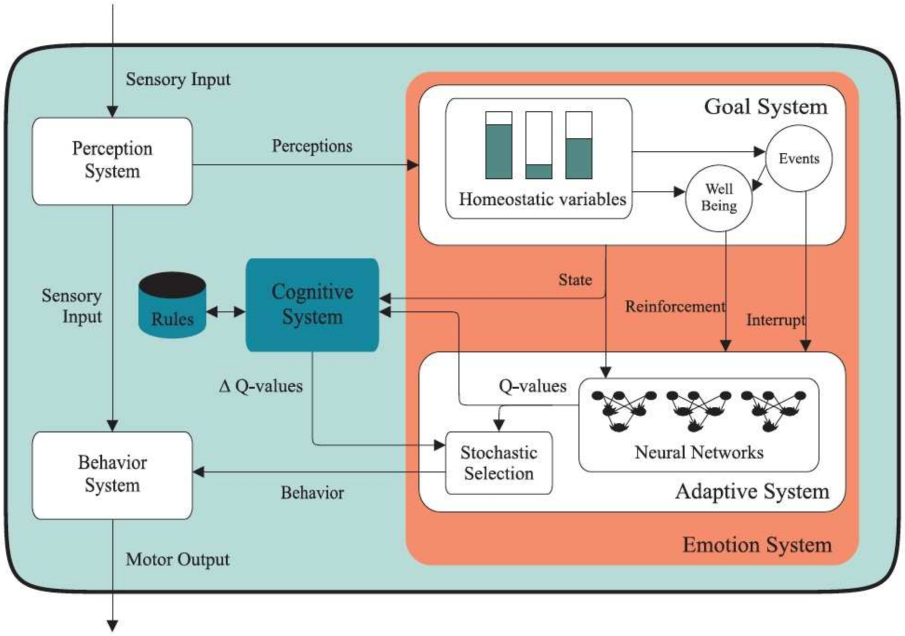


Figure 3.9: The Asynchronous Learning by Emotions and Cognition architecture [3]

The emotion system is composed in turn by other two subsystems: the goal system and the adaptive system. The goal system evaluates the behaviors selected and notifies when a behavior should be interrupted. In other words, it determines the reinforcement and when behavior switching should occur. The performance of a behavior is measured in terms of the state of the homeostatic variables which must be maintained within a certain range. In order to reflect the hedonic state of the agent, a wellbeing value is created which mainly depends on the value of the homeostatic variables, their states, their transitions, and their predictions. This wellbeing value is used as the reinforcement function.

The adaptive system is in charge of the learning process. It implements the Q-Learning algorithm, so it learns the utility value for each action. These values are stored by neural networks which are fed with the homeostatic variables and other sensory data. As a result, the agent will try to maximize the reinforcement received by selecting among all available actions.

Finally, the cognitive system is based on a set of rules extracted from the agent-environment interaction which represent particular successful behavior selections. These rules can be updated, deleted, or even merged. When one of these rules fits the current state, the suggested behavior is promoted by adding a constant value to the respective Q-value.

As said before, following Tomkins' idea that the human decision making process consists on maximizing the positive emotions and minimizing the negative ones, emotions in ALEC architecture are related to pleasant/unpleasant feelings working as reinforcement. The wellbeing value plays this role and it also can be seen as an emotional feeling of the overall state of the agent. Moreover, the learning process results on associating behavior-state pairs expecting long-term wellbeing value which indicates the *goodness* of the available options, similar to the somatic markers proposed by Damasio[132]. The performance is measured in terms of the state of these homeostatic variables which must be maintained within a certain range.

3.3.4 Breazeal's model (2000)

Probably, one of the most influential works in this area is the Cynthia Breazeal's thesis [4]. She continued Velásquez's work and, as far as the author knows, she presented the first social robot, Kismet (Figure 3.3(a)), endowed with a motivational system with emotions and drives. Later, the system was also implemented in the robot Leonardo (Figure 3.3(b)). She proposes a rather complex net of intertwined systems (Figure 3.10): the Emotion System where the robot's affective state is determined, the drives that correspond to the *innate* needs, the Behavior System which is in charge of the arbitration of the available behaviors, and other modules which are directly connected with the hardware.

Breazeal thinks on emotions and drives as two related motivational systems. Drives are involved in the homeostatic regulation processes that maintain critical parameters within a bounded range. Emotions are models of basic emotions which have particular functions. They arise under particular circumstances, and motivate the robot to react in an adaptive

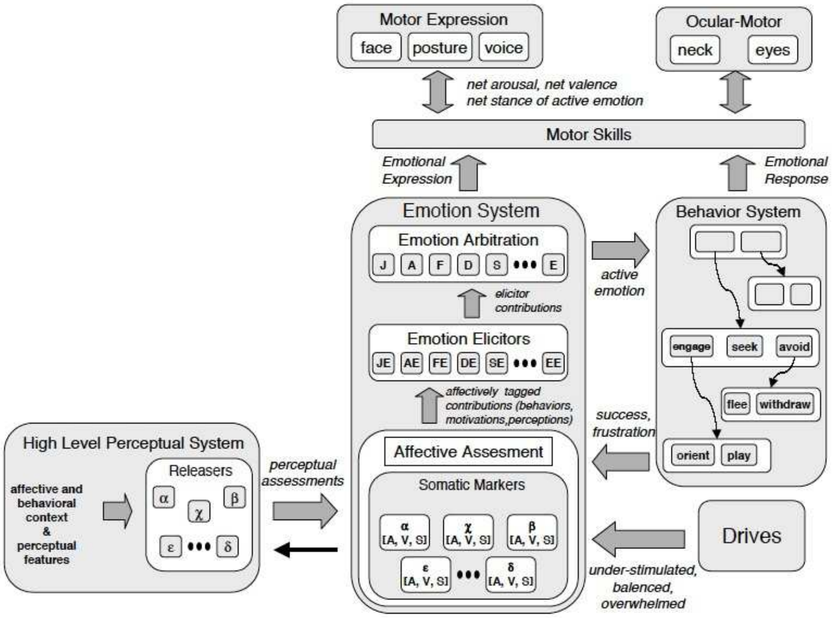


Figure 3.10: An overview of the net of systems in Breazeal's thesis [4]

manner. Each emotion has a corresponding expression which is exhibited when the emotion arises. Breazeal centers her study on the communicative role of emotions and how they improve the human-robot relationships. The role of the emotional system is to influence the cognitive system to promote appropriate and flexible decision making, and to communicate the robot internal states [133, 134, 135].

Kismet's drives influence the behavior selection by passing activation to some behaviors over others. Besides, drives also pass activation energy to emotions influencing the robot's affective state too. The main characteristic of drives is their temporally cyclic behavior, i.e. a drive will tend to increase in intensity unless it is satiated. Moreover, drives have an homeostatic nature: their intensities should be within a bounded range, the homeostatic regime. The changes in a drive's intensity reflects an ongoing robot's need and the urgency to satiate it. Kismet's drives are maintained within the homeostatic regime in a never ending process which involves the satiatory stimuli. When drives are in the homeostatic regime, they spread activation energy to positive emotions. In contrast, when drives are out of the homeostatic regime, negative emotions are enforced.

The Emotion System determines the active emotion in a particular context. Each emotion is elicited under certain, defined conditions and provokes a specific behavior to serve a particular function. Thus, an emotional reaction of Kismet consists of some environmental factors (releasers) and their affective appraisal, a characteristic expression, and a behavioral response.

The emotional releasers are evaluated with respect to: drives, the current affective state, the active behavior, and relevant stimuli; with all, their activation level is determined. Releasers with activation level above certain threshold are affective appraised. Inspired by the Somatic Markers of Damasio [132], each releaser is tagged with three values: the arousal A (how arousing it is), the valence V (how favorable it is, pleasant/unpleasant), and the stance S (how approachable it is) markers. Then, each emotion has an elicitor that filters all the incoming $[A, V, S]$ tuples from the somatic markers and, with those that passes its filter, computes the average $[A, V, S]$. These average values are used to calculate the activation level for the elicitor which is passed to the arbitration phase. In this phase, just emotion elicitors with activation level over a threshold level compite in winner-tale-all manner. Since the activation level of an elicitor informs about its relevance to the current situation, the, the highest one determines the active emotion. This emotion can evoke the corresponding behavioral response and/or affective expression.

Kismet's observable behavior is not just determined by the active emotion, but drives, perceptions, and others are involved too. However, the active emotion spreads activation energy to specific behavior process. If this activation is strong enough, the active emotion decides the robot's behavior.

Kismet's Behavior System is organized into a layered hierarchies of behavior groups (Figure 3.11). Each group contains behaviors that compete for activation with one another (the behavior's relevance is determined by perceptual factors and internal factors). The highest level is responsible for maintaining the homeostatic functions. Here, the influence of the robot's drives is very strong and this motivates the robot to come into contact with the satiatory stimulus of the most urgent need. When a behavior in a group requires more specific tasks these are embraced in a child behavior group representing different strategies for achieving the parent's goal.

As said, an emotion can take control of the robot's behavior by sending sufficient activation energy to its affiliated behavior such that this wins the competition among other behaviors and becomes active. Recalling, each emotion is mapped to a distinct behavioral response. In this model, the active behavior also influences the affective state, and vice versa. For example, the succeed in achieving the goal of behavior is an antecedent condition for eliciting *happiness*.

3.3.5 Other works

Blumberg's approach (1996)

Blumberg presented an architecture for autonomous virtual creatures, or agents, that combines learning with action selection [136]. These virtual creatures are endowed with motivational internal variables used to model internal state, such as level of hunger or thirsty (similar to drives in other works). In this system, the agent learns the existing behaviors

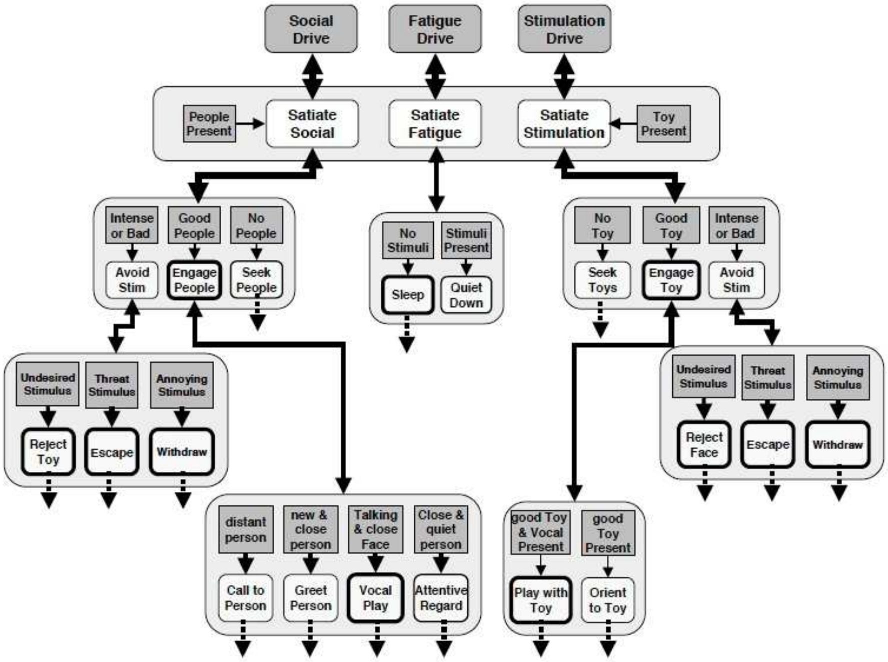


Figure 3.11: Kismet's behavior hierarchy [4]

leading to the fulfillment of some previously-unassociated motivational goal when it is performed in a novel context. Then, the Behavior System coordinates the available high-level behaviors in a potentially unpredictable environment.

Blumberg affirms that, in nature, most of the learning focuses on the discovery either of situations in which a consummatory behavior should become active (it satisfies its associated motivational variable), or of behaviors that bring them closer to attaining some goal (i.e. appetitive behaviors). Thus, it can be said that the internal variables that these consummatory behaviors satisfy lead most of the learning in animals. For that reason, Blumberg adopted this perspective and the motivational internal variables lead the discovery of new strategies for their satisfaction. Based on this approach, the variation in the value of the motivational internal variables due to the activity of behaviors is used as the reinforcement signal for the learning process.

In the Blumberg's approach, there is no centralized learning. In contrast, each motivational internal variables serves as independent reinforcement signal. This means that the behaviors for each motivational internal variable are separately learned.

Waiter-task robots (Murphy, 2002)

A curious application of emotional control in robots is the work presented by Murphy [137]. In this paper, a team of two heterogeneous robots collaboratively perform a "waiter" task. A robot is the waiter (it serves items to an audience) and the other is the refiller (it brings a tray of refills upon request). The controller for both robots is implemented in a script-like manner. Both receive as inputs the task progress and the refiller has an extra input: commands from the waiter. Besides, both have a Behavior State Generator (BSG) achieving the action selection, and a Emotional State Generator (ESG) which determines the current emotion. The emotion can be sent to the BSG, affecting the action selection, or to the sensory-motor level, affecting how behaviors are performed. The authors have defined four emotions (happy, confident, concerned, and frustrated) based on high task-dependent variables (*time til empty* (tte) and *time to refill* (ttr)). For example, the waiter robot's emotional state is *happy* when the time to be refilled is greater than the time it should take to be refilled if the refiller is moving at expected speed. That is, $tte > ttr$. Each emotion has a preprogrammed corresponding action tendency. E.g. if the emotion is *concerned*, the waiter sends the "hurry" request to the refiller, and the refiller attempts to move at her maximum speed. Therefore, the emotion's influence is performed at two different levels: the waiter's emotion alters the action selection, and the refiller's emotion affects the sensory-motor level.

Color shirt-based emotional system (Hollinger, 2006)

In the work presented by Hollinger et al. [138], a continuous multidimensional emotion space is used to determine the affective state of a social robot in large crowds. While moving around, the robot uses a state machine to determine which actions it should perform. When a face is detected, the emotional state, in combination with the person's color shirt, determines the reaction the robot executes. This reaction is composed of brief movements, saying a sentence, or playing a sound.

The affective space to determine artificial emotions is based on the Mehrabian PAD scale, where the axes represent pleasure, arousal, and dominance. So, in this work, twelve emotions are mapped into this three-dimensional space. In this approach, the emotional releasers are related to different color shirts, and each color has a certain coordinate in the PAD scale. The (P,A,D) values for each emotion define the sentence to say, the sounds to play, and the parameters of the controller (maximum and minimum speed, minimum distances, amplitude and duration for wiggling, and other constants). The system was tested on a crowded environment and, during the experiments, people interacted longer when the robot exhibited sad or happy behaviors than when it was angry.

Lisseti's approach (2007)

Finally, another approach is the one presented by Lisseti and Marpaung in [139], where the behavior of the robot is selected according to its current emotional state. They generate this emotional state based on the data received from the input sensors of the robot. In fact, each emotion is related to certain external events, e.g., the parameter of the *Sad* emotion is increased if the door is closed or the robot does not recognize someone. Once the emotional state is determined, the robot will execute the proper action tendency, i.e., the robot identifies the most appropriate (or a set of) actions to be taken from that emotional state.

In this work, each emotion has several properties: the *valence* describes the pleasant/unpleasant dimension of an affective state, the *intensity* represents the importance and urgency of the affective state, *focality* indicates if the emotion is related to an event or an object, the *agency* indicates who is responsible for that emotion (the agent itself of other), *modifiability* refers to the duration and time perspective, *action tendency* identifies the most appropriate action to be taken from an emotional state, and *causal chain* identifies the causation of a stimulus related to an emotion (e.g. happy was caused because something good happened to me)

The resulting emotion is used for determining the facial expression. After, the *Behavior State Generator* executes the corresponding behavior according to the input from the sensors and the *action tendency* of the emotion. For example, *avoid_left_wall* and *avoid_right_wall* behaviors can be activated when the robot is *surprised*, whose *action tendency* is *avoid*.

Full-configurable user-oriented emotional robot (Lee, 2008)

Lee et al. [140] follow a different approach to the use of emotions for shaping robots' behavior. They follow a user-oriented approach in developing an interactive framework for configuring the robot's behavior, i.e. the user can customize the behavior of his own pet-robot. Authors propose a behavior-based control with a full-configurable emotion subsystem for behavior coordination.

The emotion system models basic emotions (happy, angry, fear, bored, shock, and sad) to coordinate the behavior controllers the user has pre-chosen for his pet. Each emotion is independently quantified based on a set of events predefined by the user. For example, a user can define that the appearance of a stranger implies the fear emotion to be incremented in one unit. Moreover, another set of homeostatic variables are defined to describe the robot's body state (e.g. hungry). These variables have to be maintained in a specific range. These ranges, as well as the events which modify the homeostatic variables, can be defined by the user too.

In order to select the appropriate behavior at any time, a feed-forward neural network maps emotion values and body states into the desire behavior. This neural network can be trained by the user in order to obtain the desire pet's behavior.

The TAME architecture (Moshkina, 2011)

Moshkina [5] presented cognitive and psychological models of human Traits, Attitudes, Moods and Emotions for their application to robots. These models were integrated into an architecture called TAME which is intended to influence the perception of a user regarding the robot's internal state and the human-robot interaction itself. All these affective elements strongly influence each other and intertwine in order to show life-like appearance in robots. A conceptual view of TAME is exposed in Figure 3.12.

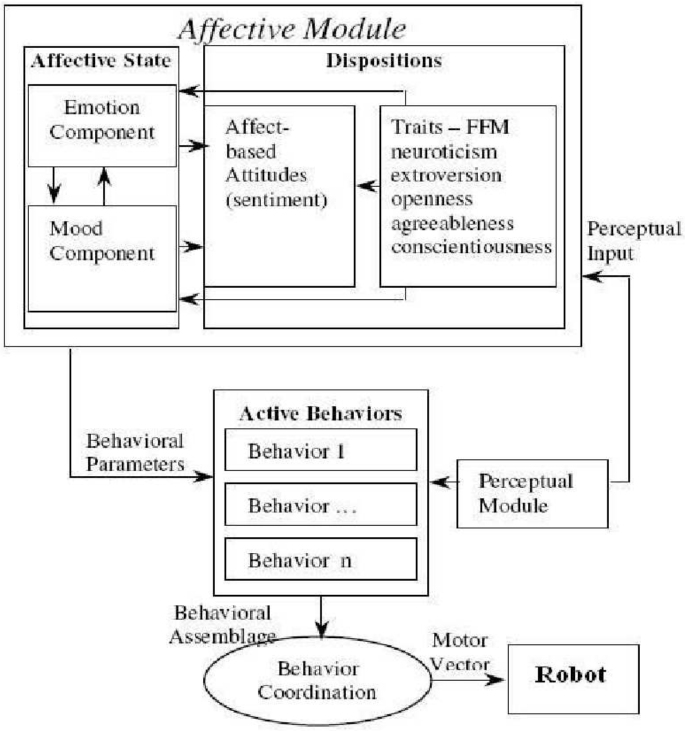


Figure 3.12: Conceptual view of the TAME architecture [5]

Personality traits and affective attitudes represent a propensity to behave in a certain way. Personality traits are not essentially affective, but they influence on other affective phenomena. They are permanent values that identify the patterns of behaviors and affects that characterize individuals. Traits are defined a priory by a human

Attitudes are *"general and enduring positive or negative feelings"* about objects, people, or issues. Attitudes are object-specific and they do not directly alter the behavior, but rather through the emotions they invoke. They justify the use of attitudes because robots sharing attitudes with human companions easier and better engage in interaction.

Affective state is formed by moods and emotions. Mood is low-activation, slowly-varying diffuse affective state; it is as a slow smooth undulation. Moods represent a continuous affective state, cyclically changing and subtle in expression. So, they only produce small effects on the currently active behavior. Expressing mood can alert to changes in the environment or in the robot itself. The level of mood is computed as a weighted summation of external and internal variables. Considering an example from the paper, positive mood is more susceptible to energy consumption, and negative mood to darkness.

Emotions are high-intensity short-duration peaks of affective state and provide fast and flexible responses to relevant stimuli from the environment. In this paper, emotions' functions are mainly communicative and expressive. The selected emotions implemented were *fear*, *anger*, *disgust*, *sadness* and *joy*, because of their universal, well-defined facial expressions. Emotions are endowed with a set of properties which define their intensities: activation, saturation, response decay, and linearity. Moreover, they are highly dependent on traits and moods.

Emotions alter the robot's behavior, from example *"subtle slowing to avoid disgustful object"* or *"drastic flight in response to extreme fear"*. Experiments presented show how the robot expresses the corresponding emotion or how emotions modify the current behavior (for instance, slowing down the walking speed), but they do not decide the goal or the behavior to execute.

Behavioral arbitration or the changes to the behavioral parameters are performed on the robot controller side, providing high portability and scalability. Actually, affect can be implemented in continuous or discrete manner. In the humanoid Nao, a discrete approach with a number of affective expressions has been implemented. The appropriate expression is selected according to the actual values of TAME variables. These variables influence the robot's behaviors by altering certain parameters or selecting a predefined affective expression. Then, as said before, emotions are mainly employed to show the robot's internal state, and they are not involved in the decision making process, which is achieved in the robot's side considering the TAME variables.

The emotional robot head MEXI (Esau, 2011)

Esau and Kleinjohann [6] present a *"fully emotional competence"* robot head called MEXI, which is intended for interaction with people by communication. It recognizes human emotions from speech and facial expressions and it is able to adequately react to them. In addition, MEXI is endowed with drives and artificial emotions which are used to manage the control of reactive behaviors, and the corresponding robot's internal state is shown by facial expressions and utterances. The presented control architecture is a model-free approach, so there is not an explicit world model and goal representation.

MEXI is endowed with a set of three drives, *communication*, *playing*, and *exploration*, for achieving pro-active behavior. For example, when the *communication* drive is very high, MEXI looks for people in the environment. When a person is perceived, the *communication* drive is satisfied by following their face with its view and it implies the emotion *happiness*, which is expressed by the smiling behavior. MEXI's basic behaviors are classified into *expressive behaviors* when they depend on its emotions state generating the corresponding expression (facial, speech, and prosody), and *coping behaviors* when they depend on the drive state (talk, playing, following faces, etc).

For each drive, a range is defined and when the drive is inside, it is in homeostasis, i.e.

it is balanced. The course of a drive changes over time, the drives internally increase and decrease in a cyclical manner, even in the absence of external stimulation. By default, without external stimuli, the drives follow a sine wave. The default course of a drive is altered according to the acceleration factor, which determines the influence of stimuli. Stimuli are perceptions and robot's own behaviors influencing specific drives. They may accelerate or decelerate the drive's increase or decrease. The *coping behaviors* satisfy drives. This course of drives is not very accurate in relation to animals, e.g. hunger in animals always increases until it is satisfied due to the ingestion of food. According to Esau and Kleinjohann's model, hunger would increase and decrease without any ingestion, just over time.

In relation to emotions, *happiness*, *anger*, *sadness*, and *fear* have been implemented in MEXI. It strives for *happiness* and avoids the others. As drives, emotions develop over time. For each emotion, there is a threshold that defines when the robot has to show the emotion. Stimuli influence emotions too by an acceleration factor related to emotions. This acceleration factor is affected by the current perceptions and drive state. In this work, drives are linked to emotions and, hence, the variation of a drive concerning a certain emotion, can influence its increase and decrease.

In order to show emotional competence, MEXI selects behaviors according to the emotions of the human counterpart. Moreover, it also maintains and regulates MEXI's emotional state in such a way that its drives are kept in the homeostatic area, and positive emotions are reinforced while negative ones are avoided.

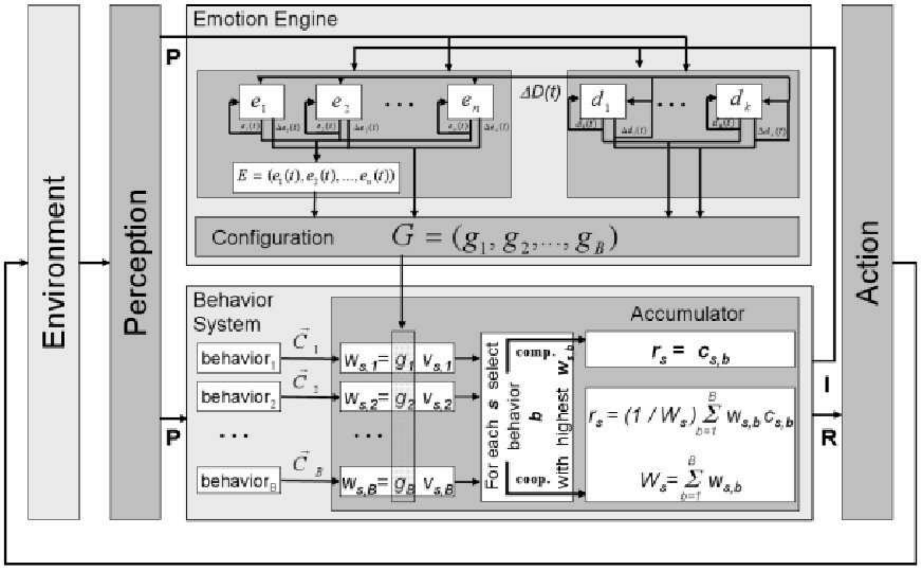


Figure 3.13: Control of the Behavior System by the Emotion Engine in the robot MEXI [6]

Behaviors are weighted according to external perceptions, the emotional state, and the drives. When the strength of an emotion reaches certain threshold, the gain value for certain predefined *expressive behaviors* is set to the maximum value. In relation to *coping behaviors*, their gain values depend on the drives: if a drive increases/decreases, the gain for the corresponding *coping behavior* is increased/decreased by a certain amount per time. Then, the relation between drives and *coping behaviors* are predetermined by the robot's designer.

Once behaviors have been weighted, they are ranked. Considering that the available behaviors are also classified as *competitive* or *cooperative*, if the behavior with the highest value is competitive, this is the resulting behavior. Otherwise, the first behavior is cooperative and then all *cooperative behaviors* are proportionally combined to produce the final one.

A multi-agent approach to emotions (Nair, 2011)

In other paper, Nair et al. [141] present a multi-agent approach for using emotions in robots. In this work, dedicated *emotion agents* work concurrently. Multiple software agents interact with one another to produce a set of emergent emotions based on the external perceptions the robot perceives. These agents stimulate or suppress other such *emotion generating agents* for certain time to finally result on an *emotional control juice* that can eventually alter the robot's behavior.

An interesting point is that adrenaline is used as inspiration for the rate at which the sensors are sampled. Higher negative emotion generation causes this rate to increase making the system more aware of its environment. Higher positive emotions intensities cause this rate to slow down. This metaphor for adrenaline is determined by the robot's mood which is generated by fuzzifying the emotion intensities. If the mood goes down, the system starts to sample at faster rate as an attempt to ameliorate its condition. The higher the mood, the lesser is the sample rate.

The emotion intensities are determined by the emotion resource, a time-to-live and decay for stimulations and suppressions. The concept of emotion resource relates to the affective capacity of a system to generate the associated emotion intensity. This is similar to the intensity of happiness of a poor man who finds a 100€ bill and subsequently finds more such bills within a short time: the emotion intensity does not actually double or treble. The emotion resource diminishes with every generation of the emotion intensity, it has a maximum limit of emotion generating capability, and it does not eventually lose its secreting capability. After certain time the resource is depleted, the *emotion agent* charges, thus augments, the emotion resource. Also external sources augment resources: rewards augment the resource of positive emotions, and penalties equally perform for negative emotions (reward and penalty are based on the task the robot performs). This is referred as the replenishing capability of *emotion agents*. The conversion of an emotion resource into emotion intensity is proportional to the intensity of the stimulations received and also to the resource currently available. Moreover, the emotions decay over a period of time.

The system is implemented in a Lego NXT robot where three emotions control its motion along a path: *happy*, *fear*, and *anger*. Each of them are determined by the inputs coming from relevant, specific sensors: the robot becomes happy when it senses an increasing light intensity gradient, fear is sensed when something comes very close, and angry when the sound level exceeds a threshold. Rewards and penalties are simulated. The speed of the robot is modulated based on the mood, so it is proportional to the sampling rate. In the experiments shown, the robot moves along a straight path. It moves away faster from areas with obstacles, sounds, and darkness. The dynamic sampling rate inspired on the effects of adrenaline drives the robot out of situations which increase the negative emotions. Such a mood thus serves as the *emotional control juice* to moderate the behavior of the robot.

The use of separate agents for emotion generating allows to run them in different locations, or easily add/remove concrete emotion generating agents (scalability). The elaborated dynamics of emotions contribute in making the transitions from one emotion type to the other more biological equivalent. However, the generation of emotions is very steady and predefined. This makes the experiments rather simple, and simple results.

Fuzzyfied emotions (Kowalczuk, 2011)

Kowalczuk and Czubenko [7] propose to utilize models of psychology of living creatures for adapting autonomous robots to the environment. They are more concerned about the interaction of the robot and its environment, where humans can be part of it too, instead of focusing on human-robot interaction as others do. In their paper, robots are endowed with a set of needs and these are influenced by several emotions. Then, emotions are used for modeling the sense of fulfillment of needs.

Using fuzzy methods, each need results in three possible states labeled as satisfaction, pre-alarm, and alarm. Emotions (referred as *classical emotion* in the text) are some states of mind, which modify the system of needs and reactions. The *classical emotion* is reduced to a single variable, and decomposed into seven fuzzy sets, representing each one a single fuzzy emotion (Figure 3.14). These fuzzy emotions are labeled as *fear*, *anger*, *sadness*, *indifference*, *happiness*, *curiosity* and *joy* just for differentiating them. Also emotions are modulated by "*impressions*" related to external objects (this is referred as "*sub-emotions*" by the authors). Besides, the concept of mood is also applied in this work. In this case, its value is formed by the *classical emotion* and moderates the fuzzy membership parameters of the needs.

The decision on the reaction is made by a combined criterion composed by the maximum satisfaction level of the needs and a minimum distress level (related to the alarm and pre-alarm thresholds). The influence of reactions on the needs are predefined. Using a fuzzy-neural network, each reaction is computed by performing a simulated estimation of the effects of its application. Then, the best reaction is executed with the expectation

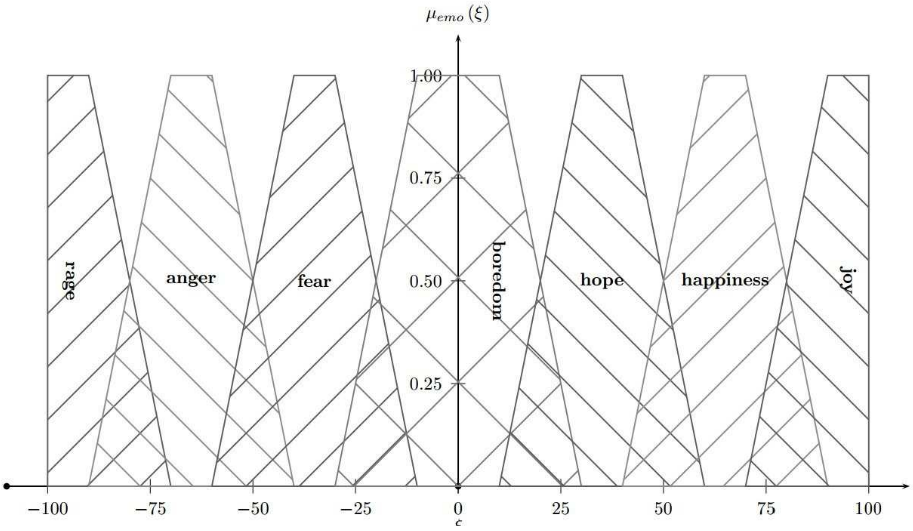


Figure 3.14: Fuzzy model of *classical emotions* [7]

that the satisfaction of needs will be improved. The reaction, the context of the current emotions, and the amendments on particular needs are stored to improve the estimation of the effects of any reaction and, thus, to optimize the decision.

This system has been tested on simulation and a simplified version, based solely on needs, has been implemented on a laboratory mobile platform in a easy environment. Authors state that the robot acts like a baby satisfying the robot needs.

Arkin's moral emotion of guilt (2012)

An interesting application of emotions in military robots is presented in [142]. Arkin proposes a moral decision making for lethal military robots based on ethical issues. Moral emotions are used to modify the robot's behavior based on the results of its actions. Focusing on ethical behaviors in autonomous agents, Arkin considers the moral emotions proposed by Haidt [143] and, particularly, *guilt* is implemented in his system. In this case, *guilt* follows the definition given in [143]: guilt is "*caused by the violation of moral rules and imperatives, particularly if those violations caused harm or suffering to others*". In Arkin's work, guilt is originated when the military robot's actions cause undesired effects and it is used to alter the future robot's behavior by preventing the same actions to occur.

Guilt is implemented as a variable which will increase according to the feedback provided by external operators and self-monitoring processes. When this variable exceeds a certain threshold, the robot can not perform any lethal action because it is not considered as ethical and there is no option for permission-to-fire. Once this happens, the robot can stay in the battle field but just for non-lethal operations (surveillance, reconnaissance,etc.). This non-lethality state remains active in the robot until an “after-action review” or an operator override the restriction and explicitly takes the responsibility.

In order to modify the robot’s behavior, the weapon system of the robot has been classified according to the different destruction potential, each one with a different guilt threshold. Once guilt value exceeds one of these thresholds, the weapons corresponding to the associated class are deactivated. The higher destructive a weapon class is, the lower threshold is assigned. When guilt reaches its maximum, all weapons are deactivated and the robot cannot engage targets any more until “after-action review”. This use of guilt recognizes the bad behaviors and provides the opportunity to reconsider specific actions and their results for the future. Hence, guilt can alter the robot’s behavior for an autonomous agent.

3.3.6 Comparative analysis

After this overview, it seems that there are several elements present in most of the works. The majority of the authors considers some kind of internal variables representing needs. Furthermore, external perceptions directly influence the decision making process itself, or indirectly through altering other elements such as motivations, emotions, or perception. In addition, emotions influence, in one way or another, the behavior selection and behaviors alter the internal state of the robot. Then, these ideas will be kept in this thesis too. However, the above commented works differ in many aspects. Next, the most relevant differences are mentioned.

The homeostatic approach

Drives, needs, or internal stimuli, are all synonyms of the same concept related to homeostasis in living beings. This implies a temporally cyclic course of these drives, but the homeostatic approaches are significantly different. In this thesis, drives have an ideal value of zero, and any deviation from it represents the need and urgency to satiate it. The bounded range in Breazeal’s drives does not exist in the model considered in this thesis, but there are activation levels for motivations that play a similar role but at a higher level. Velásquez suggests releasers for drives that check certain conditions for increasing or decreasing their values.

The influence of behaviors

Despite of almost all authors suggest the influence of behaviors to drives (specially as satiatory stimuli), Breazeal proposes also that behaviors influence over emotions too. However, in this thesis, the focus is put on how emotions help to shape the decisions made by the robot, so the flip side is not covered.

Behavior arbitration

Just few of the authors provide a mechanism to execute several behaviors concurrently (Velásquez and Esau), in contrast with the others who propose a winner-take-all manner where one behavior is exclusively executed according to the most relevant emotion or motivation. The present dissertation follows the last approach.

Different models of emotions

Two main categories of models for emotions have been observed: continuous (or dimensional) and discrete.

Many researchers think that the relation between situations and emotions is mediated by a set of intermediate variables. These variables act as dimensions of an affective space and each emotion is associated to a different zone of the affective space [31, 144, 7]. These dimensional theories represent emotions as points in a continuous dimensional space.

On the other hand, other authors, such as Velásquez or Cañamero, consider emotions as discrete categories. This approach is more focused on looking for the adaptive function for each emotion and, regardless of its implementation, include them into a model [145]. These represent a functional approach, also referred as rational theories [78].

In the discrete emotional approach, dimensions of emotional intensity can be still employed, but these are applied within each emotional category (e.g. Moshkina, Esau, or Lisseti). However, as Lazarus says [145], the dimensional theories underestimate the importance of distinctions among emotions because they look for the minimum number of dimensions for emotion differentiation. Moreover, the dimensional models miss interesting features of emotions when several emotions fall extremely close on the affective space [65]. These emotions occupy a small space and may be indistinguishable in the affective space, but easily distinguishable with characteristic features.

Besides the above mentioned models of emotions, Olteanu [78] considers another group of theories: anatomic theories, which try to recreate the neural links and processes that underlie organism's emotions. However, to the best of the author's knowledge, this has not been implemented in robots yet.

The role of emotions

One of the controversial aspects of some of these works is that some authors claim that they implement all or the main functions of emotions. From the author's point of view, several implemented emotions miss one of the key roles of emotions: the motivational role, that is, the capacity of emotions to incite to act. For example, Esau et al. claim that their implemented emotions are used to control the behavior of the robot MEXI but its emotions are just considered in the control of *expressive behaviors*. Therefore, in relation to the inner robot's state, emotions are used for showing the affective state. Moshkina, in the

TAME architecture, allows emotions to modify the way a behavior is executed, but not what behavior to execute or what goal to pursue. Moreover, the goal's of the Lee's adaptive pet-robot are not affected by emotions at all.

How many emotions?

According to Spinola and Queiroz [146], another important issue related to the implementation of artificial emotions in robots is: How many and which emotions must be selected? Some authors defended the idea of implementing a varying number of emotions, from 3 (Nair) to 12 (Hollinger).

The models with the highest number of emotions correspond to those following a dimensional approach. This is due to the "easy" of defining a new artificial emotion just by delimiting a region in a dimensional space.

One very different point of view is presented by Cañamero in [147]: *"Do not put more emotion in your system than what is required by the complexity of the system-environment interaction"*. Therefore, she suggests to include just the required emotions for the task.

Learning

Several works consider some level of learning in their architectures for different purposes. For example, Velásquez's architecture allows to learn the emotional releasers. However, learning is mainly applied to learn when a behavior must be activated. Blumberg uses its motivational variables as reinforcement signal for learning the situations for each behavior. These signals are independently employed, so the behaviors for each motivational variable are separately learned. This might result on situations where certain behavior is appropriate for certain motivational variable, but rather detrimental for others. In contrast, Gadanho considers a broader measure of *satisfaction* as reinforcement signal: the wellbeing, which depends on all the homeostatic variables and other values. This avoids the potential detrimental effects of Blumberg's approach.

Bio-inspiration

Some of these works do not follow a bio-inspired approach to emotion and they are very task-dependent (e.g. Murphy and Lee). These systems lack generality, and flexibility. In addition, emotions lack its functionality and they are loosely couple with its original reason to exist. However, these works present different applications and contexts that proof the applicability of emotion-inspired systems.

Moreover, most of the works lack some "cognitive" aspects of emotions in animals such as anticipation, appraisal of situations and consequences, control of emotions, or emotion learning (e.g. Cañamero's, Arkin's, or Hollinger's works). This dissertation tackles two of them: appraisal and learning.

3.3.7 Why do robots need emotions?

After reviewing the most relevant works where emotions are considered in robots, many readers perhaps are still asking about their utility. Some researchers are against including artificial emotions in artificial creatures. In 1979, Hofstadter [148] stated that simulation of emotions cannot approach the complexity of human emotions, which arise indirectly from the organization of our minds. However, several authors have expounded their reasons to include artificial emotions in robots besides their importance in the human-robot interaction.

According to Arkin, motivations/emotions provide two potential crucial roles for robotics: survival and interaction [27]. Cañamero considers that emotions, or at least a sub-group of them, are one of the mechanisms founded in biological agents to confront their environment. This creates ease of autonomy and adaptation. For this reason she considers, similarly to Arkin, that it could be useful to exploit this role of emotions to design mechanisms for an autonomous agent [29]. Both researchers believe that emotions significantly enhanced human-robot interaction.

Moreover, Cañamero claims that emotions must be included to build *"better adapted and more life-like creatures"* [66]. In [149], Cañamero lists possible application of emotion to problems of autonomous robots: management of goals, repetitive and inefficient behavior, autonomous learning, and cognitive overload. Moreover, Scheutz proposes twelve potential roles of emotions in agents [150]: action selection, adaptation, social regulation, sensory integration, alarm mechanisms, motivation, goal management, learning, attentional focus, memory control, strategic processing, and self model. Hence, there seems to be many applications where emotion-inspired systems could be beneficial.

In relation to motivations, Cañamero states that motivations have to be integrated in artificial systems to promote decision making, activity selection, and autonomy [66].

On the other hand, Ortony explains that robots need emotions for the same reason as humans do: one of the fundamental functions of emotions is that they are a requisite for establishing long-term memories. The second function is that emotions provide opportunities for learning, from simple forms of reinforcement learning to conscious and complex planning [151].

In the same line, Bellman [28], Fellows [152], and Kelley [153] state that, since emotions allow animals with emotions to survive better than others that lack emotions, robots should be provided with features related to emotions in a functional way.

Picard [154] justifies the use of artificial emotions to mimic living humans and animals, create intelligent machines, and try to understand human emotions.

Finally, Olteanu [78] states that artificial emotions are beneficial for social robots because improve human-robot interaction, gives information to the user (robot's internal state, goal, intentions, etc.), and can drive the behavior. He affirms that emotion-based robot architectures enhance believability and effectiveness of robots.

Minsky summarized all these ideas in just one sentence: *“The question is not whether intelligent machines can have any emotions, but weather machines can be intelligent without any emotions”* [155]. Following this same idea, Alvarado does not question either about the inclusion or not of emotions and motivations into intelligent systems, but how to do so [156].

3.3.8 Differences with the followed approach

This dissertation has been mainly inspired by Cañamero’s, Gadanho’s, and Velásquez’s works. As will be shown in following sections, homeostatic drives related to motivations are employed, as those authors do. In the approach followed in this dissertation, the motivations, and not the behaviors (as referred to in Velásquez’s, Breazeal’s, or Esau’s approaches), compete among each other following the point of view of Cañamero, and the dominant motivation drives the robot’s behavior. Nevertheless, in her approach, the winner motivation has a related behavior that satisfies the associated need. Moreover, a discrete emotional approach is followed and it is considered that the relation between situations and emotions is different for each emotion. Therefore, each emotion requires a particular study to establish this relationship. Following this last point of view, currently, this research focuses on three emotions: happiness, sadness, and fear.

In fact, one of the main differences of this thesis with other motivational decision making methods is that the behaviors are not necessarily previously linked with a need, a motivation, or an emotion. This means that there are no pre-wired motivational or emotional behaviors. Then, the robot will learn by itself, using a reinforcement learning algorithm, which behavior to select in order to satisfy each drive, following the same approach proposed by Gadanho. Therefore, they are not known in advance. In contrast, in Breazeal’s thesis certain behaviors are assigned to certain emotions. Others (Velásquez, Sloman, Esau, Shivashankar, or Esau) propose certain predefined influences between emotions and behaviors. In Cañamero’s works, it is assumed that there is only one behavior able to satisfy one need. This fact can be seen as a disadvantage, since it limits the flexibility of the decision making system. It could happen that several behaviors satisfy the same need. This point of view seems to be more bio-inspired since, in nature, in order to satisfy, for example, hunger, we can eat something but also drinking some water can reduce this need. In other works, behaviors are not just linked to emotions, but to drives too (Esau, Breazeal, Kowalczuk, Lee, and Sevin). This is viewed as putting extra knowledge into the system.

Besides, also emotional releasers are predefined in almost all systems. For example, in Breazeal’s work, there are predefined conditions that elicit different emotions [4]. In this thesis, the robot learns from all available actions the best one in each context. Moreover, the emotional releasers follow a very high-level pattern (e.g. happiness is elicited when the robot’s wellbeing increases), and particular cases are learned by reinforcement learning.

Other difference is that, in the approach followed in this dissertation, the way each emotion is defined in the architecture is different. This means that emotions are not defined as a whole as most authors do. As can be observed, there are two points of view in relation to the role of emotions in the decision making process. Cañamero, Gadanho, Velásquez, and Breazeal used emotions to influence the decision making process, not for selecting the behavior directly according to them. On the contrary, others, such as Hirth et al, Hollinger et al, and Lisseti and Marpaung consider emotions as the central aspect of their decision making system so, in some cases, the behavior is selected according to the current emotional state. In the present dissertation, the role of emotions are not limited to one of them, but both points of view are exploited. On one hand, some emotions are used as the reinforcement function in the learning process, as Gadanho also proposed, not determining directly the action selection. On the other hand, other emotions are defined as motivations so, the behaviors will be completely oriented to cope with the situation that generated those emotions. Hence, drives and emotions are not considered as different motivation systems, as Breazeal proposes. Both are integrated into a unique motivation system where the motivational aspects of some emotions and “physiological” needs are considered in a similar manner, in relation to motivational aspects.

3.4 Summary

In the first part of this chapter, the most relevant social robots have been shown and commented according to their characteristics and functionalities. This thesis has been developed in a different robotics platform: the social robot Maggie, which is detailed in Chapter 5.

Following, the most important control architectures and interesting applications where motivations and emotions shape the robot’s behavior have been analyzed. The foregoing overview did not aim to be an exhaustive record about the role of emotions in robots found in the literature, but a brief summary of the most important emotion inspired methods used to tackle the decision making in robots. As mention in the last section, several of these works have served as inspiration for this thesis. Furthermore, in the last section, the main differences with previous works have been remarked.

Next chapter presents the details of the approach followed in this thesis.

CHAPTER 4

The decision making system

4.1 Introduction

One of the main goals stated in Chapter 1 corresponds to increase the robot autonomy by means of a decision making system (from now on referred as DMS) based on the ideas presented in Chapter 2. This chapter presents the theoretical principles of the DMS which is implemented in a robot. As mentioned, it is composed by drives, motivations, emotions, and self-learning. Following, the bio-inspired motivational DMS is introduced (Section 4.2). Later, the principles and concepts of the self-learning process are exposed (Section 4.3). Finally, the emotions involved in the decision making process are analyzed (Section 4.4).

4.2 A motivational decision making system for a social robot

In this thesis, a DMS for a social robot based on motivations, where no specific goals are given in advance, is implemented. The objective of the robot is to *feel good*, in the sense that it has to keep its needs within an acceptable range. Nevertheless, the way to achieve this goal is not defined.

In this DMS, the autonomous robot has certain needs (drives) and motivations. The goal is to survive by maintaining all its drives satisfied. For this purpose, the robot must learn to select the right action in every state in order to maximize its wellbeing. The wellbeing of the robot is defined as a function of its drives in the next section.

The decision making system presented in this section was initially designed by Malfaz [49]. This model was tested on virtual agents [157, 42, 158], and in this thesis it is adapted to a real robotic platform living in a laboratory.

First, considering the ideas presented in Chapter 2, the concepts of drive and motivation in the proposed system are introduced. As mentioned in Chapter 2, **drives** indicates a deficiency or a demand that causes the desire to satisfy this demand or to overcome the deficiency. Such a demand usually motivates and evokes action for its satisfaction. So, drives are often viewed as homeostatic processes that motivate actions in order to reach and keep a certain balance [6]. Recalling, the term homeostasis means maintaining a stable internal state [53]. Then, the robot's internal state is configured 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: the drive. These drives constitute urges to act based on bodily needs related to self-sufficiency and survival [124]. In this approach, the drives are considered as the needs of the robot. The ideal value for a drive is zero, which corresponds to the lack of need. As time goes, the drive increases until it is reduce or satiated (reset to zero).

Motivations are those internal factors, rather than external ones, that urge the organism to take action [59]. Following the ideas of Hull [60] and Balkenius [159] [160], the intensities of the motivations of the robot are modeled as a function of its drives and some external stimuli. The motivational states represent tendencies to behave in particular ways as a consequence of internal (drives) and external factors (incentive stimuli) [161]. In other words, the motivational state is a tendency to correct the error, i.e. the drive, through the execution of behaviors.

In order to model the motivations of the robot, the Lorentz's hydraulic model of motivations is used as an inspiration [162]. In Lorenz's model, the internal drive strength interacts with the external stimulus strength. **External stimuli** are perceptions coming from the environment that alter the tendency to act, that is, the motivations to behave in one way or another. For example, in animals, the smell of a tasty food increases the motivation to eat. Therefore, if the drive is low, then a strong stimulus is needed to trigger a motivated behavior. If the drive is high, then a mild stimulus is sufficient [53]. If the drive or the stimuli separately are strong enough, a behavior can be induced without the influence of the other. The general idea is that we are motivated to eat when we are hungry and also when we have food in front of us, although we do not really need it. In nature, a weak stimulus (e.g. spoiled food) but a strong motivation (e.g. starving) may result in the same behavior as a strong stimulus (e.g. chocolate cake) but weak motivation (e.g. full stomach) [136]. Therefore, the intensities of the motivations are calculated as shown in Equation 4.1

$$\begin{array}{l} \text{If } D_i < L_d \text{ then } M_i = 0 \\ \text{If } D_i \geq L_d \text{ then } M_i = D_i + w_i \end{array} \tag{4.1}$$

where M_i is a particular motivation, D_i is the related drive, w_i corresponds to the related external stimuli, and L_d is called the activation level. Motivations whose drives are below their respective activation levels will not be able to lead the robot's behavior.

According to Balkenius [159, 160], all excited motivational states cannot be allowed to direct the robot at once since this would generate incoherent behaviors. In his opinion, this problem cannot be handled solely by behavioral competition but must be resolved at an earlier stage of processing. The solution proposed is a motivational competition, as Cañamero also proposed in [124]. Therefore, in this approach, once the intensity of each motivation is calculated, they compete among themselves for being the dominant one. The motivation with the highest value, and which drive is over its activation level (Equation 4.1), is considered the **dominant motivation**, and it determines the internal state of the robot. If the drive is below the activation level, it does not compete for being the dominant motivation.

When none of the drives is greater than its activation level L_d, it happens that there is not a dominant motivation. This occurs when all drives are satisfied or, at least, their values are close to their initial values of zero. This implies that the robot's wellbeing is very high, close to the ideal wellbeing (Section 4.3.2). The lack of dominant motivation means that all needs are not high enough to induce the robot to act, so it is in a pleasant state. This is interpreted in such a way that a particular behavior that reduces the drive related to the dominant motivations is not necessary.

The state of the robot is a combination of the inner and external state. The inner state, as has just been explained, is determined by the dominant motivation of the robot. The external state is defined by its relation to every object in its environment (detailed information about how the state is formed can be found in Chapter 6). 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. In humans, for example, we can not eat if we do not have food. It is important to note that initially the robot does not necessarily know the consequences of its actions, nor the reinforcement that it will receive. For instance, the robot does not know that after recharging its batteries, its level of energy will be high. The robot just has the knowledge about which actions can be executed in every state.

In this DMS, there are not predefined, motivational behaviors. This means that the robot does not necessary know in advance which actions to select in order to satisfy the drive related to the dominant motivation. There is a repertory of actions and they can be executed depending on the relation of the robot with its environment, i.e. the external state. For example, the robot will be able to interact with people as long as it is accompanied by someone, or it cannot turn the music player on if the robot is far from it. Through the learning process, the robot learns what action is the best in every situation.

4.3 Learning in the DMS

One of the aims of this DMS is to provide the robot with a mechanism for learning how to behave in order to maintain its needs within an acceptable range. That is to say, as mentioned before, the robot must learn to keep its wellbeing as high as possible. For this purpose, it uses reinforcement learning (Chapter 1) to learn from its bad and good experiences. Following, reinforcement learning (RL) is introduced and the well-known Q-Learning algorithm is summarized in order to provide a better understanding of the learning process.

4.3.1 Reinforcement Learning

In a decision making process, the main concern is related to the decision of which action to take as a function of the available state. By means of RL, the robot learns what to do so as to maximize the reward. Then, it maps states to the actions that are the best in those situations. This map is called the policy.

The decision making loop for an agent in a RL framework is shown in Figure 4.1 [5]: at time t and in a certain state (s_t), it executes an action (a_t) leading it to a new state (s_{t+1}). As a consequence, the environment responds with a reward (r_{t+1}). From that new state the agent executes another action and so on.

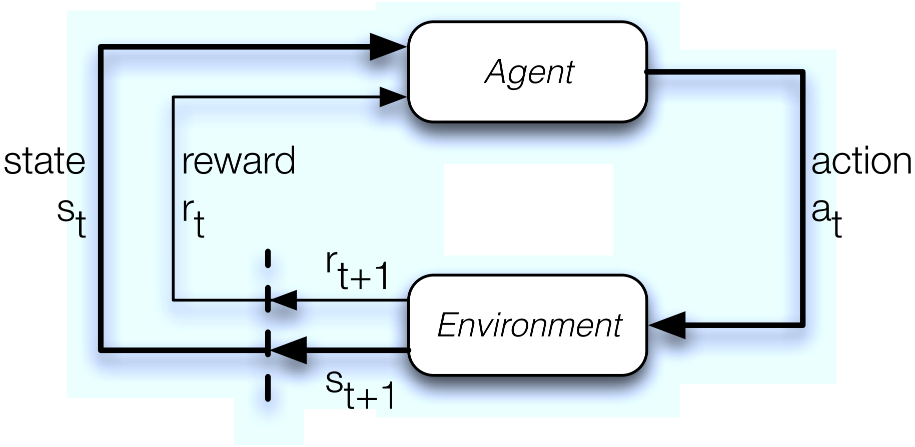


Figure 4.1: Typical iteration in a reinforcement learning context

The reward informs about the suitability of an action in a particular state (how good the action has been in the current state) [163]. The *value* evaluates the long run and it is

[5] This figure was originally presented in [163]

defined as the discounted sum of all the expected reinforcements:

$$value = r_1 + \gamma \cdot r_2 + \gamma^2 \cdot r_3 + \gamma^3 \cdot r_4 + ... \tag{4.2}$$

The parameter γ ($0 < \gamma < 1$) is known as the discount factor, and defines how much expected future rewards affect decision now. The goal of RL is to maximize the total expected reward [164].

RL is a goal-directed learning because the reward function defines the goal. Rewards, more specifically, reward functions in RL, determine the problem the learning agent is trying to solve. RL algorithms address the problem of how a behaving agent can learn to approximate an optimal behavioral strategy, called a policy, while interacting directly with its environment. Roughly speaking, an optimal policy is one that maximizes a measure of the total amount of reward the agent expects to accumulate over its lifetime, where reward is delivered to the agent over time via a scalar-valued signal. Then, this type of learning allows the agent to adapt to the environment through the development of a policy that determines the most suitable action in each state.

RL allows an agent to learn behavior through trial and error interactions with a dynamic environment, i.e. the agent learns from its own experiences how to behave in order to fulfill a certain goal. An agent is connected with its environment via perception and action and they continuously interact. On each iteration the agent receives information about the state s of the environment. Then, the agent chooses an action a and executes it. The action changes the state of the environment. Finally, the agent receives a reinforcement signal r which gives an idea about how well action a performances from state s. The goal of the agent is to find a policy, mapping states to actions, that maximizes some long-run measure of reinforcement [165]. Therefore, the behavior of the agent should choose actions that tend to increase the long-run sum values of the reinforcement signal.

In the case of a robot, pairs formed by the state of the robot and an action (s, a) have an associated value which represents the utility of that action in that state for the robot. These values will be tuned by interaction between the robot and the environment during the learning process (details about the particular learning algorithm can be read in Chapter 6). Then, the autonomous robot learns, from scratch or using some a priori information, the proper behavior to select in every state through its interaction with the environment.

One of the key points in RL is the trade-off between **exploration** and **exploitation**. This refers to how the next action to execute is selected. When a RL agent wants to obtain the highest reward, it chooses the already tried actions which produces the highest reward. But, in order to identify these actions, it has to try unknown actions. That is to say, the agent has to *explore* new actions to *exploit* later the best ones [163].

The Markov property in RL

In RL environments, the decisions and values are functions of the current state. Then, the states must provide enough information to make a good decision. When a state space retains all relevant data it is said that it has the Markov property [163]. This implies that all that is relevant for the future is kept in the state. In other words, the current state must include all relevant data observed in past experiences.

In a general RL case, the response from the environment, at a certain moment, depends on all what has happened before. If the Markov property is present, the response just depends on the last state and action. Mathematically, it is defined in Equation (4.3) and it is graphically presented in Figure 4.2.

$$Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t\} \tag{4.3}$$

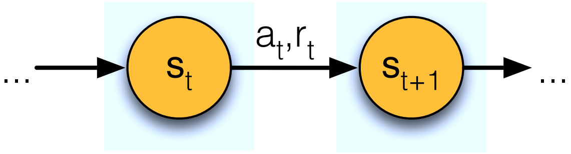


Figure 4.2: A Markovian RL problem

In spite of the Markov property seems to be a requirement for solving RL problems, Sutton and Barto do not think that it is a must: *“Even when the state signal is non-Markov, it is still appropriate to think of the state in reinforcement learning as an approximation to a Markov state”* [163].

Markov states provide an excellent support for predicting future rewards and selecting the most appropriate actions. Then, the closer the state is to the Markov property, the better results from RL systems are obtained. In conclusion, RL algorithms can be successfully applied to problems with states that do not strictly fulfill the Markov property.

The same situation can be observed in humans as well. Humans are able to make correct decision even if they do not have all the information. For example, imaging you desire to forecast the tomorrow’s weather. In this case, the state is determined by all the relevant information that you have ever observed about the weather (the current period of the year, the current temperature, the color of the sky, the weather during that week, the wind, the humidity, etc) and how it influences the tomorrow’s weather. In practice, this is far to much to remember and analyze, and much of these data will not be considered in your forecast. However, some people are very proficient at weather forecast, even if they do not have access to a perfect Markov state representation.

In this thesis, the internal state of the robot just considers the dominant motivation, but not the others. Therefore, the final state (internal and external) is not fully Markov because the rest motivations are not represented. This lack of representation would cause violations of the Markov property. However, following the explanation exposes in the last paragraphs, the Q-Learning algorithm is used for learning the proper policy and the results obtained have been successful, as presented in Chapter 9.

The Q-Learning algorithm

RL has been successfully implemented in several virtual agents and robots [166, 167, 168, 169, 170, 171]. One of the main applications, for robots or agents, is the learning of complex behaviors as a sequence of basic behaviors. Those complex behaviors allow to optimize the adaptation of the agent or robot to its environment. The reinforcement learning algorithm named Q-learning [172] has become one of the methods that is most used in autonomous robots [173, 174, 175, 176]. Actually, the learning algorithm implemented in this thesis is a variation of the Q-Learning (Chapter 6).

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 [48]. Every $Q(s, a)$ is updated according to:

$$Q(s,a) = (1-\alpha) \cdot Q(s,a) + \alpha \cdot (r + \gamma V(s')) \tag{4.4}$$

where:

$$V(s') = \max_{a \in A} (Q(s', a)) \tag{4.5}$$

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 represents a single 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 value estimate [164]. This parameter gives more or less importance to the learned 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 ($0 < \gamma < 1$), the discount factor, defines how much expected future rewards affect decision now (it was introduced in Equation (4.2)). 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 [164].

A policy π defines the behavior of the agent. It defines a map $\pi : S \longrightarrow \Pi(A)$ from state s and action a ($a \in A(s)$), to the probability of taking action a when in state s. This value corresponds to the expected return when starting in s and following the policy thereafter [163].

As previously said, the final goal of the agent is to learn the optimal policy, the one that maximize the total expected reward. This is a deterministic policy that 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:

$$\pi^*(s) = \arg\max_a Q(s,a) \tag{4.6}$$

In practice, the optimal policies rarely happen. The required extreme computational cost and memory are a relevant constrain. Therefore, approximate optimal policies are the goal.

RL, optionally, can consider models of the environment. These models are replicas of the behavior of the environment and they are used by some RL methods (e.g. dynamic programming) for state-space planning. Its utility is limited because of its computational cost and the assumption of perfect models [163]. However, the Q-Learning algorithm follows a model-free approach because the system knows neither the consequences of executing an action (the next state) nor the reward that will be obtained. It just knows the actions that can be executed with each object.

4.3.2 The robot's wellbeing

As previously said, RL requires a reward function which determines the goal. As said before, the objective is to keep the robot's needs as low as possible. Therefore, the reward function is related to the its wellbeing.

In this implementation, based on the drive reduction theory, which states that the drive reduction is the chief mechanism of reward [60], 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:

$$Wb = Wb_{ideal} - \sum_i \alpha_i \cdot D_i, \tag{4.7}$$

where α_i are the ponder factors that weight the importance of each drive on the wellbeing of the robot. Wb_{ideal} is the ideal value of the wellbeing which corresponds to the value of 100. It is easy to observe that as the values of the needs of the robot (the drives) increase, its wellbeing decreases. Thus, drives are inversely proportional to wellbeing: the lower the drives are, the higher the wellbeing is. Therefore, the reward value for one action executed in certain state corresponds to the variation of all the drives during its execution, that is, the wellbeing variation. For example, for an action a, the reward is computed according to Equation 4.8.

$$reward_a = \Delta Wb_a = Wb_{after\,a} - Wb_{before\,a} \tag{4.8}$$

Considering Equations (4.8) and (4.7), the total reward for an action depends on how fast drives change their values during the execution of that action. Moreover, the positive/negative variation of the wellbeing is interpreted as *happiness/sadness* in the learning algorithm (Section 4.4.1).

At the beginning of the learning process, the values for all actions can be set to the same number. This means that no knowledge is provided in advance, so there is not relevant information about the action selection; i.e. there are 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, the robot should not *die*, i.e. battery is depleted, so the knowledge to survive can be initially predefined. However, in this work, the former has been applied. This implies that, if the robot *needs* energy, the robot could run out of battery since it does not know yet what to do in that particular case. In order to learn it, the robot has to try different strategies which can success or fail.

It is important to note again, that the actions are not related to the motivations. This means that the robot does not know in advance that, for example, it must recharge its battery in order to satisfy its need of energy.

In short, the decision making process is cyclic and it can be described in the following steps:

1. Update of the drives and the motivation intensities.
2. Motivation competition and selection of the inner state (the dominant motivation).
3. Determine the external state.
4. Evaluation of possible actions
5. Execution of one action.
6. Update of the wellbeing function.
7. Generation of the reinforcement function (*happiness/sadness*).
8. State-action evaluation (RL).

In every loop, the DMS needs data from the environment in order to update the robot's state. These data is provided by the robot's control architecture. In the way around, the DMS communicates the proper action to execute. Therefore, a two-way communication between the DMS and the control architecture is required (Figure 4.3).

An overview of the decision making process and its elements can be seen in Figure 4.4. Drives and, by extension, motivations determine the internal state. This internal state together with the external state determine the state which is used to make a decision. After

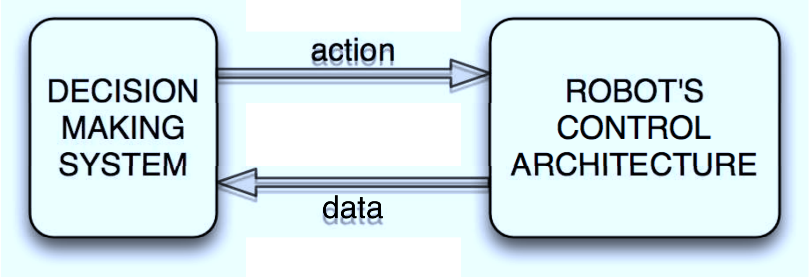


Figure 4.3: Communication between the DMS and the robot's control architecture.

an action is selected and executed, its consequences affect to the world where the robot is "living" and to its drives. Thus, the wellbeing is affected and it is used as the rating to evaluate the performance or suitability of an action in a state (learning process). This experience will be considered in future decision making.

At this dissertation, a real robot learns from scratch the best possible actions at each world configuration (the dominant motivation plus the state related to each object). The tuples formed by the dominant motivation, the objects, the state related to these objects, the feasible actions, and the values of these actions decide the next action to be selected.

4.4 Considered emotions

The presented DMS considers different emotions with different functions. This section introduces them and presents how they are included in the DMS.

In relation to artificial emotions, their functionality in agents is quite diverse. In this approach *happiness* and *sadness* are used as the reinforcement function in the learning process. Besides, *fear* motivates behaviors oriented towards self-protection.

As introduced in Chapter 2 (Section 2.4.5), it is believed that emotions are elicited from the subjective appraisal of the environment of the agent. Moreover, a discrete approach for generating the artificial emotions (*happiness*, *sadness*, and *fear*) is followed in this work.

Before going into the artificial emotions in robots, two ideas must be clarified:

1. According to Damasio [89], the impact of emotions in human mind depends on the feelings induced by the emotions. He states that the full and lasting impact of feelings, and by extension of emotions, requires consciousness. Thus far, robots (or any other artificial creature) do not have consciousness so robots can not *feel* emotions. Moreover, Castelfranchi [68] affirms that since nowadays it is not clear what *feel* means, it is not correct to say that robots *feel* emotions. Consequently, it can be said that robots **have artificial emotions**.

2. Throughout this text, many times emotions are referred to robots or agents. In these

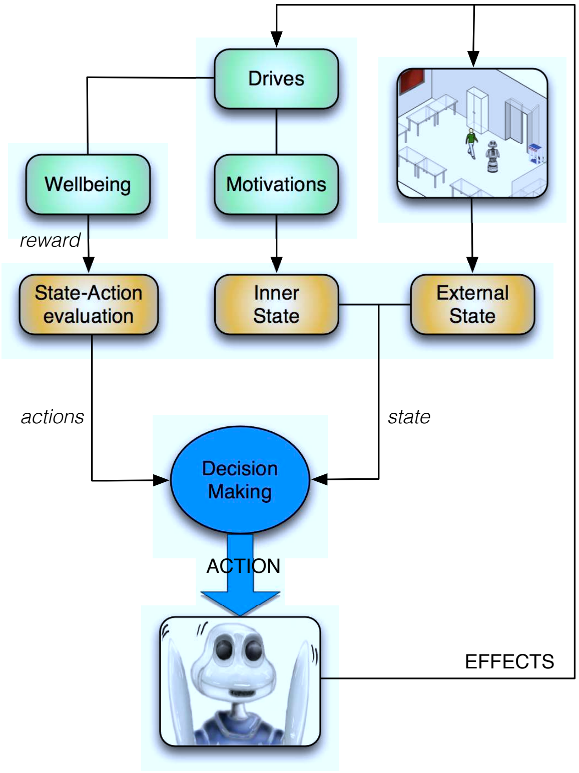


Figure 4.4: The DMS and how its elements are related each other.

situations, *emotion* means indeed *artificial emotion*. Therefore, when talking about artificial systems, the word *emotion* can be used as a shortcut of *artificial emotion*, and they have been interchangeably used.

Following, the emotions involved in this work are presented in detail. Initially, the roles of each artificial emotion is commented. Later, how they are individually generated is exposed.

4.4.1 Happiness/sadness

The role of happiness and sadness

As shown in Figure 4.5, *happiness* and *sadness* are used in the learning process as the reinforcement function and, as just presented, they are related to the robot's wellbeing.

The role of *happiness* and *sadness* as the reinforcement function was inspired by Gadanho's works, as shown in Section 3.3, but also by Rolls [177]. He proposes that emotions are states elicited by reinforcements (rewards or punishments), so our actions are oriented

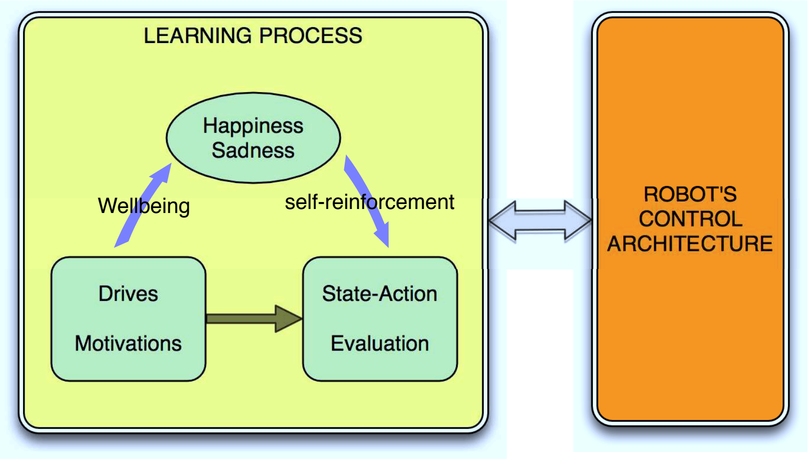


Figure 4.5: The role of happiness and sadness in the DMS

to obtaining rewards and avoiding punishments. Following this point of view, in this proposed DMS, *happiness* and *sadness* are used as the positive and negative reinforcement functions during the learning process respectively. Moreover, this approach seems consistent with the drive reduction theory introduced in Section 2.3.1 where, according to this theory, the drive reduction is the chief mechanism of reward.

According to Starzyk [178], this learning process is a kind of Motivated Learning because it uses internal reward signals which are related to abstract motivations and goals (*happiness* and *sadness* are related with the robot's wellbeing which refers to the robot's drives). Then, the actions are evaluated considering how well these actions satisfy the internal goals.

The appraisal process

In order to define *happiness* and *sadness*, the definition of emotion given by Ortony [179] is taken into account. In his opinion, emotions occur due to an appraised reaction (positive or negative) to events. According to this point of view, in [83], Ortony proposes that *happiness* occurs because something good happens to the agent. On the contrary, *sadness* appears when something bad happens. Moreover, according to Esau [6], the satisfaction of a drive is usually accompanied by positive emotions reflecting the associated comfortable feeling. On the contrary, if a drive is not satisfied over a longer period of time, negative emotions can often arise. These positive and negative emotions can be related to the *happiness* and *sadness* emotions.

In the proposed system, *happiness* is related to a reduction of a drive (e.g., a positive reaction because the robot recharges its battery) and *sadness* to an increment of a drive (e.g. a negative reaction because the robot was damaged by a user). Taking into account that the wellbeing of the agent is a function of its drives (Equation (4.7)), *happiness* and *sadness* are related to the positive and negative variations of the robot's wellbeing (ΔWb). In a formal way, *happiness* and *sadness* are defined by Equation 4.9.

$$\begin{aligned} \Delta Wb = Wb_{t+1} - Wb_t > 0 \Rightarrow happiness \\ \Delta Wb = Wb_{t+1} - Wb_t < 0 \Rightarrow sadness \end{aligned} \tag{4.9}$$

It is important to note that low wellbeing does not imply sadness as well as high wellbeing does not implies happiness. These emotions are related to increments or decrements of the wellbeing (positive and negative reactions). This means that, for example,a person having fun when he is starving would imply happiness. The intensity of these emotions is proportional to the variation suffered by the wellbeing.

Using this approach, every event or situation that produces a positive or negative appraisal of the environment (internal and external) of the robot is considered as *happiness* or *sadness*. It is worth mentioning that there is not a fixed set of situations that elicits *happiness* or *sadness*, but the robot evaluates all situations and pursues happy situations while avoiding sad ones. This approach seems similar to the natural one.

4.4.2 Fear

The role of fear

The emotion of *fear*, based on some theories that state that emotions can motivate behaviors [26, 31, 127], is defined as a motivation. Therefore, according to the proposed decision making process, *fear* could be the dominant motivation and, in that case, the robot would be "scared". When this happens, the robot must learn the right action to execute in order to cope with the situation that caused this inner state.

The role of the artificial emotion of *fear* is inspired by the idea that emotions can also constitute motivational factors and constitute "value systems" that affect the selection of goals and goal-directed behavior [127]. Another point of view is given by Arkin [27], who says that emotions constitute a subset of motivations which gives support to the survival of an agent in a complex environment.

Moreover, Breazeal [31] also states that emotions are an important motivational system for complex systems. In fact, according to her [134], the unique function of fear is to motivate avoidance or escape from dangerous situations. This response protects the robot from possible harm when it is faced with a threatening stimulus. This is, in fact, the approach that is followed in this work.

The appraisal process

According to Ortony [83], fear is a negative reaction related with the possibility of something bad happening. In this approach, the possibility of something bad happening means that the wellbeing of the robot may decrease (a need or drive may be increased).

Fear is normally associated with avoiding dangerous situations. Those situations could be considered as situations were something bad could happen to the robot, but it does not have any control over it.

The fear emotion can be considered as an adaptive response to threatening situations [50]. As commented in Section 2.4.5, some of these threatening situations are innately identified as dangerous, but others are learned. In this work, the attention is aimed at learned releasers of fear. It is important to note that, in the approach presented in this dissertation, the appraisal of a dangerous situation is based on an automatic process using associative learning. As will be shown farther ahead, the robot, using RL, is able to identify dangerous situations without using any deliberative mechanism. Therefore, this section exposes how a dangerous state is detected (appraisal of fear) following a learning process.

The idea is similar to what happens when a person kicks us for no reason. Since this fact causes an intense emotional experience, even if that person has just sporadically hit us, we remember this situation and the consequential pain for long time. Therefore, whenever that person is close to us, we relive this situation and evaluate the possible consequences. The final result is that we are afraid of that person. Another example could be observed when a person is afraid of thunders during a storm because of an unpleasant situation in her childhood. That person is afraid whenever he is facing a storm, and this afraid is not under his control.

Then, threatening situations, or **dangerous states**, are those where the robot can be significantly damaged. This damage is caused by the effects of actions external to the robot, so it is not responsible of them. These external actions can be originated by other individuals (e.g. the abuser) or even environmental circumstances (e.g. the storm). Hence, in order to prove the practical use of *fear*, the robot's environment has to be able to affect the robot's drives, by other's actions or due to circumstances coming from the environment itself. These external actions are called **exogenous actions**, and the objects capable of executing actions by themselves are referred as **active objects**.

Exogenous actions alter the robot and its environment but they are not under the robot's control. Then, they affect the situation of the robot and the reinforcement received. Exogenous actions lead to complex domains where, from the robot's point of view, they can produce unwanted state transitions. Many times, just their effects are perceived or observed, but not the actions themselves. Then, these domains are quite hard to model because of the difficult to foresee them.

Following the example of the person who hit us before, we do not have any control over that action. The received punishment is not due to any of our actions, but it depends on the other person. For example, considering our abuser, if we are walking and a person hits us, we suffer pain but, in the case of using reinforcement learning, we would not know if this negative reward, the pain, is because of the walking or because a person hit us. Then, the effects of the hit are mixed with our actions. In the robot, the exogenous actions (and their effects) are mixed with the actions executed by the robot and their effects. Therefore, in a reinforcement learning framework, the reward of an action executed by the robot could be altered by an exogenous action. Consequently, a key issue is to undoubtedly identify the effects caused by the actions of the robot and the effects of the exogenous actions.

In order to distinguish between the effects from the robot's actions and the effects from the exogenous actions, the exogenous actions are just considered when the robot is doing nothing. More precisely, the exogenous actions are considered during the execution of a robot's action without any kind of effects, so the resulting effects during its execution can be certainly assigned to the exogenous actions. In this situation, all modifications in the robot's wellbeing, as well as in the world configuration, is due to the exogenous actions. These kind of robot's actions are represented as a_{exog}.

Then, in this approach, *fear* appears when the robot is in this kind of situations that are considered as "dangerous". This means that the appraisal of these situations is the elicitor of the *fear* emotion. In this dissertation, elicitors of *fear* are not given a priori but learned.

Three different processes are involved in the generation of fear:

- Storing the worst experiences
- Detecting new dangerous states
- Updating the fear motivation

Some of these processes can occur in parallel.

Storing the worst experiences As mentioned above, dangerous states are those situations where other agents have caused a considerable damage can be caused to the robot. These dangerous states are used as the releasers of the emotion of *fear*. These releasers can be pre-defined by the programmer, what corresponds to innate releasers of fear in living beings. Conversely, in this work, they are learned after an appraisal of the situation, following the previously mentioned appraisal theory.

Usually, dangerous states correspond to situations where the robot is not usually damaged but some adverse exogenous actions sporadically cause harm to the robot. An adverse exogenous action provokes a considerable decay on the robot's wellbeing. When the "harm" caused to the robot in a certain state is greater than a certain threshold, it is appraised as a dangerous state and *fear* will emerge every time the robot transits to this state. For this reason, in order to identify the dangerous states, the worst experience in every state is cached. That is, the worst Q values for each state must be stored in order to remember those worst experiences. This is similar to animals which remember their worst experiences and relive them when they are facing the same situation.

For all the above reasons and taking into account the definition of dangerous state, the worst Q values are computed for the robot's actions where exogenous actions can be considered, i.e. a_{exog}. For this reason, in addition to the values of the actions, the Q_{worst} values have to be stored too. They are computed according to the next equation for each iteration:

$$Q^{obj_i}_{worst}(s, a_{exog}) = min(Q^{obj_i}_{worst}(s, a_{exog}), r + \gamma \cdot V^{obj_i}_{worst}(s')) \tag{4.10}$$

where a_{exog} is an action related to the object obj_i ($a_{exog} \in A_{obj_i}$) in state s and, during its execution, the effects of exogenous action can be undoubtedly measured (that is, it is an "effectless" robot's action); the resulting state is s', r is the reward corresponding to the variation of the robot's wellbeing, and γ is the discount factor. $V^{obj_i}_{worst}(s')$ means the best $Q^{obj_i}_{worst}$ value from the new state and it corresponds to:

$$V^{obj_i}_{worst}(s') = \max_{a \in A^{obj_i}_{exog}} (Q^{obj_i}_{worst}(s', a)) \tag{4.11}$$

$V^{obj_i}_{worst}(s')$ computes the best possible action among the $Q^{obj_i}_{worst}$ values from the state s'. In other words, it stores the value of the least harmful action from the new state.

The states considered for the appraisal of *fear* just correspond to the external state of the robot. This is the state related to the objects in the world. This is because, considering the definition given at the beginning of this section of dangerous state, a state is dangerous independently of the internal state. For example, in humans, if you are afraid of spiders, you will experience fear if you see a spider, independently of any internal need; i.e. it does not matter if a person suffering arachnophobia is hungry or thirsty, he is terrified when he sees a spider. Likewise, the states during the appraisal of *fear* are just related to the objects in the world.

Moreover, when an active object harms the robot, the damage is just due to the actions of the active object itself. Therefore, the appraisal of *fear* considers the state of each active object individually.

The proposed approach agrees with Olteanu [180], who states that the evaluation of internal and external situation is a crucial process for the appraisal of emotion. Here, the variation of robot's wellbeing and external state of the robot are involved in the appraisal of *fear*.

Detecting new dangerous states The above computed Q_{worst} values are used to identify the dangerous states. These dangerous states are recognized by the robot itself, so they are not pre-programmed in advance.

A state s is considered as a dangerous situation when there is a $Q^{obj_i}_{worst}(s, a_{exog})$ value which is below a certain threshold: L_{danger}. The contrary is considered as a safe state.

Mathematically, it is expressed as:

$$\begin{array}{ll} If\ Q^{obj_i}_{worst}(s, a_{exog}) < L_{danger} \Rightarrow & \text{s}\ is\ a\ dangerous\ state; \forall s \in S_{obj_i},\ \forall a \in A^{obj_i}_{exog} \\ If\ Q^{obj_i}_{worst}(s, a_{exog}) \geq L_{danger} \Rightarrow & \text{s}\ is\ a\ safe\ state; \forall s \in S_{obj_i},\ \forall a \in A^{obj_i}_{exog} \end{array} \tag{4.12}$$

where S_{obj_i} is the set of all possible states related to object i, and $A^{obj_i}_{exog}$ is the set of all "effectless" actions related to object i.

Updating the fear motivation As explained before, in this work, fear is considered as a motivation which is able to govern the robot's behavior. Once the dangerous states are identified, the *fear* motivation is able to be the dominant one and to lead the robot's actions. Whenever the robot transits to a dangerous state, *fear* emerges. In a formal way, if s is the current robot's state, the fear value is updated according to the next equation.

$$\begin{array}{l} \text{If } s \text{ is a dangerous state} \Rightarrow \text{Fear} = high \\ \text{If } s \text{ is a safe state} \Rightarrow \text{Fear} = low \end{array} \tag{4.13}$$

High and low values of *fear* correspond to the presence and to the absent of *fear* respectively.

It is specially worth mentioning that the learning process of dangerous states is different to the learning process of action selection in the DMS. The later also might be useful for dealing with dangerous states under certain conditions. The learning process for selecting actions provides a mechanism to correctly react to situations where the robot is commonly damaged. In these situations the behaviors to avoid states that harm is recurrently provoked from can be directly learned by reinforcement learning since their expected values are low. This means, that actions which could lead to these low value states will be rarely selected.

However, sporadic harm from a particular state cannot be managed in the same manner. States where the robot is sporadically damaged can not be tackled by a learning process based on traditional RL. Using regular RL algorithms in sporadic harmful situations results in that the utility value of these states is still high. Therefore, this causes that the robot does not learn to avoid these situations.

The proposed mechanism for the appraisal of fear has been specifically designed to consider both circumstances and it perfectly deals with all dangerous situations. Then, considering the worst experiences perfectly works for learning both situations where damage is frequently as well as sporadically caused.

4.5 Summary

This chapter has established the main ideas for the DMS considered in this thesis. The use of drives and motivations for generating non-predefined motivational behaviors has been justified. Moreover, the definition of the robot's wellbeing and its use in the reinforcement function during the learning process perfectly fits with the drive-reduction theory which has inspired this work.

Besides, Section 4.3 has shown how the RL approach perfectly covers the requisites established in this dissertation. In particular, the Q-Learning algorithm has been deeply analyzed because it is the based of the learning algorithm implemented in the robot (Chapter 6).

In the last section, the emotions considered in the DMS have been presented. Emotions have been included in the system but with different functionalities: *happiness* and *sadness* are related to the wellbeing variation, and *fear* is considered as a motivation. The appraisal processes for these emotions have been detailed.

CHAPTER 5

The social robot Maggie and its decision making system

5.1 Introduction

This chapter presents the experimental platform where the proposed decision making system has been implemented. It is a social robot which has been used to test the performance of emotions, motivations, and drives in a real environment.

Initially, the robotic platform is briefly introduced, its purpose and its sensory-motor capacities are detailed. Later, the control architecture running in the robot is presented (Section 5.3). Finally, the DMS is featured according to the robot designers' desires (Section 5.4). Details about how it interacts with the architecture are provided too.

5.2 The robot Maggie

The presented work has been implemented in the research robotic platform named Maggie [181]. Maggie is a social and personal robot intended to perform research on human-robot interaction and improving robots autonomy (Figure 5.1). It was conceived for personal assistance, for entertainment, to help handicapped people, to keep people accompanied, etc. Its external friendly look facilitates its social robot task. Both software and hardware have been developed by the Robotics Lab research group from Carlos III University of Madrid.

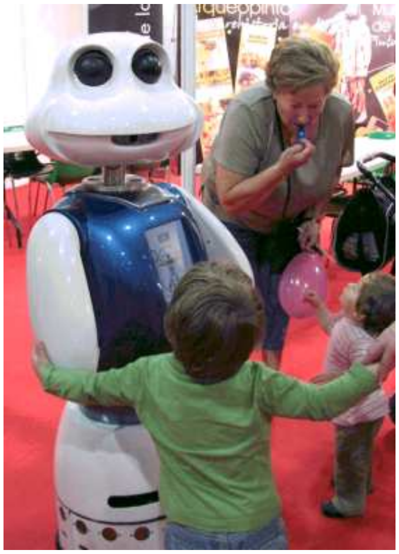


Figure 5.1: The social robot Maggie interacting with children

In relation to its hardware, Maggie is a computer-controlled system built on a wheel base which allows the robot to move through the environment. Its arms, neck, and eyelids movements 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, the robot is equipped with a text-to-speech module and an automatic speech recognition system.

More precisely, Maggie is endowed with 12 touch sensors located in different places on the robot's surface. In the head, two eyes with two mobile eyelids and voiced-synchronized leds in the mouth improve its expressiveness. Moreover, an OBID RFID reader placed inside the head provides low-range capacity for reading passive rfid tags. The neck has a two degrees-of-freedom (dof), pan and tilt, several speakers are around it. The body has two one-dof arms and the infrared device is placed inside the robot's belly, behind a screen which signals can go through. The sensory capacity is extended by another two RFID readers situated in the body that are able to read rfid tags from longer distances. In the base, it has 12 sonar sensors, 12 infrared sensors, 12 bumpers, and a Sick LMS 200 telemeter laser. A differential drive systems moves the robot around. The "brain" in charge of controlling all these components is an on board computer where a Linux system is running.

The required energy for all devices is received from two batteries which provide a power supply of 25 V. During its working life, the robot needs at least 20 V for a correct operation. The purpose is to achieve a robot working continuously in a never-ending working life. This means that the battery should always be over this threshold.

Social robots, such as Maggie, are intended for tasks where humans are very close to the robot and they interact. These users do not have to have technical knowledge or to be used to robots. Therefore, the external appearance is an important issue to arise certain empathy and confidence. Maggie was devised to be attractive with a friendly looking, and it shows a great expressivity by means of leds, voice, and movements. Moreover, different kinds of mechanisms for interaction are combined in a multimodal dialogs.

5.3 The Automatic-Deliberative control architecture

The robot's control architecture has been developed by the Robotics Lab research group [182, 183, 184, 185, 186] and it is named Automatic-Deliberative (AD). This biologically inspired architecture is based on the ideas of the modern psychology expressed by Shiffrin and Schneider [187, 188], so it considers two levels, the automatic and the deliberative, as shown in Figure 5.2. The communication between both levels is bidirectional and it is carried out by Short-Term Memory and events [189].

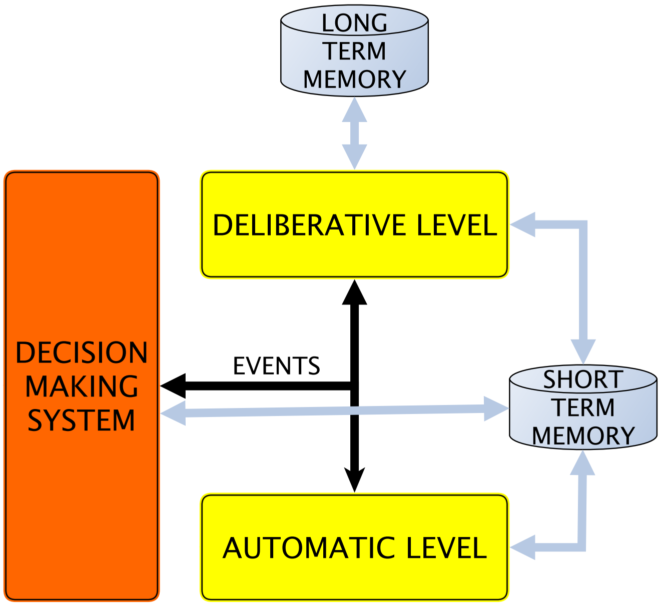


Figure 5.2: The Automatic-Deliberative architecture with the DMS

In the AD architecture [190], both levels are formed by skills, which endow the robot with different sensory and motor capacities, and process information. Skills can be coordinated by sequencers and, previously, the Main Sequencer managed the deliberative skills according to a fixed script where all possible situations that the robot can face are considered. This means that this script has been programmed in advance and it is exclusive for certain objectives. The present thesis replaces the Main Sequencer with a DMS based on drives, motivations, emotions, and self-learning. The details of the implemented DMS are presented in Section 5.4.

The proposed DMS has a bidirectional communication with the rest of the control architecture (Figure 5.2). On the one hand, the DMS selects actions in order to satisfy the most urgent need. These actions are translated into skills (deliberative or automatic), which are activated and blocked by the AD architecture. On the other hand, the DMS needs information from the environment in order to update the state of the robot (internal and external states) and to assess the suitability of the skills activated. This information will be provided by the sensors of the robot, where this data is interpreted by the AD architecture and, then, transferred to the DMS.

5.3.1 Deliberative level

In the natural world, human deliberative activities are characterized by the fact that these are carried out in a conscious form. Moreover, temporal dimension is an important property: deliberative processes require a large quantity of time to be dedicated to the analysis. These activities are carried out sequentially, that is, one after another, and it is not possible to carry out more than one deliberative activity at a time.

In the AD architecture implementation, deliberative skills are based on these activities and only one deliberative skill can be activated at once [183].

5.3.2 Automatic level

Living beings' automatic activities are characterized by the fact that their actions and perceptions are carried out without the necessity of having consciousness of the processes responsible for controlling those activities. Examples of this would be the heart beat, the hand movement when writing, or that of legs when walking. An automatic activity can be carried out in parallel with other automatic activities and with a deliberative activity. For example, a person can be driving a vehicle and maintaining a conversation simultaneously. The level of complexity of automatic activities may be very variable and goes from the simplicity of moving a finger to the complexity of playing a sonata previously memorized in the piano.

In the AD implementation, the automatic level [191] is mainly formed by skills which are related with sensors an actuators. Automatic skills can be performed in a parallel way and they can be merged in order to achieve more complex skills.

5.3.3 AD Communications

Memories

One of the main characteristics of human beings is their ability to acquire and store information from the world and from their own experiences. Memory can be defined as the capacity to recall past experience or information in the present [192].

Based on the memory model proposed by Atkinson and Shiffrin [193], the AD architecture considers two different memories: the *Short-Term Memory* (STM from now on) and the *Long-Term Memory* (LTM), see Figure 5.2.

In this architecture, STM is defined as a temporary memory. This memory is regarded as a working memory where temporal information is shared among processes and skills. The STM is a memory area which can be accessed by different processes, where the most important data is stored. Different data types can be distributed and are available to all elements of the AD architecture. The current and the previous value, as well as the date of the data capture, are stored. Therefore, when writing new data, the previous data is not eliminated, it is stored as a previous version. The STM allows to register and to eliminate data structures, reading and writing particular data, and several skills can share the same data. It is based on the blackboard pattern.

On the other hand, LTM is a permanent repository of durable knowledge. This knowledge can come from learning, from processing the information stored in STM, or it can be given a priori. In the AD architecture this memory refers to a permanent memory where stable information is available only for deliberative skills. The LTM has been implemented as a data base and files which contain information such as data about the world, the skills, and grammars for the automatic speech recognition module.

Events

Events are the mechanism used by the architecture for synchronizing and working in a cooperative way. An event is an asynchronous signal for coordinating processes by being emitted and captured. The design is accomplished by the implementation of the publisher/-subscriber design pattern so that an element that generates events does not know whether these events are received and processed by others or not.

The asynchronous signals are emitted with an attached parameter, an integer, that can be read by the subscribers.

5.3.4 AD Skill

As already stated, the essential component in the AD architecture is the skill [189] and it is located in both levels. In terms of software engineering, a skill is a class hiding data and processes that describes the global behavior of a robot task or action. The core of a skill is the control loop which could be running (the skill is *activated*) or not (the skill is *blocked*).

Skills can be activated by other skills, by a sequencer, or by the decision making system. They can give data or events back to the activating element or other skills interested in them. Skills are characterized by:

- They have three states: ready (just instantiated), activated (running the control loop), and blocked (not running the control loop).
- Three working modes: continuous, periodic, and by events.
- Each skill is a process. Communication among processes is achieved by STM and events.
- A skill represents one or more tasks or a combination of several skills.
- Each skill has to be subscribed at least to an event and it has to define its behavior when this event arises.

The AD architecture allows the generation of complex skills from atomic skills (indivisible skills). Moreover, a skill can be used by different complex skills, and this allows the definition of a flexible architecture.

5.4 Featuring Maggie's DMS

The aim of the presented DMS is to achieve an 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 modeled in order to be processed by the implemented decision making module.

This section presents the configuration of the DMS presented in Section 4.2. Roughly speaking, the DMS setup can be divided in three groups according to the scope of the variables. These groups are:

1. The internal variables of the robot
2. The external world
3. How the next action is selected

These categories will be individually detailed in the next subsections.

All the parameters considered in this implementation shape a specific robot's personality. That is, the DMS setup defines the robot's behavior during its lifespan. Changing these parameters, new personalities or behaviors are exhibited by the robot. The parameters which are presented in the next sections have been defined at design time by the author.

5.4.1 The robot's inner world: what drives and motivations?

This section details all the inner variables which define the robot's behavior.

As expressed by Equation (4.1), each motivation is represented by a value and 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 are not considered for the dominant motivation.

As mentioned, the internal needs, the drives, represent an internal value. Each motivation is connected to a drive. The choice about which drives (and consequently motivations too) must be implemented were made at design time. The number of drives and motivations should be flexible and correlated to the tasks to perform [194, 7]. Therefore, since the system has to be running on a robot intended to interact with people, some social motivation is needed to "push" the robot into human-robot interaction. Moreover, the authors want the robot to be endowed with play-oriented aspects, hence, a recreational nature is required by the robot. In contrast with the need of fun, once in a while, it wants to relax; then, also some kind of rest is desired. Nevertheless, the first primitive drive for all entities is to survive and, in this particular case, it is translated to the need of energy.

All things considered, the selected drives are:

- **Energy:** this drive is necessary for survival.
- **Boredom:** the need of fun or entertainment.
- **Calm:** the need of peace.
- **Loneliness:** this is the lack of social interaction and, then, the need of companion.

All these drives represent the deviation from the ideal state. This ideal state corresponds to the value zero for all drives (no needs).

Since we want Maggie to be an autonomous social robot and considering the defined drives (each motivation is connected to a drive), the motivations that have been considered are:

- **Survival:** it refers to the energy dependence. This motivation is connected to the need of *energy*. Then, the *survival* motivation is the most critical one. This is the major requirement to be achieved by an autonomous robot.

- **Fun:** this motivation is related to entertainment purposes and its associated drive is *boredom*. The motivation of *fun* refers to the need of entertainment of the robot itself. This means that this drive can be satisfied when Maggie is having fun and this is achieved when it is dancing.

- **Relax:** it is linked to a peaceful environment and it is related to the drive of *calm*. In contrast with *fun*, *relax* is its counterpoint: it searches for noiseless conditions.

- **Social:** it corresponds to the need of human-robot interaction. It is associated to the *loneliness* drive. As presented in Chapter 5, Maggie is a social robot so one of its main goals is to stablish relationships with people. This attitude is enforced by this motivation.

- **Fear:** this motivation arises in dangerous situations and it guides the robot towards a secure state. In this case, there is not a drive associated to it.

Some researchers from psychology field could believe that these are not conventional motivations, and they do not should be treated as them. However, in this thesis, they have been considered as motivations because all of them impulse the robot to act.

All motivations have been defined considering that Maggie is a social robot designed to interact with users and move among people. Then, its behaviors have to be as natural as possible, i.e. its behaviors have to be comprehensible by humans sharing the environment with the robot.

The use of fear as a motivation in a robot is one of the cornerstones of this thesis. How it is generated, its appraisal, and the reactions to fear are novel ideas presented in this dissertation. As seen, *fear* is treated in a different way than the other motivations. Fear is considered a motivation but there is not a drive related to it because fear does not represent a deficiency in any physiological need. However, it is able to lead the robot's behavior.

In addition, as said before, when all drives are below their respective activation level, none motivation can be considered as the dominant one. This situation is considered in the proposed system too and consequently an extra motivation, referred as **none** or **non-motivation**, is included. Therefore, the most convenient behavior for this situation will be learned and studied too. This special motivation is related to a special drive, which has a constant value of 1, and its activation level is set to 0. In consequence, this motivation is always ready for becoming the dominant motivation, but it does not represent any need.

Taking human beings as inspiration again, it is not common that all human motivations compete, at the same time, for being the dominant motivation. For example, it is not usual that a person simultaneously needs to eat, to learn, sex, friends, and to be safe (these are just some examples of human motivations). Despite this is not common, this situations rarely could happen (it could point out some abnormal situation in that person). Therefore, dynamics of drives and their parameters have been fixed considering that it is not desired that all motivations compete at the same time. Then, usually, there will be some motivations competing but not all at once. In case all motivations were constantly available to become in the dominant one, it would end up with a kind of hyperactive robot.

Dynamics of drives and motivations

In a similar way to any need on humans or animals, drives fluctuate. After we eat and the digestive process has begun, the need of energy is inhibited due to satiety signals. These satiety signals slowly dissipate until the hunger again takes over. Then, drives vary according to several signals and parameters [52]. Drives in the robot evolve in an analogous way. The evolution functions of drives are set by the designer and they affect the behavior of the robot. Since drives temporally evolve from scratch, motivations do as well.

Figure 5.3 shows the dynamics of all drives. The evolution functions for all drives do not have to be all equal. In fact, each drive fluctuates according to different functions.

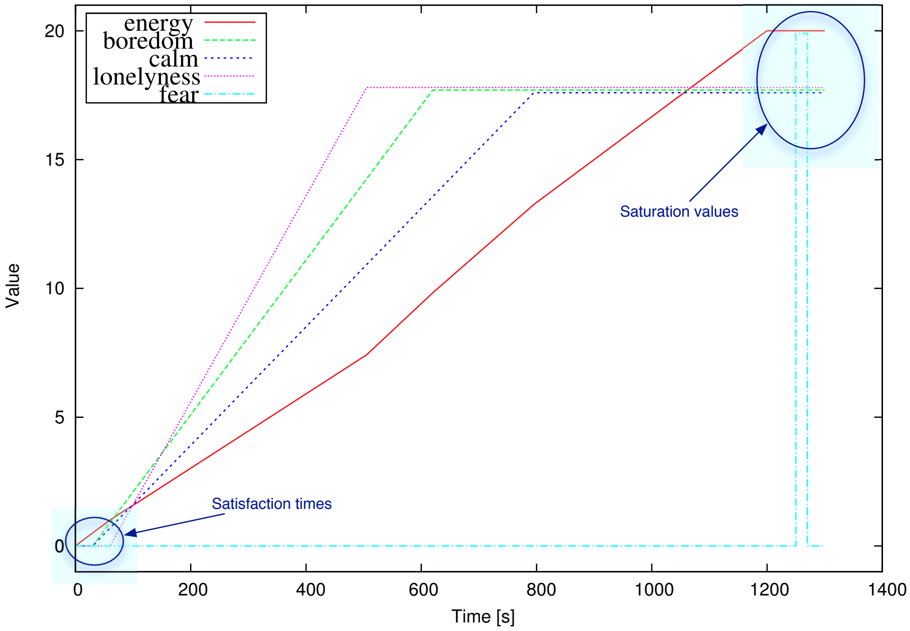


Figure 5.3: Comparison of drives progression.

Drives evolution is determined by three factors: the satisfaction time, the increasing function, and the saturation level. Following, each of this components is explained for each drive.

Satisfaction times After a drive is satisfied, it does not immediately start evolving, there is a *satisfaction time* before it increases again. The same idea occurs with human beings:

once we have eaten, we do not feel hungry again but it takes some time before hunger increases and we need to eat again.

In this implementation, each drive has a *satisfaction time*. This represents the period of time the drive remains at its initial value after it has been satisfied. During this time the drive does not evolve.

Each satisfaction time has been empirically set and they are summarized in Table 5.1. At the very beginning of Figure 5.3, the satisfaction times can be observed.

Table 5.1: Satisfaction times for all drives

Drive	**Satisfaction time**
energy	-
boredom	30s
calm	30s
loneliness	60s
fear	-

Since the drive *energy* mirrors battery level, it does not make sense to consider its satisfaction time. Besides, considering the previous definition of the *fear* motivation, satisfaction time does not make sense in relation with fear.

Increasing Functions After the satisfaction time passes, the drives start to increase. In the implementation proposed in this thesis, the *boredom*, the *loneliness*, and the *calm* drives linearly increase but with different slopes. It means that, as time goes by, these drives become bigger and bigger, and so do the corresponding motivations.

Considering that being social is one of the main characteristics of the robot Maggie, interaction with people is one of the most relevant aims. Therefore, the *loneliness* drive is the fastest one. This means that the motivation associated to this drive, *social*, frequently competes to be the dominant one. Consequently, the behaviors learned for this motivation are exhibited more often. It ends up with a robot whose most frequent behavior is the one related to human-robot interaction. The other drives evolve slighter.

The *boredom* drive goes after. This is because Maggie is conceived as a nice robot for people and a robot having fun is more attractive than a passive one. The *fun* motivation leads the robot to perform enjoyable reactions.

Finally, *calm* evolves smoother so it is the slowest drive. This implies that it is harder to exceed its activation level in order to struggle for being the dominant motivation. In addition, this drive just evolves when music is been played: Maggie needs to relax after it has been listening music for a while. Consequently, this provokes that the *relax* motivation scarcely becomes the dominant motivation.

As said before, the *fear* motivation is different. Theoretically, there is not drive linked to this motivation. However, from a computational point of view, a drive needs to be linked to the *fear* motivation. Then, the value of the drive *fear* rises to its maximum at once when a dangerous situation is detected (this can be seen on the right of Figure 5.3). When the state is considered as "safe", the fear dissipates. These high and low values of fear correspond to the numerical values of 19.9 and 0, respectively (Table 5.2). It is important to note again that these dangerous states are not predefined but they are learned by the robot itself through interaction with the environment. The appraisal of the fear emotion has been detailed in Section 4.4.2.

Table 5.2: Levels and values for *fear*

Level	**Value**
high	19.9
low	0

In order to achieve a fully autonomous robot, power autonomy is the first step. Therefore, the most relevant inner need, due to the implicit necessity of survival, is the *energy* drive. Therefore, this drive evolves as the battery level varies. So, its value matches the battery level.

As mentioned, many of the ideas related to the DMS have been previously developed and tested on simulations [49]. However, when it is implemented on a real robot like Maggie, new issues related to the energy management come up. Since the robot learns from the ground up how to behave in each situation, it may be the case that the robot is running out of battery and the selected behaviors are not the most appropriated. This may lead to the end of operation of the robot. The robot "dies" because its battery is depleted, so it cannot perfectly keep on working. In order to avoid this situation, during the learning process, the progress of the battery level is simulated. Hence, the drive of *energy* progresses as presented in Figure 5.3.

In addition, the battery progress emulation reduces the length of the experiments. Real full battery recharging takes up to two hours; this would imply experiments of very long duration (several days). Virtual battery recharging has been set to two minutes so the length of the experiments is reduced up to several hours.

Besides, when the robot is recharging its battery, it is similar to an asleep person. According to [52], during sleep (specially during non-REM stages which roughly are 75% of the total sleep time) human body rests: the temperature and energy consumption of the body are lowered, and heart rate, respiration, and kidney function slow down. This is imitated by the system: the *boredom*, the *calm*, and the *loneliness* drives are almost frozen during the battery recharging. This is required because, if the drives' rates are not reduced, since the action *recharge battery* takes a long time in comparison with the rest of the actions, after this action is over, all drives would enormously increase. Consequently, the robot's wellbeing would be greatly reduced and, therefore, the reward would be always negative for the action *recharge battery*. Then, the robot would not properly learn what it has to do in order to recharge its battery. Permanent negative reward for a certain state-action pair prevents the system from executing that action from that state because its value is really bad. Therefore, "freezing" the drives during the charging of the battery is necessary.

Saturation levels In order to avoid an unstopped increase of the value of the drives, a saturation level is defined for each one. The saturation level correspond to the maximum value of a drive: once a drive has reached its saturation level, it does not exceed this value and remains at it.

Different drives have different saturation values which affect the dominant motivation in case of a never-ending expansion of the drives. These saturation levels can be seen as an emergency control mechanism in case that several drives are saturated and their motivations compete to be the dominant one. In this situation, the saturation levels work as predefined priorities that determine the dominant motivation in those exceptional situations. These priorities can be seen as inherited knowledge or instincts in living beings which allow them to face extreme situations. Table 5.3 presents the sorted list of saturation levels.

Table 5.3: Saturation level for all drives

Drive	Saturation level
energy	20
fear	19.9
loneliness	17.8
boredom	17.7
calm	17.6

In this implementation, *energy* has the highest saturation level because it is the most urgent since it is related to survival: if the energy drive is saturated it means that the battery level is really low and it is critical to get the battery recharged.

Fear is the second one so it is over the rest of drives. As explained before, when a dangerous situation is perceived, the *fear* value is set to its maximum, which corresponds to the saturation value. This value is over the others because *fear* represents a really dangerous situation which must be avoided somehow as soon as possible. Just survival can be more urgent than *fear*.

The rest of the saturation values where fixed considering the same reasons used for the evolution functions of the drives (see Table 5.3).

External stimuli

Just like human beings can be thirsty when they see water, the motivations are influenced by some objects when they are present in the environment. These are called the **external stimuli** or incentives. These stimuli may have more or less influence: their values depend on the states related to the objects (this means, if they are near or far from the robot). The external stimuli are included in Equation (4.1). In this implementation, all external stimuli values have been empirically fixed to the same value of 2 and, according to Section 2.3.2, they anticipate the reduction of a certain drive.

Table 5.4 lists all the external stimuli included in this work. Since the robot likes dancing when music is being played, the robot perceives it and the motivation to have *fun* increases. If Maggie perceives the docking station, the motivation of *survival* is augmented. Lastly, due to the fact that Maggie is a very friendly robot and loves people, the presence of a person close to it strengths its *social* motivation.

Table 5.4: All external stimuli used in this work

Motivation	**External stimuli**	**State related to ext.stim.**
fun	music	listening
survival	docking station	plugged
social	any person	close

5.4.2 The external world: sensing and acting

The world is perceived by the robot in terms of objects and the states related to these objects (the external state). Objects are not limited to physical objects but abstract objects too. In this dissertation, the world where Maggie is living in is limited to the laboratory and the following objects: a music player, the music in the lab, the docking station for supplying energy, and the people living around the robot.

Also the states related to all these items have to be defined and the transitions between states are detected by several skills running in Maggie.

Moreover, the robot interacts with its environment through the actions that can be performed with the objects in the robot's environment. These actions are also implemented as skills in the AD architecture.

At this point, it is worth mentioning the difference between two concepts which, many times, are mixed and used as synonyms: **behavior** and **action**. Considering the definition given by Breazeal in [4], in this work, behavior is viewed as a self-interested, goal-directed entity that establishes the current task of the robot. In general, a behavior is composed of a sequence of related actions which are activated in turn. For example, the behavior to reduce hunger is composed of an action for eating and other for moving near the food. Therefore, there are two kind of actions (from an ethologist point of view, these are referred as behaviors too) [4, 124]:

- **Consummatory**: this action directly satiates the active drive, i.e. the most urgent need. Then, they contribute to the balance of resources that ensure self-sufficiency

- **Appetitive**: when an appetitive action is performed in a certain situation, leads to a state of the world for allowing the activation of the desired consummatory action. In other words, it is an action that makes more likely the conditions that bring closer some goal.

In the previous example, eating is a consummatory action and moving towards the food is an appetitive one because it is necessary before the drive can be satiated. Both together form the behavior to reduce hunger.

In this thesis, actions are individual, indivisible tasks which corresponds to skills in the AD architecture (Section 5.3.4). The behaviors are sequence of actions which are determined by the dominant motivation and the external stimuli. These behaviors are learned, so they are not predefined.

Besides, actions can be categorized into *endogenous* and *exogenous* actions. Endogenous actions are those which are executed by the robot. In contrast, as mentioned before, exogenous actions refer to actions executed by other agents. These actions are not “observable” by the robot, that is, it can not identify the action, but their effects are perceived by the robot. These effects are mixed with the effects coming from the robot’s own actions. In order to distinguish both effects, the effects from the exogenous actions are just considered when certain endogenous actions are running. These endogenous actions do not affect the robot or its environment, so the variation of the robot’s wellbeing is due to exogenous actions. In short, in this thesis, the robot has two kinds of actions: actions disturbing the robot and/or its environment, and “effectless” actions that allow to consider the effects of exogenous actions.

In Figure 5.4, the states related to each object, the actions, and the transitions from one state to another are shown. 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 music* if it is *far* from the player; or it cannot *interact* with a person if it is alone.

Following, the available items, the states related to them, and their actions are introduced.

Music player

Maggie is able to operate any home appliance with an infrared interface by means of an infrared emitter/receiver placed at Maggie’s belly and several skills. In this work, this has been applied to a music player located in the lab (all details have been published in [195]).

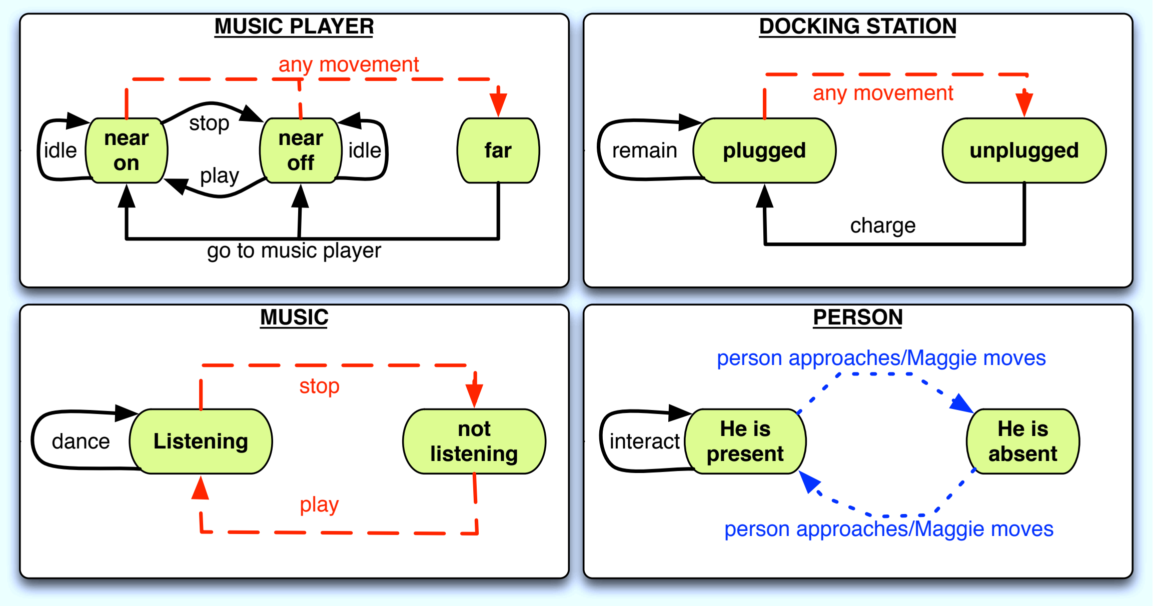


Figure 5.4: States, actions and transitions related to the items of the robot's environment: a music player, the docking station, the music, and a person. Round sides rectangles represent the states related to each object, the arrows are the transitions, and the labels of the arrows are the actions which may cause the transition if no errors occur. Black arrows correspond to transitions triggered by actions executed with the object. Red dashed arrows mean transitions activated by actions with other objects. And purple dotted arrows are dedicated to transitions due to actions executed by other agents

In order to operate the music player, the robot has to be located at a certain distance and facing the appliance. Therefore, in relation to the position of the robot, there are two states: *near*, when the robot is close enough to operate the player, and *far*, if the robot is in a position where it is not able to operate the player. The information required to determine the position related to the player is provided by the geometrical navigation system [196] which knows where the music player is.

Moreover, related to the operational state of the music player, other two states have to be distinguished to avoid sending the same command twice to the player: *near-on* and *near-off*. When the robot is close to the player and it is already working, the state is *near-on*; but, when the robot is also close and the player is off, the state is *near-off*.

About the available actions, if the robot is *far* from the player, the only possible action related to the music player is to *go to player*. Once it is close, and according to the last state in relation to the player, the robot can *play* or *stop* music.

Then, the possible actions with the item *music player* are:

- Go to player: Maggie approaches the music player. The robot will move to a position where the player is reachable by the infrared emitter on board Maggie. If the robot was plugged to the charging station, this action unplugs the robot.

- Play music: music is played because it turns the player on when it is off. This action produces a change of state in relation to other object, the *music*, from *non-listening* to *listening*.
- Stop music: music is stopped when it is being played because the music player is turned off. This action produces a change of state regarding the object *music*, from *listening* to *non-listening*. This action keeps a peaceful atmosphere.
- Idle: it represents the possibility to remain next to the player for a while.

Music

The robot's environment is the lab, and *music* can be playing there. Then, the robot can be *listening*, or *not*, to *music*. Just when the robot is *listening* to *music*, it is able to *dance*. If *music* is mute, it cannot *dance*. As commented before, the infrared emitter is used to play/stop the music when Maggie is close 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* to music. This action can be executed at every place inside the lab because the music is perceived from anywhere in the room.

Docking station

The *docking station* is the source of energy. If the robot is *plugged*, the battery is charging, so its level increases. Otherwise, the robot is *unplugged* and the battery level decreases. 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 in charge of reading the battery data. This information is read by the *battery sensor* skill.

When the robot is *unplugged*, it just can go to the docking station and *charge* its battery. After that, it is *plugged* in and the available action is to *remain* there. If, when the robot is plugged, a skill that moves the robot around is selected, it leaves the station and transits to the state *unplugged*.

The attainable actions with the docking station are:

- Charge: Maggie approaches the docking station, plugs into it, and stays there until the battery is full. At the end of this action the robot is still *plugged* and the battery is recharged.
- Remain: it keeps plugged for a while.

Person

The robot Maggie is intended to interact with people. Hence, people are considered as "objects" of the environment. Regarding interaction, a person has to be close enough to touch, speak or being recognized. For that reason there are two states in relation to a person: *present* and *absent*. The state *present* means that there is a person nearby the robot. In contrast, *absent* represents the absence of any person or, at least, no person within a distance close enough for interaction. These states are determined by merging two technologies, bluetooth and RFID, which are handled by the skills *Bluetooth discoverer* and *RFID discoverer* (Section 7.5.7).

Each person, or user, is equipped with an RFID tag which provides a low range distance identification. These tags are read by an UHF antenna placed at Maggie's chest which provides around 1 meter range. In addition, each person's mobile phone is detected by its bluetooth interface which offers medium range distance identification. The combination of both technologies results on a reliable identification method.

As stated in Section 4.4.2, the undoubted identification of the effects of exogenous actions in each state is the cornerstone for learning the dangerous states and, by extension, the realeasers of the *fear* emotion. This is achieved through the *interact* action. By means of this action, the robot does not induce any change on its internal variables or its environment. The assumption is that all the changes experienced by the robot during the action *interact* are a consequence of external elements. In this scenario, these available conative "external elements" are people who interact with the robot during the *interact* action. Thus, in this dissertation, active objects are people or individuals coexisting with the robot. Using this approach, the robot estimates how good the current state is in relation to the exogenous actions because all effects and transitions are due to the people's actions. This estimation is based on the variation of the robot's wellbeing and it is used to learn new dangerous states.

The *person* item offers an available action:

- Interact: it perceives human-robot interaction. With this action the robot is not executing any particular ability or task, so the robot does not cause any particular effect over itself or the environment. Therefore, the possible consequences during this action are certainly caused by the exogenous actions. Since *persons* are the available active objects, during this action, the robot perceives the effects of the people's action over the robot's wellbeing. These effects are evaluated through oral and tactile interfaces: the user can offend or say compliments to the robot, or he can "stroke" or "hit" the robot (Section 7.5.7).

Considering that, in the robot's environment, people are the only active objects, it is assumed that the effects during the action *interact* are caused by *persons*'s actions. However, this would not be necessarily true because, the system just considers the robot's state and the effects over the robot's wellbeing. Then, the effects could be provoked by any other active object different than the *person* items.

The system provides identification for different users. Then, different users are treated as different objects of type *person*. Therefore, the robot learns what to do with each user independently. This object is a key component in this work because, since it is an active object, it is able to execute its own actions. Several types of object *person* have been used to prove the performance of the *fear* motivation.

An overview of the robot's environment is displayed in Figure 5.5. It provides a good perspective of the scenario and the different types of objects the robot interacts with during the experiments.

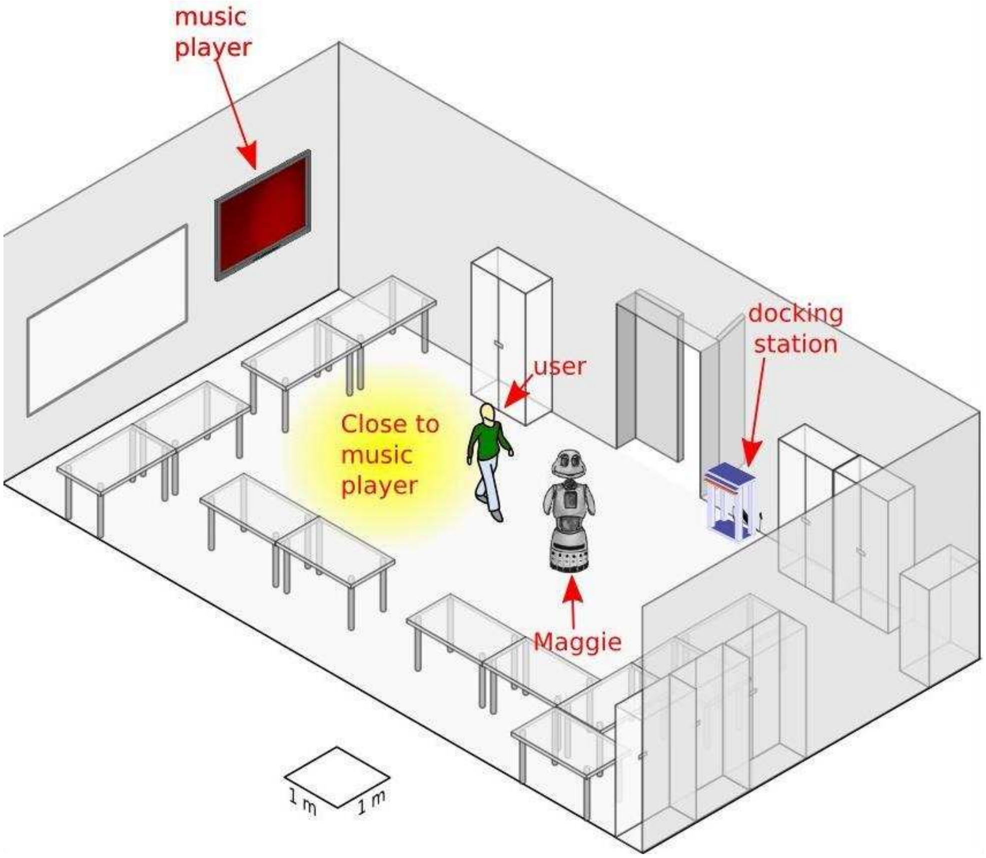


Figure 5.5: Overview of the robot's environment and the objects the robot interacts with

5.4.3 Acting in the world: what to do next?

After the environment where the robot lives has been presented, the action selection process by the DMS is explained. Within robot lifetime, the action selection loop is executed in order to determine the next skill to activate. Remembering how the DMS works, 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. Considering the dominant motivation, the current states related to objects, and the Q values associated to each feasible action in this state, the next action is chosen. These Q values represent how good a particular action is at a particular state.

At the beginning of the robot's life, it does not have any knowledge, so learning is essential. In order to help learning, the robot explores all possibilities many times. But, in order to live better, the robot has to exploit the acquired knowledge to make the best decisions. This is the dilemma of exploration vs. exploitation, several times refereed in the field of reinforcement learning [163]. The level of exploration represents the probabilities of executing actions different than those with the highest values. Exploitation means the selection of the action with the highest value for each situation. Therefore, during the robot's life, there are two phases clearly differentiated: learning or exploration phase, and exploiting phase.

Then, according to a specific level of exploration/exploitation, the probabilities for selecting an action differs. Using the Boltzmann distribution, the probabilities of selecting an action a in a given state s is determined by Equation (5.1).

$$P_s(a) = \frac{e^{\frac{Q(s,a)}{T}}}{\sum\limits_{b \epsilon A} e^{\frac{Q(s,b)}{T}}} \tag{5.1}$$

$Q(s,a)$ is the value for action a in state s, and A represents the set of all possible actions; T is the *temperature* and it ponders exploration and exploitation. A high value of T gives the same likelihood of selection to all possible actions and the next action is almost randomly selected; low T enforces actions with high values: the higher value, the higher probability to be executed. This approach has been previously used by Gadanho [197, 198]. As presented in [49], T value is set according to Equation (5.2).

$$T = \delta * \bar{Q} \tag{5.2}$$

where $\bar{Q}$ is the mean value of all possible Q values. According to the Equation (5.2), high δ implies high temperature and, therefore, exploration dominates: all actions have the same probability of being selected. Low δ produces low temperatures and, consequently, exploitation prevails: actions with high values are likely chosen.

Therefore, when learning is essential, δ is set to a very high value so actions are randomly chosen, independently of their values, so all actions are explored. However, when it is desired to select the most appropriate actions, δ is minimized. Then, the action with the highest values are always chosen.

During the experiments, δ is varied depending on the phase of the robot's life: during learning, high level of exploration is required ($\delta = 100$), then the action selection is totally random; when exploiting the learned values, the exploration is null ($\delta = 0.1$) and the next action is the one with the highest value.

Considering that this work is implemented in a social robot which interacts with humans, it should be kept in mind that a robot which is programmed for always selecting the best actions leads to monotonous behaviors and the robot's actions become very predictable. Consequently, the human-robot interaction can be negatively affected due to the potential lost of interest by the user. In order to allow some randomness in robot's behavior, the value of δ can be tuned for providing certain unpredictability to the process.

5.4.4 The consequences of the robot's actions

Once an action is selected and executed, it may disturb the robot in two manners: first, an action provokes a change in the world, e.g. *charge* action results on the robot is plugged to the charger; and second, the action causes effects over the drives, e.g. after the *charge* action the need of *energy* is reduced. In order to apply the effects over the drives, the action has to successfully end. If an error occurs during the execution of a skill or its result is not satisfactory, this situation is notified and its effects over the drives are not applied. The changes affecting the external state are monitored by specialized skills.

Summing up, effects of the actions can modify the state related to items and influence the needs of the robot. In relation to the robot's drives, the effects can be positive or negative, in terms of robot's wellbeing. A positive effect reduces the value of a robot's drive (this implies an increase in the robot's wellbeing). Actually, when the drive is set to zero (the ideal value), it is said that the action satisfies the drive. Some actions can also "damage" some drives of the robot increasing their values (so the robot's wellbeing drops).

A positive effect, i.e. the reduction of one drive, does not necessary imply an improvement in the wellbeing. If the reduced drive had a value close to the ideal one, the effect of the action in the total wellbeing is minimum. Other drives could increase faster or the external state has changed resulting in a decrement of the wellbeing.

All effects are shown in table 5.5.

Table 5.5: Effects of actions

Action	**Object**	**Drive**	**Effect**
stop	music player	calm	set to 0
dance	music	boredom	set to 0
positive interaction	person	social	set to 0
negative interaction	person	social	+10

When the music player is switched off, the drive *calm* is satisfied; then, a quiet environment is achieved. The need of *fun* is satiated when the robot dances, so the drive *boredom* is set to zero. Since human-robot interaction involves a user, the result of this actions is not always the same. Depending on how this user behaves, the action *interact* is positive or negative. A positive interaction is related to a stroke or a compliment and satisfies the *social* drive. In contrast, a negative interaction provokes an increment of ten units in the *social* drive. This happens when the robot is damaged because of a hit or an insult.

It is important to mention that the transitions between two states and the effects of the actions are not given to the DMS, this is, the model of the world is not provided in advance. Therefore, this is a model-free approach. However, these effects are defined by the designer and applied to the robot's drives.

As already stated, the potential actions in each state depend on the state itself. Hence, different actions are associated to the state related to every object. For example, in order to *play music*, Maggie has to be close to the *player* and the music has to be switched off (*near-off* state). In some cases, the states and the actions are impossible. For instance, if the robot is *unplugged* from the *docking station*, the action *remain* plugged cannot be executed because it is not plugged. In these cases, there are not Q values associated to these state-actions pairs.

In other circumstances, some actions seem not be very appropriated. For instance, it does not make sense to execute the *charge* action when the robot's battery is full. By means of the learning process, these combinations receive minimal values and, in consequence, they will never be selected for execution during the exploitation phase.

5.5 Summary

At the beginning, this chapter presents the robotic platform where the DMS is implemented: the social robot Maggie. The hardware forming the robot is described as well as its control architecture. In this thesis, the AD architecture is extended by adding the DMS.

In the last part of the chapter, the specific configuration of the DMS which has been used in this thesis is presented. Drives, motivations, objects, actions, and other variables are defined and justified. The modification of some of these variables results in a robot which behaves different, like if the "personality" of the robot had changed.

In short, this chapter presents the robot and the configuration of the DMS. This configuration can be modified according to different requirements without a great effort.

CHAPTER 6

Learning to make decisions

6.1 Introduction

In this dissertation, the learning process is achieved by real interactions between a robot and its real environment. Interaction in real environments takes a considerable amount of time. This makes the learning time a key feature. Then, achieving the learning task in a reasonable amount of time is a must.

As mentioned before, the external state of the robot is formed considering the state of all objects in relation to the robot. Then, in a traditional RL approach, the number of states exponentially increases as the number of objects linearly increases. Consequently, the learning time exponentially increases too because, in RL theory, in order to reach the convergence of the learned values, all states must be visited an infinite number of times.

This chapter presents the solution to the learning process implemented in this dissertation: the Object Q-Learning algorithm. This solution was initially designed for and tested in virtual worlds. Then, it has been extended with several improvements in order to deal with the problems of learning in a physical world.

6.2 Object Q-Learning

Malfaz presented in [49] a variation of the traditional Q-Learning algorithm (Section 4.3.1). This is called **Object Q-Learning** and it has two key points:

1. A reduction of the state space

2. The Object Q-Learning algorithm

Both are explained in the following sections.

6.2.1 The state space

In this thesis, it is assumed that the robot lives in an environment where it can interact with 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:

$$S = S_{inner} \times S_{external} \tag{6.1}$$

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:

$$S_{external} = S_{obj_1} \times S_{obj_2} ... \tag{6.2}$$

therefore,

$$S = S_{inner} \times S_{external} = S_{inner} \times S_{obj_1} \times S_{obj_2} ... \tag{6.3}$$

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

For example, considering a situation where the robot's battery is almost depleted, its internal state is related to the survival motivation ($S_{inner} = survival$). Besides, in relation to the objects (the external state), the robot is alone, far from the player, plugged and it is listening music. Then, the resulting state is computed in Equation (6.4).

$$\begin{array}{l} S = S_{inner} \times S_{external} = S_{inner} \times S_{obj_1} \times S_{obj_2} ... = \\ S_{dominant\, mot} \times S_{person} \times S_{player} \times S_{charger} \times S_{music} = \\ survival\, and\, alone\, and\, far\, and\, plugged\, and\, listening \end{array} \tag{6.4}$$

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 Equation (6.2), 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 (Equation (6.3)), the final number of states would be even bigger since the number of external states must be multiplied 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, $Q(s, a)$ in Q-Learning, for every state-action pair, could become really high. This great number of Q values to calculate presents problems since it would take really long time for those values to converge.

6.2.2 Reduced state space

As previously stated, as the number of variables (objects) linearly increases, the number of states increases exponentially. This problem is known as the curse of dimensionality [199]. 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 [200].

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 [201]. In [202] and [199] 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 [203], 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 [204, 205]. Others consider that states should be aggregated if they have the same optimal action, or similar Q-values [206], etc.

In Malfaz's work [49], she proposes 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 behavior, 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, for 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. The set of the reduced external states, $S^{red}_{external}$, is represented in Equation (6.5).

$$S^{red}_{external} = \{S_{obj_1}, S_{obj_2}, S_{obj_3}, ...\} \tag{6.5}$$

For example, following the example presented at the end of the previous section, the 10 objects present in the world results in $10 \times 16 = 160$ 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:

$$s \,\epsilon\, S_i = S_{inner} \times S_{obj_i} \tag{6.6}$$

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

Recalling the example, exposed in Section 6.2.1, where a robot is running out of battery, and considering the reduced state space just presented, the state of the robot is expressed in Equation (6.7).

$$\begin{array}{l} S = S_{inner} \times S_{external} = S_{inner} \times \{S_{obj_1}, S_{obj_2}, ...\} = \\ S_{dominant\,mot} \times \{S_{person}, S_{player}, S_{charger}, S_{music}\} = \\ survival\, and\, \{alone\, or\, far\, or\, plugged\, or\, listening\} \end{array} \tag{6.7}$$

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.

Considering this simplification, the Equation (4.4) is adapted for the updating of the $Q^{obj_i}(s, a)$ value of the state-action pairs for an inner state and an object i:

$$Q^{obj_i}(s, a) = (1 - \alpha) \cdot Q^{obj_i}(s, a) + \alpha \cdot \left(r + \gamma \cdot V^{obj_i}(s')\right) \tag{6.8}$$

Where:

$$V^{obj_i}(s') = \max_{a \in A_{obj_i}} \left(Q^{obj_i}(s', a)\right) \tag{6.9}$$

The super-index obj_i indicates that the learning process is made in relation to the object i; therefore, $s \in S_i$ 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 $s' \in S_i$ 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.

As a consequence of this simplification, the learned Q values, instead of being stored in a table of *{total number of states × total number of actions}* dimension, are stored for a certain inner state and for every object in a table of *{number of states related to that object × number of actions related to that object}* dimension.

6.2.3 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 music player, this action will not affect to the rest of objects. Nevertheless, if the robot was previously recharging its battery in the docking station, this action (to go to the music player), which is related to the object music player, has affected to its state in relation to the docking station. Moreover, if a person is nearby the robot, after it has moved towards the music player, now this person is not present anymore. As result, an action (approaching the music player) related to a particular object (the music player) may influence other items (the docking station and a person). This is exactly what happens in real life: a person, who is close to water, goes for food, and the resulting state is that now the person is close to food but far from water. Therefore, the assumption of that objects are independent among them is not totally true. The consideration of **collateral effects** in the learning algorithm deals with this problem.

The collateral effects are those effects produced by the robot in the rest of the objects when interacting with a certain object. Therefore, in order to take into account these collateral effects, the Object Q-learning has to consider how the action with a particular object affects the rest of objects. Using this viewpoint, the Q values are still updated according to Equation (6.8) but, now, $V^{obj_i}(s')$ is calculated according to Equation (6.10).

$$V^{obj_i}(s') = \max_{a \in A_{obj_i}} \left(Q^{obj_i}(s', a) \right) + \sum_{m \neq i} \Delta Q_{\max}^{obj_m} \tag{6.10}$$

This is the value of the object i in the new state s' considering the possible effects of the action a executed 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 (6.9).

These increments are calculated as follows in Equation (6.11).

$$\Delta Q_{\max}^{obj_m} = \max_{a \in A_{obj_m}} \left(Q^{obj_m}(s', a) \right) - \max_{a \in A_{obj_m}} \left(Q^{obj_m}(s, a) \right) \tag{6.11}$$

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. Then, 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. In other words, it measures if the value of the new state is better or worse than the value of the previous state in relation to each object. This algorithm has been introduced in previous works [158, 207], where it was successfully implemented in virtual agents.

Considering the example presented at the beginning of this section, if the objects the robot can interact with are limited to a *music player* and the *docking station*, the current states related to these objects are *far* from the player and *plugged* to the charger. Once the action *go to* the player is executed, then the new states are *close* to the player and *unplugged* from the *docking station*. Therefore, the Object Q-Learning is applied as follows [1]. From Equations (6.10) and (6.11), the Q value is computed according to the next equation:

$$\begin{aligned} &Q^{player}(far, go\,to) = \\ &= (1-\alpha)\cdot Q^{player}(far, go\,to) + \alpha \cdot \big(r + \gamma \cdot V^{player}(close)\big) \end{aligned}$$

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

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

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

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

where a is any action related to *charger*.

6.2.4 The algorithm

Once the ideas of the algorithm have been stated, the algorithm itself has to be analyzed. In a RL framework, an agent in a state executes an action, it transits to a new state, and a reward is obtained. In an Object Q-Learning framework, the state is determined in relation to the objects and the potential actions are restricted by the state: an agent is in a state related to a particular object i (s_{obj_i}) and it executes an action with this object (a_{obj_i}); this action can provoke a change in the state related to this object (s'_{obj_i}) and a reward (r); in addition, this action can also provoke changes in the state related to other objects ($s_{obj_j}, \forall j \neq i$), which have been called the collateral effect. All these elements are presented in Figure 6.1; the collateral effects are represented by dashed arrows.

The algorithm updates the Q values after an action is executed. Then, these values are refreshed according to the reward obtained, the anterior and new states, and the prior Q

[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.

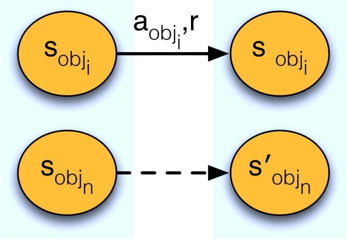


Figure 6.1: The Object Q-Learning framework

values. Every time a Q value is updated, it is referred as an iteration. The pseudo-code for the algorithm is detailed in Algorithm 6.1. Initially, all Q values have to be set to a random value, in this case they were fixed to 1 (line 2). Then, the algorithm iterates every time the robot acts. First, the collateral effects are computed (lines 4-11). For each object, the difference between the best the robot can do from the new state with that object and the best it could do from the anterior state is calculated and added to the *collateral_effect* variable. Once the collateral effects for all items are calculated, the value for the object i in state s' is determined as the sum of the Q value corresponding to the best action with object i from the state s' (line 12) and the collateral effects. With these values and the prior Q value, the new Q value for the object i in the state s when the action a is accomplished is updated (lines 13-15).

In order to provide a clear understanding of this algorithm, several real examples will be analyzed step by step. The calculations shown in the next examples are the numbers resulting at single iterations during the experiments with the robot. Different experiments could result in different values. In all these examples, the actions executed have been related to the object *music player* (it is marked with an asterisk in the state transition tables), but in different situations. Trying to keep the examples as clear as possible, no user has been included in the following scenarios. Besides, when there are not feasible actions from a particular state, this is represented in collateral effects tables with a hyphen.

The learning rate and the discount factor for all the scenarios have been fixed to $\alpha = 0.3$ and $\delta = 0.8$, respectively.

Algorithm 6.1 Object Q-Learning algorithm

```
procedure COMPUTE OBJECT Q-LEARNING
    Initialize all Q values to 1
    repeat for each iteration
Require: $s \leftarrow$ current state
Require: $a \leftarrow$ executed action
Require: $object_i \leftarrow$ object the action is executed with
Require: $s' \leftarrow$ new state
Require: $r \leftarrow$ reward
        $collateral\_effects \leftarrow 0$
        for all $object_j$ do
            if $object_j \neq object_i$ then    ▷ The collateral effects do not consider the object that the action was executed with
                $max\_q\_s \leftarrow max[Q^{obj_j}(s_{obj_j}, a)]$
                $max\_q\_new\_s \leftarrow max[Q^{obj_j}(s'_{obj_j}, a)]$
                $collateral\_effects \leftarrow collateral\_effects + (max\_q\_new\_s - max\_q\_s)$
            end if
        end for
        $value\_obj_i\_new\_s \leftarrow max[Q(s'_{obj_i}, a)] + collateral\_effects$
        $q \leftarrow Q(s_{obj_i}, a_{s_{obj_i}})$
        $new\_q \leftarrow (1 - \alpha) \cdot q + \alpha(r + \delta \cdot value\_obj_i\_new\_s)$
        $Q(s_{obj_i}, a_{s_{obj_i}}) \leftarrow new\_q$
    until learning ends
end procedure
```

Scenario 1

In this first scenario, the robot needs calm (i.e. *relax* is the dominant motivation), it is *unplugged* to the *docking station*, it is *listening* to *music*, the robot is close to the player and there is not users around. Then, the robot decides to *stop* the *music player*. The state transitions are shown in Table 6.1. This action affects three elements; first, the dominant motivation changes: after the player is turned off, there is not a new dominant motivation because the need of calm has been satisfied and the intensity of the other motivations is not high enough; also the states of the *music player* and the *music* have changed too. This action is related to the object *music player* but also the object *music* is affected. The value of the collateral effects is calculated in Table 6.2.

In this particular case, the corresponding Q value, $Q^{player}_{relax}(near\text{-}on, stop)$, is updated as follows in Equations (6.12) and (6.13).

Table 6.1: State transitions due to the action *stop music* in Scenario 1

	anterior state (s_t)	**new state (s_{t+1})**
dominant motivation	relax	none
docking station	unplugged	unplugged
music player *	near-on	near-off
music	listening	non-listening
user	absent	absent

$$Q^{player}_{relax}(near-on, stop) =$$

$$(1-\alpha)\cdot Q^{player}_{relax}(near\text{-}on, stop) + \alpha\cdot\left(r+\gamma\cdot V^{player}_{none}(near\text{-}off)\right) \tag{6.12}$$

$$V^{player}_{none}(near\text{-}off) = \max_{a\in A_{player}}\left(Q^{player}_{none}(near\text{-}off, a)\right) + \sum_{obj_m\neq player}\Delta Q^{obj_m}_{\max} \tag{6.13}$$

Table 6.2: Collateral effects due to the action *stop music* in Scenario 1

$\mathbf{Object_m}$	$\mathbf{max_{a\in A_{obj_m}}\left(Q^{obj_m}\left(s_{t+1},a\right)\right)}$	$\mathbf{max_{a\in A_{obj_m}}\left(Q^{obj_m}\left(s_t,a\right)\right)}$	$\mathbf{\Delta Q^{obj_m}_{max}}$
docking station	$Q^{station}_{none}(unplugged, charge) = -1,17895$	$Q^{station}_{relax}(unplugged, charge) = 1$	-2,17895
music	$Q^{music}_{none}(non\text{-}listening, -) = -$	$Q^{music}_{relax}(listening, dance) = 1$	-1
user	$Q^{user}_{none}(absent, -) = -$	$Q^{user}_{relax}(absent, -) = -$	$-$
$\mathbf{\sum_{obj_m\neq player}\Delta Q^{obj_m}_{max}}$			-3,17895

The reward and the rest of the parameters which are required for updating the Q value, as well as the new Q value, are presented in Table 6.3. Since this is the first time this action is executed in the state s_t, its Q value corresponds to the initial value of 1. From the new state (s_{t+1}), the best thing to do with the *music player* is to turn it on, which has a calculated value of $1,154$.

Table 6.3: New Q value for Scenario 1

$\mathbf{Q^{player}_{relax}(near\text{-}on, stop)}$	**reward**	$\mathbf{V^{player}(s_{t+1}) = V^{player}_{none}(near\text{-}off)}$		$\mathbf{new\ Q^{player}_{relax}(near\text{-}on, stop)}$
		$\max_{a\in A_{player}}\left(\mathbf{Q^{player}_{none}(near\text{-}off, a)}\right)$	**Coll.Effects**	
1	52,5399	$Q^{player}_{none}(near\text{-}off, play) = 1,154$	-3,17895	15,975982

In this scenario, the most influent parameter is the reward. Stopping the music player results in the satisfaction of the drive *calm*. Therefore, the *relax* motivation is considerably reduced and it ceases to be the dominant one. This is the reason of the high value of the obtained reward $(52, 5399)$ and, consequently, the rises of the resulting Q value, $Q^{player}_{relax}(close\text{-}on, stop)$.

This is a clear example about how the robot learns the value of the action which directly receives the reward from the satisfaction of the most urgent need (the dominant motivation). This is a consummatory action. The value of the state-action pair is back-propagated as shown in the next example.

Scenario 2

In this example, the robot again needs to relax, but now it is plugged and far from the music player and it decides to approach it (action *go to the music player*). Moreover, the dominant motivation does not change after this action, and the robot ends unplugged from the charger and close to the player. The music is still listening and users are not present in the scenario of the experiments (Table 6.4).

Table 6.4: State transitions due to the action *go to the music player* in Scenario 2

	anterior state (s_t)	**new state (s_{t+1})**
dominant motivation	relax	relax
docking station	plugged	unplugged
music player *	far	near-on
music	listening	listening
user	absent	absent

Now, the new Q value is computed according to the Equations (6.14) and (6.15).

$$Q^{player}_{relax}(far, go\,to) = \\ (1-\alpha)\cdot Q^{player}_{relax}(far, go\,to) + \alpha \cdot \left(r + \gamma \cdot V^{player}_{relax}(near\text{-}on)\right) \tag{6.14}$$

$$V^{player}_{relax}(near\text{-}on) = \max_{a \in A_{player}} \left(Q^{player}_{relax}(near\text{-}on, a)\right) + \sum_{obj_m \neq player} \Delta Q^{obj_m}_{\max} \tag{6.15}$$

The collateral effects are just related to the transition from *plugged* to *unplugged* (Table 6.5). The object *music* does not change its state and, then, its collateral effect is null. The summation of all collateral effects is a negative value, which means that the state-action pair is not positive from the perspective of the other items. That is, if the robot needs to relax, approaching the music player, when the robot is far from it, is not a good action just considering the collateral effects.

The reward obtained after approaching the music player is very poor, $-1, 555$ (Table 6.6), because this action does not have any particular effect over the motivations. However,

Table 6.5: Collateral effects due to the action *go to the music player* in Scenario 2

$\mathbf{Object_m}$	$\mathbf{max_{a\in A_{obj_m}}\left(Q^{obj_m}\left(s_{t+1},a\right)\right)}$	$\mathbf{max_{a\in A_{obj_m}}\left(Q^{obj_m}\left(s_t,a\right)\right)}$	$\mathbf{\Delta Q^{obj_m}_{max}}$
docking station	$Q^{station}_{relax}(unplugged, charge) = -6,9738$	$Q^{station}_{relax}(plugged, stay) = 0,31$	-7,2838
music	$Q^{music}_{relax}(listening, dance) = 2,71541$	$Q^{music}_{relax}(listening, dance) = 2,71541$	0
user	$Q^{user}_{relax}(absent, -) = -$	$Q^{user}_{relax}(absent, -) = -$	$-$
$\mathbf{\sum\limits_{obj_m\neq player}\Delta Q^{obj_m}_{max}}$			-7,2838

the value of the new state is really high; this is not because of the collateral effects (in fact, its value is negative) or the reward (also negative), but because of the value of the best action that can be executed in the new state with the object *music player*: to stop the music $(50,9211)$. Then, the new Q value rises up to $15,117677$.

Therefore, the high reward obtained in the first scenario, after the execution of the action *stop* when the robot needs to relax, is propagated to the state-action pairs required to achieve it. These actions correspond to appetitive actions. This high reward is strong enough to back-propagate even with negative reward and negative collateral effects.

Table 6.6: New Q value for Scenario 2

$\mathbf{Q^{player}_{relax}(far, go\,to)}$	**reward**	$\mathbf{V^{player}(s_{t+1}) = V^{player}_{relax}(near\text{-}on)}$		$\mathbf{new\ Q^{player}_{relax}(far, go\,to)}$
		$\max\limits_{a\in A_{player}}\left(\mathbf{Q^{player}_{relax}\left(near\text{-}on,a\right)}\right)$	**Coll.Effects**	
7,30175	-1,555	$Q^{player}_{relax}(near\text{-}on, stop) = 50,9211$	-7,2838	15,117677

Scenario 3

This scenario shows how the collateral effects positively influence the update of a Q value. In this scenario, the dominant motivation is *fun*. The robot is close to the *music player* and, initially, it is not listening the *music*. The robot executes the action *play* which switches the *music player* on. The state transitions are detailed in Table 6.7.

Table 6.7: State transitions due to the action *play music* in Scenario 3

	anterior state (s_t)	**new state (s_{t+1})**
dominant motivation	fun	fun
docking station	unplugged	unplugged
music player *	near-off	near-on
music	non listening	listening
user	absent	absent

According to the state transitions previously mentioned, the new Q value is calculated as presented in Equations (6.16) and (6.17).

$$Q^{player}_{fun}(near\text{-}off, play) = \\ (1-\alpha)\cdot Q^{player}_{fun}(near\text{-}off, play) + \alpha \cdot \Big(r + \gamma \cdot V^{player}_{fun}(near\text{-}on)\Big) \tag{6.16}$$

$$V^{player}_{fun}(near\text{-}on) = \max_{a\in A_{player}} \Big(Q^{player}_{fun}(near\text{-}on, a)\Big) + \sum_{obj_m \neq player} \Delta Q^{obj_m}_{\max} \tag{6.17}$$

In this scenario, the collateral effects occur in the *music* object: once the *music player* is switched on, the *music* starts to listen and its related state changes from *non listening* to *listening* (Table 6.8). The new state of the object *music*, *listening*, has a very large value ($56,1831$) due to the fact that it is necessary for *dancing*, which is the action that satisfies the motivation of *fun*. In contrast, when the *music* is not listening, there is not possible action with *music* because it is not present. This results in a very elevated collateral effects value.

Table 6.8: Collateral effects due to the action *play music* in Scenario 3

$\mathbf{Object_m}$	$\mathbf{max_{a\in A_{obj_m}}\left(Q^{obj_m}(s_{t+1}, a)\right)}$	$\mathbf{max_{a\in A_{obj_m}}\left(Q^{obj_m}(s_t, a)\right)}$	$\mathbf{\Delta Q^{obj_m}_{max}}$
docking station	$Q^{station}_{fun}(unplugged, charge) = -17,0099$	$Q^{station}_{fun}(unplugged, charge) = -17,0099$	0
music	$Q^{music}_{fun}(listening, dance) = 56,1831$	$Q^{music}_{fun}(non\text{-}listening, -) = -$	56,1831
user	$Q^{user}_{fun}(absent, -) = -$	$Q^{user}_{fun}(absent, -) = -$	$-$
$\sum_{\mathbf{obj_m \neq player}} \mathbf{\Delta Q^{obj_m}_{max}}$			$56,1831$

Despite the low reward (-0.0566654) and the poor value of the new state related to the *music player* ($0,146101$), when the Q value is computed in this iteration, the previous Q value is already elevated. Even so, the large collateral effect makes it to increase even more (from $22,1503$ to $29,0072$).

In this scenario, the performed action is an appetitive one too.

Table 6.9: New Q value for Scenario 3

$\mathbf{Q^{player}_{fun}(near\text{-}off, play)}$	**reward**	$\mathbf{V^{player}(s_{t+1}) = V^{player}_{fun}(near\text{-}on)}$		new $\mathbf{Q^{player}_{fun}(near\text{-}off, play)}$
		$\max_{a\in A_{player}}\left(\mathbf{Q^{player}_{fun}(near\text{-}on, a)}\right)$	**Coll.Effects**	
$22,1503$	$-0,0566654$	$Q^{player}_{fun}(close\text{-}on, idle) = 0,146101$	$56,1831$	$29,0072$

Scenario 4

This last scenario is the counterpoint to the preceding scenario. Here, the collateral effects provoke a strong decrement in a Q value. This scenario corresponds to the iteration following the one presented in the Scenario 3.

In this case, the robot stops the *music player* which causes the state transitions shown in Table 6.10. Apart from the effects of this action in the *music player* object, the *music* object changes its state to *non listening*. This transition derives in a really negative value of collateral effects (Table 6.11) because *fun* is the dominant motivation and the best action (*dance*) cannot be executed without listening to music.

Table 6.10: State transitions due to the action *stop music* in Scenario 4

	anterior state (S_t)	**new state (S_{t+1})**
dominant motivation	fun	fun
docking station	unplugged	unplugged
music player *	near-on	near-off
music	listening	non listening
user	absent	absent

The equations computing the Q value for this scenario are Equations (6.18) and (6.19).

$$Q^{player}_{fun}(near\text{-}on, stop) =$$
$$(1-\alpha)\cdot Q^{player}_{fun}(near\text{-}on, stop) + \alpha \cdot \Big(r + \gamma \cdot V^{player}_{fun}(near\text{-}off)\Big) \tag{6.18}$$

$$V^{player}_{fun}(near\text{-}off) = \max_{a\in A_{player}} \Big(Q^{player}_{fun}(near\text{-}off, a)\Big) + \sum_{obj_m \neq player} \Delta Q^{obj_m}_{\max} \tag{6.19}$$

Table 6.11: Collateral effects due to the action *stop* in Scenario 4

$\mathbf{Object_m}$	$\mathbf{max_{a\in A_{obj_m}}\left(Q^{obj_m}\left(s_{t+1}, a\right)\right)}$	$\mathbf{max_{a\in A_{obj_m}}\left(Q^{obj_m}\left(s_t, a\right)\right)}$	$\mathbf{\Delta Q^{obj_m}_{max}}$
docking station	$Q^{station}_{unplugged,charge} = -17,0099$	$Q^{station}_{unplugged,charge} = -17,0099$	0
music	$Q^{music}_{fun}(non\,listening, -) = -$	$Q^{music}_{fun}(listening, dance) = 56,1831$	-56,1831
user	$Q^{user}_{fun}(absent, -) = -$	$Q^{user}_{fun}(absent, -) = -$	$-$
$\sum_{\mathbf{obj_m \neq player}} \mathbf{\Delta Q^{obj_m}_{max}}$			$-56,1831$

The corresponding Q value already has a low value ($-1,9285$) (Table 6.12). However, although the value of the new state in relation to the *music player* is quite high ($29,0072$), the very low value of the collateral effects ($-56,1831$) and the scarce reward (-0.0566635) reduce this Q value up to $-7,88916505$. This is because, as said before, *music* is required to dance and, therefore, to have fun. Without it, in this case, it is impossible to satisfy the need of fun.

Table 6.12: New Q value for Scenario 4

$\mathbf{Q^{player}_{fun}(near\text{-}on, stop)}$	**reward**	$\mathbf{V^{player}(s_{t+1}) = V^{player}_{fun}(near\text{-}off)}$		new $\mathbf{Q^{player}_{fun}(near\text{-}on, stop)}$
		$\max_{a \in A_{player}} \left(\mathbf{Q^{player}_{fun}(near\text{-}off, a)}\right)$	**Coll.Effects**	
$-1,9285$	$-0,0566635$	$Q^{player}_{fun}(near\text{-}off, play) = 29,0072$	$-56,1831$	$-7,88916505$

6.3 Enhancing the learning process

As exposed before, learning is achieved by the robot through interaction in the real world of a laboratory. Moreover, during learning, the actions are randomly selected. This random selection is based on the theory that all situations must be experienced an infinite number of times for the learning algorithm to achieve convergence. This leads to unfeasible experiments in terms of their duration.

In order to be able to carry out full learning sessions, the reduced state space and the Object Q-Learning have been considered. However, this is not enough for experiments in the real world, Consequently, two novel mechanisms have been included:

1. Well-balanced Exploration

2. Amplified Reward

Both are intended for speeding up the learning process reducing the duration of the learning sessions. Following, they are analyzed.

6.3.1 Well-balanced Exploration

During exploration, due to the random selection of actions, some states can remain unexplored for long periods. In order to solve this problem, from time to time, these unexplored states are enforced to be discovered.

This idea is exposed in Figure 6.2: at some point, the robot is artificially transferred to a new state s' which has not been explored enough. This "guided" transition is not considered as an iteration in the learning process because it is not the "natural" result of an action selected by the robot itself.

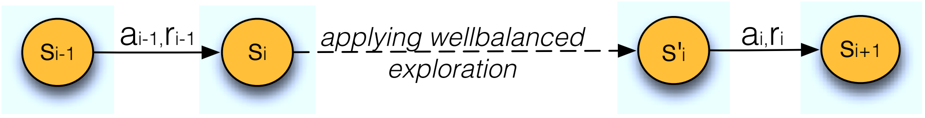


Figure 6.2: Well-balanced Exploration schematic

This idea has to be applied to the particular state space of this work. Considering the ideas presented in Section 6.2.1 , the state of the robot is composed of internal and external states (Equation (6.1)). The inner state is determined by the dominant motivation at each iteration. The motivations grow due to the drive linked to each one or to the external stimuli. As a result of the random selection of actions during learning, it could happen that the required external stimuli for a particular motivation are never presented or attained; or actions that satisfy a drive are always executed when its associated motivation is not the dominant one. Moreover, drives evolve at different rates. Thereupon, the motivations associated to the slowest drives are less likely to become the dominant motivation. For all these reasons, the proper behaviors that have to be exhibited with some slower motivations could not be properly learned in a reasonable amount of time.

In particular, in the presented implementation, the *relax* motivation is affected by this problem. Its associated drive, *calm*, is the slowest one and the robot has to be *listening* the *music* to make this drive increases. For this reasons, *relax* will hardly be the dominant motivation.

For promoting these slow motivations, it has been developed a mechanism where, every f iterations, the least frequent dominant motivation is promoted. Promoting a motivation means that the drive linked to the motivation is artificially saturated. This implies that the drive value reaches its maximum value. Therefore, the promoted motivation will easily reach the dominance over the rest of the motivations. As a consequence, the new state is likely to be related to this promoted motivation and the associated behavior will be explored.

When a motivation is promoted, the transition from the previous state to the new situation where its drive is artificially saturated is not considered by the learning algorithm. Otherwise, unreal effects of actions would have been taken into account and included in the learned policy.

In the experiments, f is set to 15 iterations. The whole process is schematized in Algorithm 6.2.

Algorithm 6.2 Well-balanced Exploration: promoting motivations

```
Require: $iter \leftarrow total\ number\ of\ iterations$
Require: $f \leftarrow frequency\ to\ promote\ the\ least\ frequent\ dominant\ motivation$
while $robot\ is\ learning$ do
    if $iter \mod f = 0$ then
        $m \leftarrow least\ frequent\ dominant\ motivation$
        $d \leftarrow drive\ associated\ to\ m$
        $d\ is\ saturated$                                  ▷ promoting motivation
        $Set\ flag\ to\ ignore\ this\ iteration\ at\ learning$
    end if
    $iter = iter + 1$
end while
```

Promoting motivations forces to explore all the possible internal states (dominant motivation) an acceptable number of times, so the exploration of dominant motivations is balanced. Thus, the experiment length can be drastically reduced.

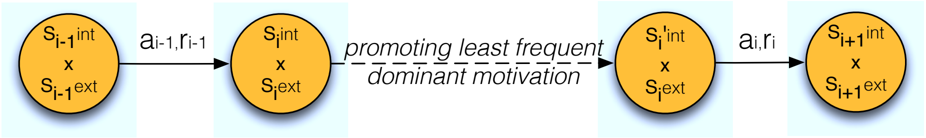


Figure 6.3: Well-balanced Exploration applied to the internal state

In this work, Well-balanced Exploration has been applied considering just unusual internal states (Figure 6.3). External states are explored enough and this technique has not been applied to them. Nevertheless, in other works where the number of objects is much higher, the same approach can also be applied to the external state in order to improve the learning time (Figure 6.4). In this case, if the state related to an item has not been enough explored, it will try to force this state.

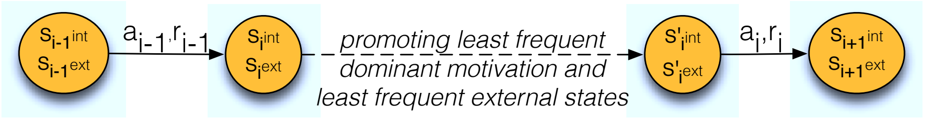


Figure 6.4: Well-balanced Exploration applied to internal and external states

As already mentioned, Well-balanced Exploration has been applied just to the internal state of the robot. However, these changes of internal state could required a change on external state too. For example, when the drive *relax* is promoted, considering the definition of this drive given in Chapter 4, the object *music* must be listening. Then, if the *music player* is off, it must be turned on. Therefore, the required transitions related to the external state are also forced. Then, in the experiments, Well-balanced Exploration will be guided by the internal state but external state transitions may be required too.

6.3.2 Amplified Reward

In order to identify as fast as possible the actions that satisfy the robot's needs, Amplified Reward has been implemented. As usual, living beings have been taken as the source of inspiration. Focusing on human beings, when a person is hungry and he eats, the benefit is really great. However, if this person is really thirsty and also hungry, eating does not provide the same level of benefit, but a smaller one. The benefits coming from satisfying the most urgent need is always huge. This is the idea behind the Amplified Reward mechanism.

In the interest of fostering this idea, positive rewards are amplified when the reward comes from correcting the drive corresponding to the dominant motivation. By means of back-propagation and the collateral effects, this amplified reward is transferred to the rest of the actions involved, even when several objects are concerned. Therefore, all actions required to satisfy a drive will be proportionally amplified: the farther the action is from satisfying the drive, the less amplified.

For example, if the robot needs to relax, it will learn that, first, it must approach the music player and, then, it stops the music. After music is muted, the need of relax is satisfied. Thus, the reward of this action is directly amplified. Approaching the music player is affected by this amplification due to the back-propagation occurring in the learning algorithm, but its intensity is lower.

In consideration of the previous ideas, the amplification is applied when the variation of wellbeing (the reward) is positive and this benefit is due to the reduction of the drive connected to the dominant motivation (the most urgent need). Mathematically, it is expressed as Equation (6.20).

$$\text{If } \Delta_a D_{dm} < 0 \text{ \& } r_a > 0 \text{ then } r \leftarrow r_a \cdot f_a \tag{6.20}$$

where $\Delta_a D_{dm}$ is the variation of the drive of the dominant motivation after executing action a. r_a means the reward obtained when action a has finished (this is the wellbeing increment), and r is the reward used by the learning algorithm. Finally, f_a is the amplification factor which determines the amount of augmentation applied to the reward. In the experiments, the amplification factor has been set to 3.

How amplified reward is applied during an iteration of the learning process can be seen on Figure 6.5. After action a has been executed, the obtained reward r_a is amplified if it positively affects the dominant motivation.

6.4 Summary

This chapter has presented the learning algorithm implemented in the robot Maggie. This algorithm is the Object Q-Learning which, together with the reduced state space, makes a great improvement in the learning time. In addition, the collateral effects allows to consider the interdependence among objects. Several detailed examples provide a clear understanding of the whole learning process to the reader.

Moreover, due to the fact that this work is implemented in a robotic platform, some modifications have been developed. The Well-balanced Exploration and the Amplified Reward provide good performance in the behavior learning task. The comparison with and without these novel techniques is presented in the experiments chapter, Chapter 8.

The learning process detailed in this chapter endows the robot with the capacity to properly learn the most convenient consummatory and appetitive actions, resulting in different

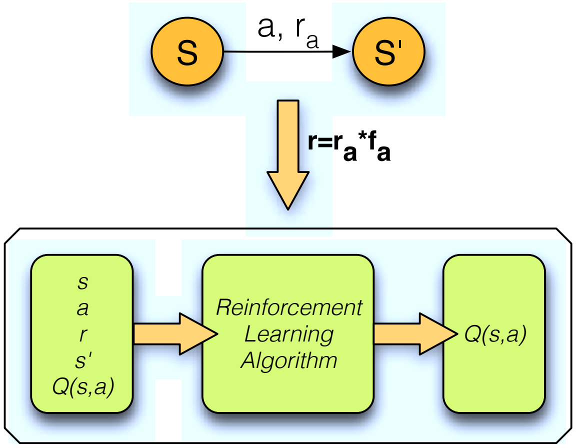


Figure 6.5: This diagram shows how Amplified Reward affects the learning process during an iteration

behaviors. The resulting behaviors after the learning process are analyzed in Chapter 9.

CHAPTER 7

Implementing the decision making system

7.1 Introduction

The DMS presented in Chapter 4 has been designed and implemented considering that it has to be as flexible as possible. This means that it has to be able to adapt to new requirements and configurations with a minimum effort.

Since this system runs in a distributed system, the required data are stored in a relational database where they can be easily accessed. It has been designed taking into account the next principles:

- simplicity: tables are kept as easy as possible.
- conciseness: redundant data is avoided.
- information: all the required data has to be represented on the logical scheme.
- logical independence: software must be robust enough to accept the modifications in the tables

This section presents the technical design of the whole system, from the decision making system to the available actions (skills) implemented in this work. First, the DMS design and how it is achieved are justified. Then, the robot's skills involved are commented.

7.2 Decision Making System database design

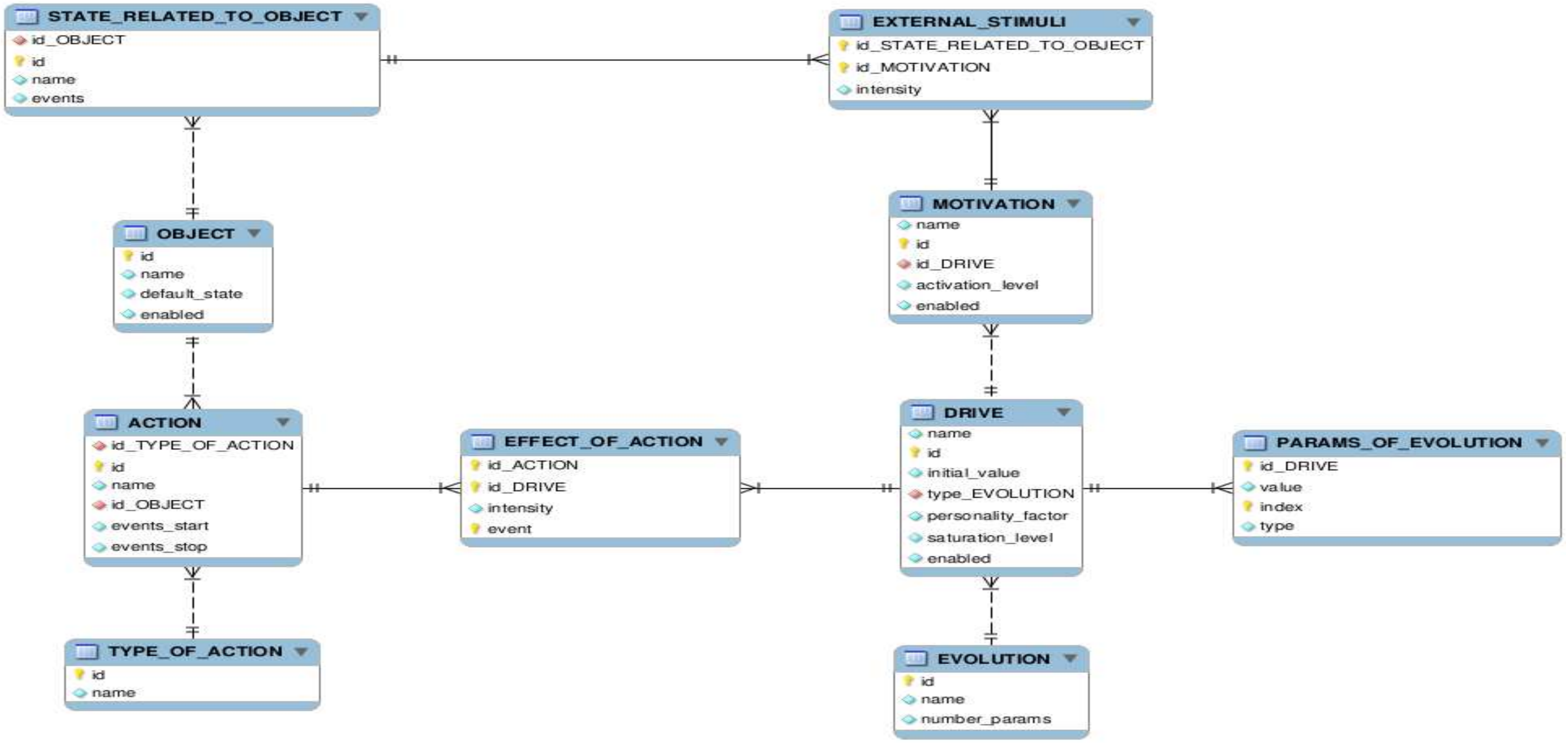
All data required for setting the parameters of the DMS is stored in a database.

The DMS database design can be observed in Figure 7.1. This figure represents the entity-relationship model of the database. Following, each element of the design is commented and justified. The design is formed of entities and attributes which are represented as tables in a database. Besides, relationships among entities represent how they are interconnected and their associations. The database engine used in this implementation is MySQL [6], the famous open source relational database management system.

The database design presented in this section is intended to contain all data required by the DMS proposed in this dissertation in order to perfectly decide and execute the most appropriate action at each moment. All entities (referred as tables when implementing the entity-relationship diagram) and their relationships are described. Each table is composed of entries in the table, and each entry is an instance of an entity.

Following all entities, their attributes, and their relationships are described.

[6] The world's most popular open source database (www.mysql.com)



▭ **Entity** a person, place or thing about we want to collect and store multiple instances of data. It has a name, which is a noun.

◇ **Attribute** features which describe the data we are interested in storing.

>+ **One or more** the instance can be once or more times associated.

++ **Exactly one** the instance can be exactly once associated.

🔑 **Primary key** attributes which uniquely identify one instance of an entity.

◆ **Foreign key** field in a relational table that matches the primary key column of another table. This key tells the relational database how the tables are related.

— **Identifying relationship** a foreign key is part of the primary key.

---- **Non-identifying relationship** foreign key is an attribute, it is not part of the primary key.

Figure 7.1: Database Entity-Relationship diagram

Motivations The *MOTIVATION* entity represents a motivation running in the system. Each one has a name and a unique id which undoubtedly differentiates this motivation from others. As already explained, the motivation value is determined by the internal and external stimuli. This is represented by the relationships with the *DRIVE* and *EXTERNAL_STIMULI* tables. Each motivation is related to just one drive, then, this relationship is limited to exactly one drive per motivation, one internal stimulus per motivation. In relation to external stimuli, the *EXTERNAL_STIMULI* table is used to retain the relationship between motivations and the items acting as external stimuli. These objects affect the corresponding motivation when they are present; this is, there is an intrinsic state in relation with the objects which determine the activation of the stimulus. Hence, the *EXTERNAL_STIMULI* table stores all states related to the objects which act as the external stimuli. Moreover, a particular motivation can be affected by several external stimuli. Therefore, the multiplicity of this relationship is one motivation to one-or-more external stimuli. Besides, there is an attribute related to the activation level of each motivation, *activation_level*.

Finally, the last attribute is called *enabled* and it is used just for debugging purposes: if *enabled* is *false*, this motivation is not considered during experiments.

Drives The *DRIVE* table is one of the key elements in the system. Entries in this table store all data related to drives and how they change as time passes. Each entry has a name, which is a human understandable reference, and an identifier. The *initial_value* attribute correspond to the initial value of the drive when the robot's *life* starts, *personality_factor* represents the personality factor which ponders the relevance of each drive in the robot's wellbeing computation. The saturation level of a drive is associated to the *saturation_level* attribute. Like it has been shown in the *MOTIVATION* table, each entry has an attribute called *enabled* which easily allows to activate or deactivate a particular drive; it is mainly used for debugging purposes too.

Evolution of drives Besides, the value of each drive changes. How the value of the drives evolves is determined by a function and the parameters that define that function. All this information is obtained from the tables *EVOLUTION* and *PARAMS_OF_EVOLUTION* respectively.

Every drive updates its value according to a particular function. The logic of all possible functions are implemented in code and their attributes are available in the table *EVOLUTION*. Each function has a different identifier and a name, in order to easily remember it. The number of parameters required for the type of evolution is at the *number_params* attribute. Each drive sets its *type_EVOLUTION* attribute to the corresponding evolution function identifier.

Considering the number of parameters of each function, it is possible to read the corresponding parameters of the desired drive from the *PARAMS_OF_EVOLUTION* table. All parameters for all evolution functions are stored in this table. Parameters associated to a particular drive are identified by the drive identifier (*id_DRIVE*) and its index in the function (*index*). In addition, these parameters can be interpreted as float numbers or text strings; this is identified by the *type* attribute: 1 corresponds to a string and 2 to a float. The parameter value itself is obtained from the *value* attribute. For example, if an evolution function requires a certain event, there will be a parameter of type string with the corresponding event. On the other hand, if the parameter relates to the increment of the drive per iteration, it will be a float value. All in all, once a drive has all the parameters required by its evolution function, it is ready to update its value as time goes.

Currently, available functions are limited to a finite set of hard-coded C++ functions which are linked to a drive evolution function id. However, new functions can easily be added if it is necessary.

Objects On the left side of Figure 7.1, tables related to items and their actions are presented. The first entity to mention, *OBJECT*, describes the objects the robot is able to interact with. That is, the objects which constitute the robot's world. Again, each entry in the table *OBJECT* has a name and an id. The *default_state* attribute is used to define the initial state or the state when an error occurs. The *enabled* attribute works as the homonymous attributes on previously commented tables.

States related to objects Objects have a finite and discrete set of states related to them which defines the situation of the robot in relation to the world. These states are used to determine the external state of the robot (Section 6.2.1). Data related to the states of the objects are represented in the *STATE_RELATED_TO_OBJECT* table: in order to clearly differentiate a state from other, it has a key attribute called *id* and a name (*name*) which describes it; *id_OBJECT* represents the identifier of the object this state is related with; and *events* attribute stores all the events and the associated parameters which are emitted when a transition to this state occurs. This attribute is a string formatted as follows:

$EVENT1 : PARAMETER1; EVENT2 : PARAMETER2; ...$

More than one event can determine a transition to the same state so several events (and their respective parameters) can be included in the same attribute. For example, the *music player* can be turned on with different commands, and, if the player is off, all these commands imply a transition from *close-off* to *close-on*.

External stimuli The key attribute of the *STATE_RELATED_TO_OBJECT* entity is also considered on the *EXTERNAL_STIMULI* table as a key. As previously stated, *EXTERNAL_STIMULI* entity represents the external stimuli for motivations. To define this relationship between states related to objects and motivations, both the state id (*id_STATE_RELATED_TO_OBJECT*) and the motivation id (*id_MOTIVATION*) are necessary. This two elements are enough to define an external stimulus but how much this stimulus influences the motivation is still required. This is defined by the *intensity* attribute. Each external stimulus is formed by one state related to an item which affects one motivation. Although, a motivation can be affected by several external stimuli. How the value of an external stimulus modifies the motivation has already been detailed in Equation (4.1).

Actions Objects in the world are items the robot can interact with. Therefore, the robot can perform a collection of actions with each object. Information required to execute this actions are compiled in the *ACTION* entity. Actions are identified by a unique id and a name. Each action is applied over a single object which is determined by its identifier in the *id_OBJECT* attribute.

Actions are implemented as skills in the AD architecture. Therefore, the most relevant information about actions is: how to activate and block the particular skill which performs that action. This data is saved in the *events_start* and *events_stop* attributes of the *ACTION* entity. The events and parameters that must be sent to activate or block the skill are saved in these attributes respectively.

Sometimes, the same action can be achieved by different skills. This implies that several events can be sent for initiating that action. For example, the robot could dance in many different manners, and each of these manners is activated by different events. This is considered in the implementation by formatting the *events_start* and *events_stop* attributes in the following way:

$EVENT1 : PARAMETER1; EVENT2 : PARAMETER2; ...$

When an action can be performed by several skills, the system randomly chooses one and emits the corresponding starting and blocking events.

Type of action There is an entity that defines the type of action: the *TYPE_OF_ACTION* entity. So far, the type of action just needs an id and a name; the logic under each type is coded into the software implementing the decision making system. Each entry of the corresponding table refers to different sort of actions. Then, this classification can be easily extended in the future by just adding new entries. The *ACTION* entity has an attribute named *id_TYPE_OF_ACTION* which must be set to an existing action type id.

In this implementation, two sorts of robot's actions were defined: "endogenous" which represent actions affecting the world and the robot itself, and "exogenous" which do not cause any effect and are used to perfectly perceived the consequences of actions not-executed by the robot (Section 4.4.2).

Effects of action Moreover, actions provoke effects on the robot's environment. These effects are: a change on the state of the robot in relation to objects; and a modification of an internal variable (a drive). The former effects are managed by skills in charge of monitoring the states related to objects in the world (Section 7.4). The later effects are defined by the *EFFECT_OF_ACTION* entity. One action can affect one drive, or different drives, and it could happen that a drive can be affected by several actions. Also, an action can have no effects over drives. The effects of the actions vary in their intensities. Each entry in the *EFFECT_OF_ACTION* table corresponds to the effect of an action over a drive: the *event* attribute corresponds to the events and parameters launched by an skill associated to the action, whose id corresponds to *id_ACTION*, indicating that the corresponding effect must be applied over the drive whose id is at the *id_DRIVE* attribute. The value (intensity) of the effect itself is determined in the *intensity* attribute. If an action *a* affects a drive *d* and the value of the effect is *e*, the resulting value of the drive *d* is:

$$\begin{array}{ll} \text{If } e = +N & \Rightarrow d = d + N \\ \text{If } e = -N & \Rightarrow d = d - N \\ \text{If } e = N & \Rightarrow d = N \\ \text{If } e = RESET & \Rightarrow d = d_{initial\_value} \end{array} \qquad (7.1)$$

where N is a number and $RESET$ is a key which identifies when the drive must be reset to its initial value ($d_{initial\_value}$) obtained from the *DRIVE* table.

7.3 Decision Making System class design

Data stored in the database have to be loaded into the software in order to be able to operate with them during robot's life time. In this work, an object-oriented approach has been considered and, therefore, several classes have been designed and implemented into C++ code.

First, a general view of the DMS class design is presented. In this initial view, the reader will get an overview of all elements and their relationships. Later, the main parts of the design will be detailed and clarified.

Two main areas can be distinguished:

(a) how to model the external world of the robot

(b) how to model the internal variables of the robot

These two areas can be observed on Figure 7.2.

In relation to the external world (the environment where the robot "lives"), the world is defined in terms of the items the robot is able to interact with, their possible states, and their potential actions. Then, several classes have been defined to manage all the objects. The *CObject* class defines all the data related to an object. Each object is endowed with a set of actions which can be executed by that object. Each action is modeled by the *CAction* class. In addition, depending on how an object is perceived (or it is not perceived) by the robot, it is said that the object is in a certain state. The required data for each state of each object is included as an instance of the *CRelatedState* class.

Now, focusing on the robot itself, its internal variables are also defined as classes. As presented in Chapter 4, the inner needs are presented as drives which are modeled by the *CDrive* class. The evolution function for each drive is declared in an instance of the *CTimeEvolution* class. Then, motivations are also a key element: each drive is linked to a motivation. This and other properties of motivations are gathered in the *CMotivation* class.

The relationship between the external and the internal world of the robot are determined by the external stimuli and the effects of actions. This is reflected in Figure 7.2 by the *CExternalStimulus* and *CEffectOfAction* classes. The first one relates objects which alter motivations to the motivations themselves. The latter is used to describe how an action affects a particular drive.

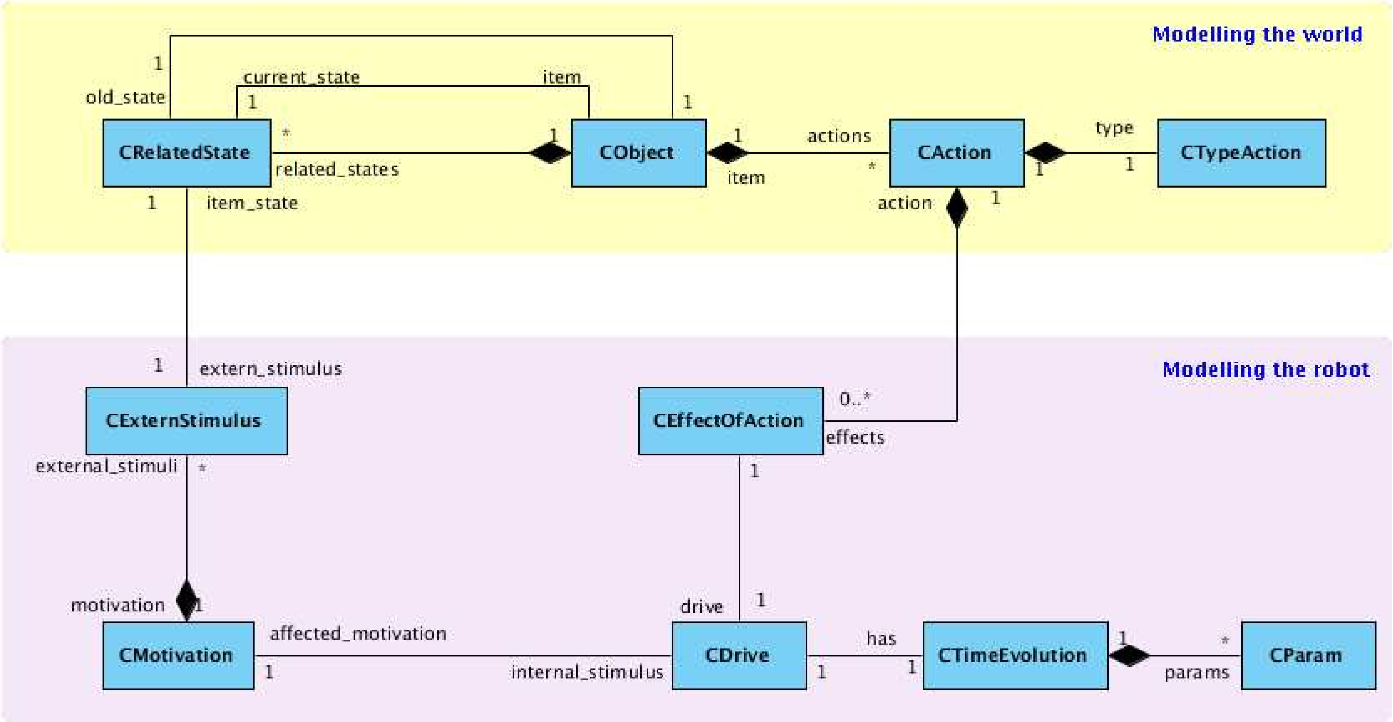


Figure 7.2: General view of the main classes which define the robot's world (external and internal)

7.3.1 The external robot's world class design

After the outline of the DMS class design has been presented, each part is detailed. First, Figure 7.3 presents the full class diagram with the relationships among all classes related to the external robot's world.

In this work, lots of elements require a unique identification. Then, all classes that need it inherit from the *CId* class which provides the operations required for managing an identifier and a name.

About the *CObject* class, it has a set of states which corresponds to the possible states for each object (the *related_states* attribute). In the same way, each object has a set of actions which are accessed by the *actions* attribute.

Objects always are in a particular state in relation to the robot. Then, the current state and the previous one are referred by the *state* and *oldstate* attributes. When the state of an object is updated, these pointers are modified. Both values are used to define the state transitions for each item.

In the class *CAction*, the events for starting and stopping an action are in the *events_start* and *events_stop* attributes. Since more than one event can be used to start/stop the action, they are brought together in vectors of *CEvent* objects. At the same time, each action keeps a pointer to its object (*item*); and the effects are stored as a vector of *CEffectOfAction* instances accessed through the *effects* field.

Since the same action can have different sort of effects, each effect has a different event which indicates when it has to be applied. This is considered in the class *CEffectOfAction* in the attribute *event_param*. This element relates the internal robot's world to the external one; then, it keeps track of the action which provokes the effect (the *action* attribute) and the affected drive (the *drive* attribute). The quantity of the effect is represented as an integer.

In relation to the states of an object, each state remembers the object it is linked to by the *item* attribute and, also, the event which determines a new transition to this state. Since states for the same object are incompatible (in this implementation an object cannot be in two different states at the same time), one event is enough to determine a transition, and an "exit" event is not necessary. As said, more than one event can determine the transition to the state, so a vector of events is considered at each state.

Besides, certain objects act as external stimuli; then, there is a connection between states and external stimuli which is represented in the class *CRelatedState* class by a vector of pointers to the external stimuli where the state takes part.

External stimuli connect the objects and the motivations. This is implemented by means of the class *CExternStimulus*. This class relates a particular state of an item (*item_state* attribute) to a specific motivation (*motivation* attribute). The intensity of this stimulus is read from the *value* attribute.

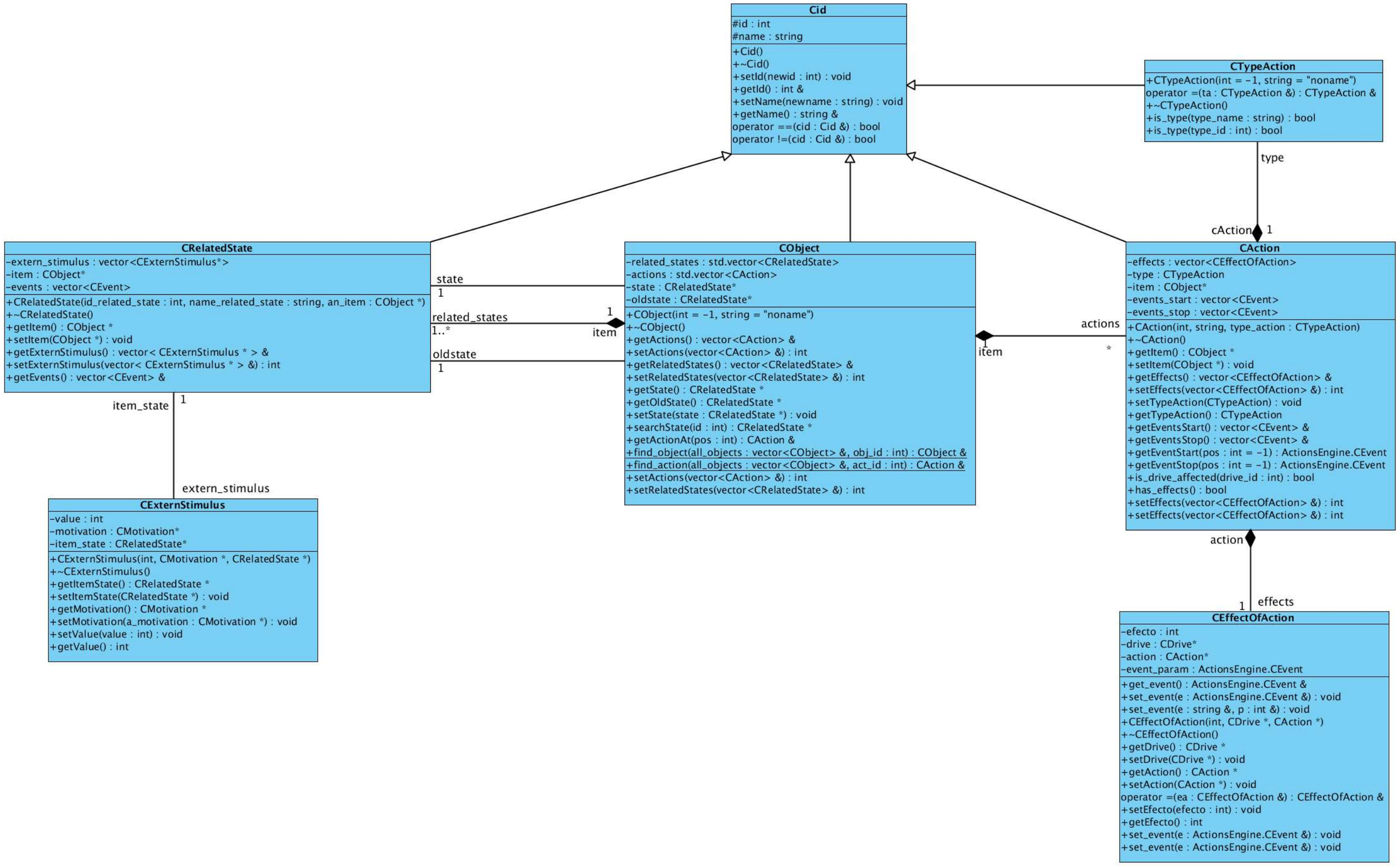

Cid
#id : int
#name : string
+Cid()
+~Cid()
+setId(newid : int) : void
+getId() : int &
+setName(newname : string) : void
+getName() : string &
operator ==(cid : Cid &) : bool
operator !=(cid : Cid &) : bool
CTypeAction
+CTypeAction(int = -1, string = "noname")
operator =(ta : CTypeAction &) : CTypeAction &
+~CTypeAction()
+is_type(type_name : string) : bool
+is_type(type_id : int) : bool
type
cAction
1
CRelatedState
-extern_stimulus : vector<CExternStimulus*>
-item : CObject*
-events : vector<CEvent>
+CRelatedState(id_related_state : int, name_related_state : string, an_item : CObject *)
+~CRelatedState()
+getItem() : CObject *
+setItem(CObject *) : void
+getExternStimulus() : vector< CExternStimulus * > &
+setExternStimulus(vector< CExternStimulus * > &) : int
+getEvents() : vector<CEvent> &
state
1
related_states
1..*
1
item
oldstate
1
item_state
1
extern_stimulus
CObject
-related_states : std.vector<CRelatedState>
-actions : std.vector<CAction>
-state : CRelatedState*
-oldstate : CRelatedState*
+CObject(int = -1, string = "noname")
+~CObject()
+getActions() : vector<CAction> &
+setActions(vector<CAction> &) : int
+getRelatedStates() : vector<CRelatedState> &
+setRelatedStates(vector<CRelatedState> &) : int
+getState() : CRelatedState *
+getOldState() : CRelatedState *
+setState(state : CRelatedState *) : void
+searchState(id : int) : CRelatedState *
+getActionAt(pos : int) : CAction &
+find_object(all_objects : vector<CObject> &, obj_id : int) : CObject &
+find_action(all_objects : vector<CObject> &, act_id : int) : CAction &
+setActions(vector<CAction> &) : int
+setRelatedStates(vector<CRelatedState> &) : int
actions
1
item
*
CAction
-effects : vector<CEffectOfAction>
-type : CTypeAction
-item : CObject*
-events_start : vector<CEvent>
-events_stop : vector<CEvent>
+CAction(int, string, type_action : CTypeAction)
+~CAction()
+getItem() : CObject *
+setItem(CObject *) : void
+getEffects() : vector<CEffectOfAction> &
+setEffects(vector<CEffectOfAction> &) : int
+setTypeAction(CTypeAction) : void
+getTypeAction() : CTypeAction
+getEventsStart() : vector<CEvent> &
+getEventsStop() : vector<CEvent> &
+getEventStart(pos : int = -1) : ActionsEngine.CEvent
+getEventStop(pos : int = -1) : ActionsEngine.CEvent
+is_drive_affected(drive_id : int) : bool
+has_effects() : bool
+setEffects(vector<CEffectOfAction> &) : int
+setEffects(vector<CEffectOfAction> &) : int
action
1
effects
CExternStimulus
-value : int
-motivation : CMotivation*
-item_state : CRelatedState*
+CExternStimulus(int, CMotivation *, CRelatedState *)
+~CExternStimulus()
+getItemState() : CRelatedState *
+setItemState(CRelatedState *) : void
+getMotivation() : CMotivation *
+setMotivation(a_motivation : CMotivation *) : void
+setValue(value : int) : void
+getValue() : int
CEffectOfAction
-efecto : int
-drive : CDrive*
-action : CAction*
-event_param : ActionsEngine.CEvent
+get_event() : ActionsEngine.CEvent &
+set_event(e : ActionsEngine.CEvent &) : void
+set_event(e : string &, p : int &) : void
+CEffectOfAction(int, CDrive *, CAction *)
+~CEffectOfAction()
+getDrive() : CDrive *
+setDrive(CDrive *) : void
+getAction() : CAction *
+setAction(CAction *) : void
operator =(ea : CEffectOfAction &) : CEffectOfAction &
+setEfecto(efecto : int) : void
+getEfecto() : int
+set_event(e : ActionsEngine.CEvent &) : void
+set_event(e : ActionsEngine.CEvent &) : void

Figure 7.3: Detailed UML class diagram with all classes involved in the external robot’s world

7.3.2 The inner robot's world class diagram

In relation to the inner variables of the robot, Figure 7.4 presents the detailed diagram of all classes involved in it. As explained before, the class *Cid* is inherited by all elements which need to be uniquely identified.

As seen in Figure 7.4, the main class is the one related to drives: the class *CDrive*. The value of a drive is stored at the attribute called *value*. Initially, its first value is read from the *initial_value* attribute and its maximum value is fixed in the *saturation_level* field. The way a drive influences the robot's wellbeing is rated by the *personality_factor* attribute (Equation 4.7).

How the value of a drive is updated is defined by the object referred by the *time_evolution* pointer. This pointer refers to an object of class *CTimeEvolution*. This class keeps the info required for each possible function: the name and id of the function, and the number of parameters required (*number_of_params*). The parameters themselves are stored as a collection of *CParam* instances at the *params* vector. Each *CParam* object has the type of parameter (*param_type*), the value of the parameter (*str_value* or *float_value*, depending on the type of value), and its position in the function (*index*). The proper implementation of the functions are hard-coded. The available functions for the evolution of drives are:

- linear: the drive evolves according to a linear function
- step: the drive evolves as a rectangular pulse.
- constant: the drive has a constant value, so its value does not change.
- interpolate: a value obtained from STM is interpolated into a determined range.
- linear according to a state: the drive evolves according to a linear function just if the robot is in a particular state.
- linear with two rates: this is a linear function with two different slopes.
- linear with two rates according to a state: similar to the previous one, but the drive is updated just when the robot is in a particular state.
- by value: this is a step function where the step is determined by a value read from STM.
- by event: this is a step function where the step is determined according to an event.

Drives are affected by actions executed in the world. This is taken into consideration by the *CEffect_of_action* class where actions and their effects over the drives are defined.

Drives are linked to motivations. This is represented in the *CDrive* class by means of the *affected_motivations* attribute, where the motivations which are affected by the drive are pointed. Likewise, the motivation class (*CMotivation*) keeps a reference to its internal stimulus by the pointer *internal_stimulus*. In addition, *CMotivation* stores all needed data for a motivation: the activation level, and the internal and the external stimuli. The *extern_stimulus* attribute represents the items in the world which affects the motivation. This is a vector of *CExternStimulus* objects because, theoretically, a motivation can be influenced by more than one drive.

In order to manage all these data, during the robot's life, this info is loaded into three variables which will be accessed by the DMS software. These variables are declared in the following way:

```
vector<CDrive> drives; //robot’s drive data
vector<CMotivation> motivations; //robot’s motivations data
vector<CObject> items; //objects in the robot’s world data
```

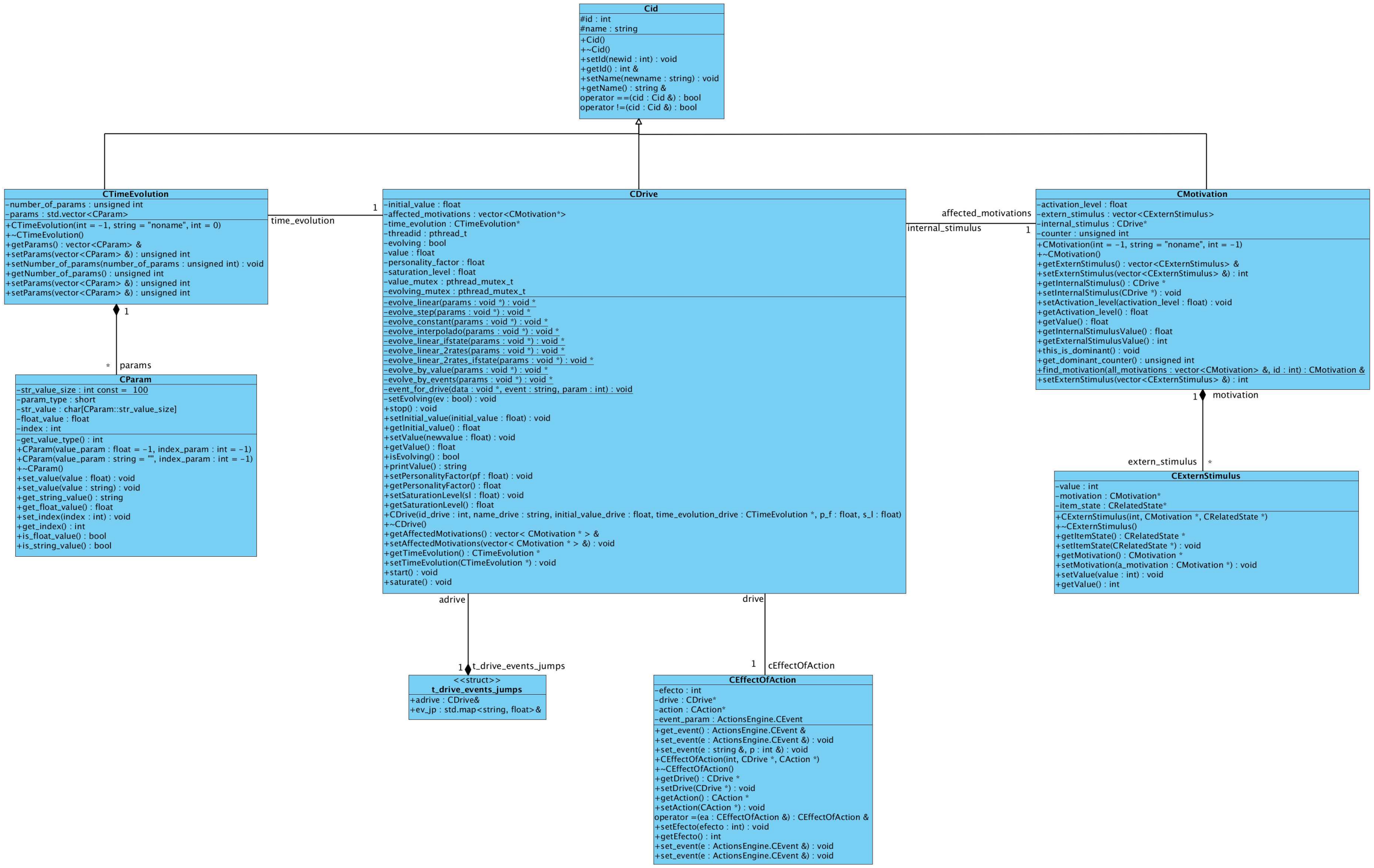

Cid
#id : int
#name : string
+Cid()
+~Cid()
+setId(newid : int) : void
+getId() : int &
+setName(newname : string) : void
+getName() : string &
operator ==(cid : Cid &) : bool
operator !=(cid : Cid &) : bool
CTimeEvolution
-number_of_params : unsigned int
-params : std.vector<CParam>
+CTimeEvolution(int = -1, string = "noname", int = 0)
+~CTimeEvolution()
+getParams() : vector<CParam> &
+setParams(vector<CParam> &) : unsigned int
+setNumber_of_params(number_of_params : unsigned int) : void
+getNumber_of_params() : unsigned int
+setParams(vector<CParam> &) : unsigned int
+setParams(vector<CParam> &) : unsigned int
time_evolution
1
1
*
params
CParam
-str_value_size : int const = 100
-param_type : short
-str_value : char[CParam::str_value_size]
-float_value : float
-index : int
-get_value_type() : int
+CParam(value_param : float = -1, index_param : int = -1)
+CParam(value_param : string = "", index_param : int = -1)
+~CParam()
+set_value(value : float) : void
+set_value(value : string) : void
+get_string_value() : string
+get_float_value() : float
+set_index(index : int) : void
+get_index() : int
+is_float_value() : bool
+is_string_value() : bool
CDrive
-initial_value : float
-affected_motivations : vector<CMotivation*>
-time_evolution : CTimeEvolution*
-threadid : pthread_t
-evolving : bool
-value : float
-personality_factor : float
-saturation_level : float
-value_mutex : pthread_mutex_t
-evolving_mutex : pthread_mutex_t
-evolve_linear(params : void *) : void *
-evolve_step(params : void *) : void *
-evolve_constant(params : void *) : void *
-evolve_interpolado(params : void *) : void *
-evolve_linear_ifstate(params : void *) : void *
-evolve_linear_2rates(params : void *) : void *
-evolve_linear_2rates_ifstate(params : void *) : void *
-evolve_by_value(params : void *) : void *
-evolve_by_events(params : void *) : void *
-event_for_drive(data : void *, event : string, param : int) : void
-setEvolving(ev : bool) : void
+stop() : void
+setInitial_value(initial_value : float) : void
+getInitial_value() : float
+setValue(newvalue : float) : void
+getValue() : float
+isEvolving() : bool
+printValue() : string
+setPersonalityFactor(pf : float) : void
+getPersonalityFactor() : float
+setSaturationLevel(sl : float) : void
+getSaturationLevel() : float
+CDrive(id_drive : int, name_drive : string, initial_value_drive : float, time_evolution_drive : CTimeEvolution *, p_f : float, s_l : float)
+~CDrive()
+getAffectedMotivations() : vector< CMotivation * > &
+setAffectedMotivations(vector< CMotivation * > &) : void
+getTimeEvolution() : CTimeEvolution *
+setTimeEvolution(CTimeEvolution *) : void
+start() : void
+saturate() : void
affected_motivations
internal_stimulus
1
CMotivation
-activation_level : float
-extern_stimulus : vector<CExternStimulus>
-internal_stimulus : CDrive*
-counter : unsigned int
+CMotivation(int = -1, string = "noname", int = -1)
+~CMotivation()
+getExternStimulus() : vector<CExternStimulus> &
+setExternStimulus(vector<CExternStimulus> &) : int
+getInternalStimulus() : CDrive *
+setInternalStimulus(CDrive *) : void
+setActivation_level(activation_level : float) : void
+getActivation_level() : float
+getValue() : float
+getInternalStimulusValue() : float
+getExternalStimulusValue() : int
+this_is_dominant() : void
+get_dominant_counter() : unsigned int
+find_motivation(all_motivations : vector<CMotivation> &, id : int) : CMotivation &
+setExternStimulus(vector<CExternStimulus> &) : int
1
motivation
extern_stimulus
*
CExternStimulus
-value : int
-motivation : CMotivation*
-item_state : CRelatedState*
+CExternStimulus(int, CMotivation *, CRelatedState *)
+~CExternStimulus()
+getItemState() : CRelatedState *
+setItemState(CRelatedState *) : void
+getMotivation() : CMotivation *
+setMotivation(a_motivation : CMotivation *) : void
+setValue(value : int) : void
+getValue() : int
adrive
drive
1
t_drive_events_jumps
1
cEffectOfAction
<<struct>>
t_drive_events_jumps
+adrive : CDrive&
+ev_jp : std.map<string, float>&
CEffectOfAction
-efecto : int
-drive : CDrive*
-action : CAction*
-event_param : ActionsEngine.CEvent
+get_event() : ActionsEngine.CEvent &
+set_event(e : ActionsEngine.CEvent &) : void
+set_event(e : string &, p : int &) : void
+CEffectOfAction(int, CDrive *, CAction *)
+~CEffectOfAction()
+getDrive() : CDrive *
+setDrive(CDrive *) : void
+getAction() : CAction *
+setAction(CAction *) : void
operator =(ea : CEffectOfAction &) : CEffectOfAction &
+setEfecto(efecto : int) : void
+getEfecto() : int
+set_event(e : ActionsEngine.CEvent &) : void
+set_event(e : ActionsEngine.CEvent &) : void

Figure 7.4: Detailed class diagram with all classes involved in the internal robot's world

7.4 How the external state is perceived

In this work, the robot perceives its environment in terms of objects around it. So, the position or the state of the robot related to these objects is crucial for the robot. For example, if the robot does not know that a person is close to it, it will never try to interact with that person. Even worse, if we consider the navigation system and the person is not detected as a close obstacle, the robot could collide with the person.

One of the key elements in this system is the detection of robot's state transitions in relation to an object. Recalling, these states represent the position of the items in the robot's world in relation to the robot itself. Each object is monitored by a skill (or several skills) which informs about any change in the state of the object in relation to the robot. The new states are notified by events which are received at a central monitoring skill. When a particular monitoring skill detects a transition to a new state, it sends the corresponding event and the attached parameter corresponds to the identifier of the new state (*N1*...*N5* in Figure 7.5). The central monitoring skill is in charge of composing the external state of the robot considering the states in relation to all the objects. The resulting external state is communicated to the DMS. The communication among all these elements is depicted in Figure 7.5.

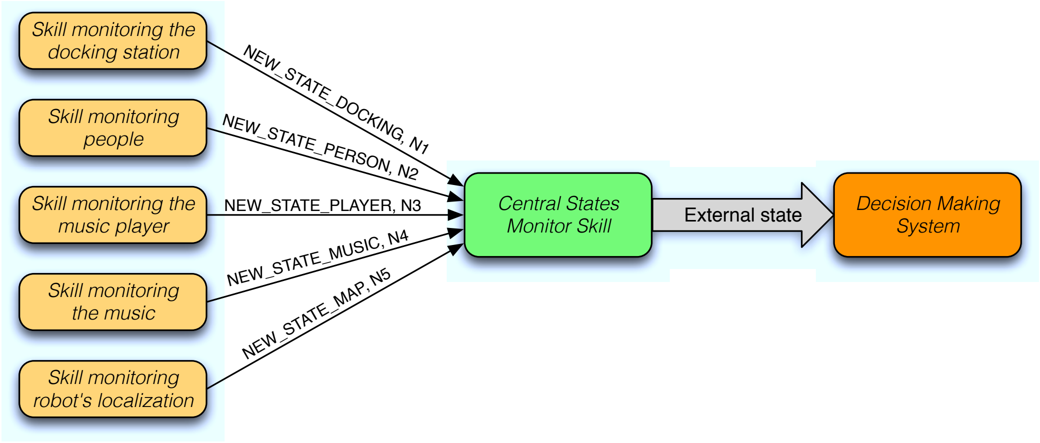

Skill monitoring the docking station
Skill monitoring people
Skill monitoring the music player
Skill monitoring the music
Skill monitoring robot's localization
NEW_STATE_DOCKING, N1
NEW_STATE_PERSON, N2
NEW_STATE_PLAYER, N3
NEW_STATE_MUSIC, N4
NEW_STATE_MAP, N5
Central States Monitor Skill
External state
Decision Making System

Figure 7.5: Skills involved in monitoring the external state

Next, how the *Central State Monitor Skill* works is presented. Figure 7.6 shows its flow chart. This is not a cyclic skill, but a skill which works by events. In this case, since the skills that monitor the individual objects employs events to communicate any change on the state, this skill is subscribed to all possible events and it filters the events related to the external state. After an event is received by this skill, it checks if the event is relevant for the state related to an object. The data related to these events are obtained from the database. Then, the current state is updated, and the external state is formed and written

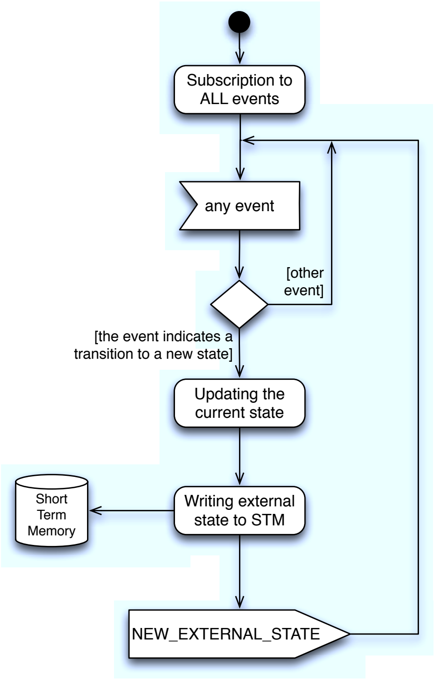

Subscription to ALL events
any event
[other event]
[the event indicates a transition to a new state]
Updating the current state
Short Term Memory
Writing external state to STM
NEW_EXTERNAL_STATE

Figure 7.6: Main process of the *State Monitor Skill*

to STM. This is notified to the rest of the elements by the `NEW_EXTERNAL_EVENT` event. Consequently, any element which needs to access the external state can read it, and it is informed about any change.

The system is always listening for new events and no one is lost thanks to the event manager system. It is endowed with queues which are in charge of managing all the incoming events.

In order to achieve a high performance and reliability, there are independent skills watching all the items in the robot's world. Following, a brief description about how each item is detected is presented. Then, a detailed description of the skills involved in these detections are explained.

The objects to be sensed are:

The docking station. The charger can be perceived by the data acquisition board which provides enough information to discriminate between when the robot is plugged and when it is unplugged.

A user. This "object" is perceived by the combination of two technics: a middle-range sensor, based on bluetooth, and a short-range sensor, based on RFID technology.

The music. In order to identify when the music is being played, there is a skill that remembers the last commands sent to the player, so it is able to determine if the music has been set on or off.

The music player. By means of the navigation system, the skill knows where the robot is and if it is far o close to the music player. In addition, it also uses the previous presented skill in order to determine when the robot is close to the player, with the music on or off.

The skills involved on the perception of the external state are: the location monitor skill, the music player control skill, the music player sensor skill, the docking station sensor skill, the bluetooth discoverer skill, and the rfid discoverer skill.

7.4.1 The location monitor

This skill has been designed to provide an easy interface with the navigation system implemented in the robot. This skill reads the geometric information of well-known locations from an XML file and they are employed for the internal use of the skill.

The skill has a dual task:

(a) it provides an easy high-level interface to send commands to the navigation system

(b) it monitors the position of the robot in the world

For example, if there is a position referred as *in front of the music player*, this skill translates the high-level command *go to the music player* to lower geometric commands which are managed by other skills running in the AD architecture. Moreover, the moment when the robot has reached this location is also notified to the rest of the architecture, .

An example of an XML file of positions is shown below. The required data for a position are an identifier, a description of the location, and the $XY\theta$ coordinates.

Code listing 7.1: XML file describing a location

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<positions>
    <position>
                <!-- all fields are required -->

                <!-- id number -->
                <id>1</id>

                <!-- short description about the position -->
                <description>in front of the docking station</description>

```

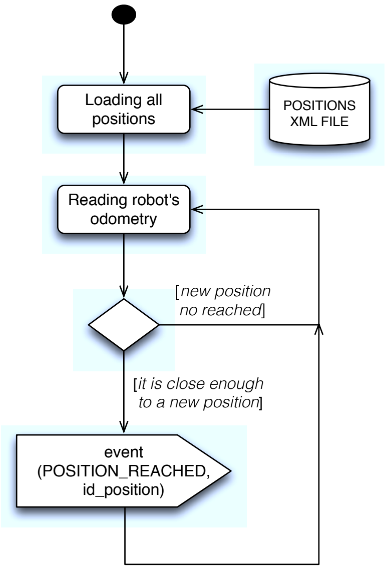

Loading all positions
POSITIONS XML FILE
Reading robot's odometry
[new position no reached]
[it is close enough to a new position]
event (POSITION_REACHED, id_position)

Figure 7.7: Activity diagram of geometric position monitoring

```
                <!-- x value -->
                <x_value>-0.267956</x_value>

                <!-- y value -->
                <y_value>-2.28114</y_value>

                <!-- theta value -->
                <theta_value>-1.46912</theta_value>
    </position>
</positions>
```

This skill is permanently monitoring the robot's position for notifying the transitions from a location to another location. The process is shown on Figure 7.7. Initially, all well-known position are read from the XML file. After that, the control loop starts. First, the odometric values are updated, and then, the current robot geometric position is compared with all the well-known positions. If the robot position is close enough to a well-known position, this is notified by sending the $POSITION\_REACHED$ event and the position identifier is attached.

Due to the noise in the sensors, a tolerance value is set before a new location is assigned to the robot. Therefore, the expression "... *is close enough...*" means that the robot position must be inside this tolerance error. For the application required in this work, the error on the x and y axes is set to $10cm$, and, for the θ coordinate, the tolerance is $0.2rad$ (about 11^{o}).

The positions defined in this work were three: *in front of the docking station*, *the center of the lab facing the door*, and *close enough to control the TV*.

The functionality previously enumerated as *(a)*, the high level interface with the navigation system, is discussed in Section 7.5.4.

7.4.2 The music player sensor

This skill is in charge of sensing and notifying the state of the music player in relation to the robot. It manages two kinds of information: a) the geometrical position of the robot, and b) the commands sent to the player. Both data are merged to determine the state related to the player.

Since the required information is coming from two independent sources of information, every time one of them is updated, the state of the player is updated too. Two independents threads monitor both sources of information.

Regarding the commands sent to the music player, the skill is listening all of them. It is subscribed to the event used to operate the music player (`COMMAND_TV`), and the command sent is obtained. Then, according to the current position of the robot, the state of the player is updated. The new state is notified emitting the corresponding event if it is different than the last state. The process is summarized in Figure 7.8(a)

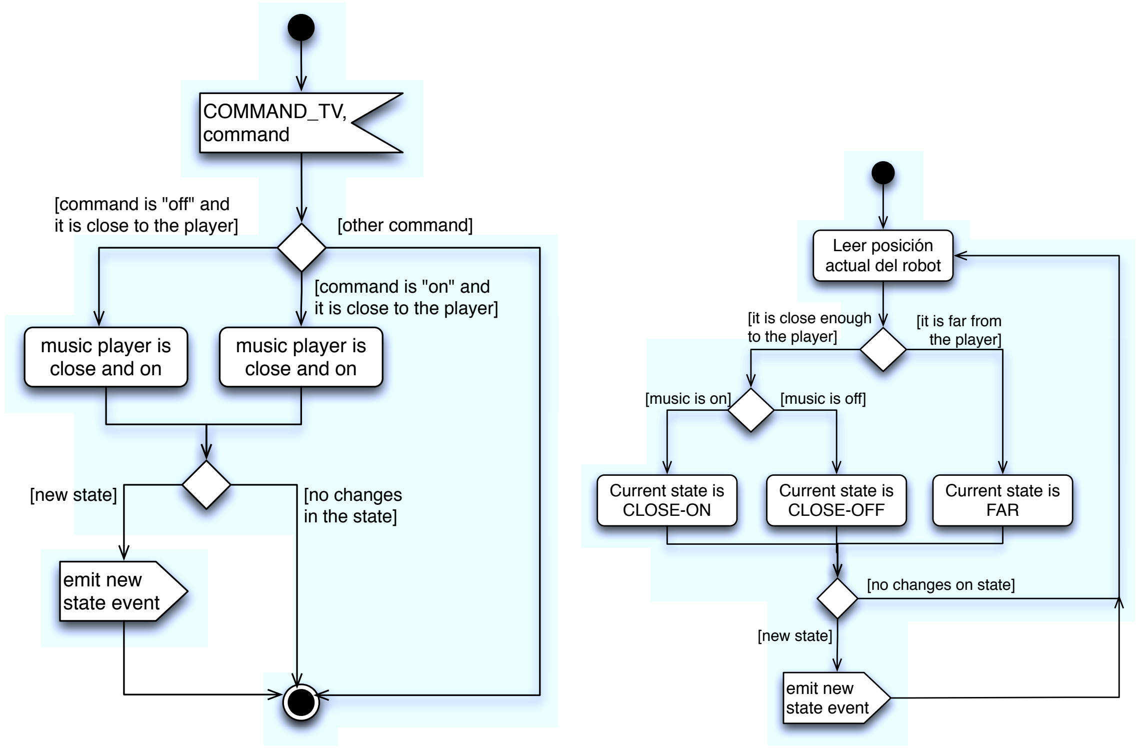
In relation to the position of the robot (Figure 7.8(b)), there is a control loop where the current position is read and it determines if the robot is close to or far from the player. This information is combined with the last command sent to the player and the final state of the music player is formed. If the state has changed, this is communicated by an event.

7.4.3 The docking station sensor

There are two possible states in relation to the docking station: plugged or unplugged. This skill senses this situation and notifies it to the rest of the architecture.

A data acquisition board reads the real voltage of the battery. When the robot is plugged, the voltage of the robot's battery reaches $27V$. Otherwise, this voltage is below $26V$. In the moment when the robot gets plugged/unplugged, there is peak/plunge on the voltage. This is the principle used by this skill to identify the transitions between both state.

The process is permanently running (Figure 7.9). Using mathematical methods, several consecutive readings of the voltage are used to approximate a straight line. The slope of this straight line is computed in order to determine if the robot is plugging or unplugging. The already mentioned peaks/plunges are reflected on the slope. Then, in order to determine the exact moment of plugging/unplugging the absolute value of the slope has to be over a threshold $L_{plugged}$. A positive slope means that the robot has just connected to the charger. A negative one implies a movement out of the docking station. In a formal way, it is


COMMAND_TV, command
[command is "off" and it is close to the player]
[other command]
[command is "on" and it is close to the player]
music player is close and on
music player is close and on
[new state]
[no changes in the state]
emit new state event
Leer posición actual del robot
[it is close enough to the player]
[it is far from the player]
[music is on]
[music is off]
Current state is CLOSE-ON
Current state is CLOSE-OFF
Current state is FAR
[no changes on state]
[new state]
emit new state event

(a) Monitoring the commands sent to the music player

(b) Monitoring the position of the robot

Figure 7.8: Activity diagrams of the music player sensor skill

expressed in Equation 7.2.

$$\begin{array}{l}\text{If } |m| > L_{plugged} \ \& \ m > 0 \Rightarrow \text{robot is plugged} \\ \text{If } |m| > L_{plugged} \ \& \ m < 0 \Rightarrow \text{robot is unplugged}\end{array} \tag{7.2}$$

where m is the value of the slope.

7.4.4 The bluetooth discoverer

The bluetooth discoverer is a skill intended to identify people around the robot. It is based on bluetooth technology which is power-class-dependent. The class of device determines its range. In Maggie, a class 2 device is on board the robot, so, its range is around 10 meters.

The idea is that each person is wearing his personal cellular phone, which is equipped with a bluetooth interface (most mobile phones are already equipped with this interface). Then, this interface is used to identify the people around the robot.

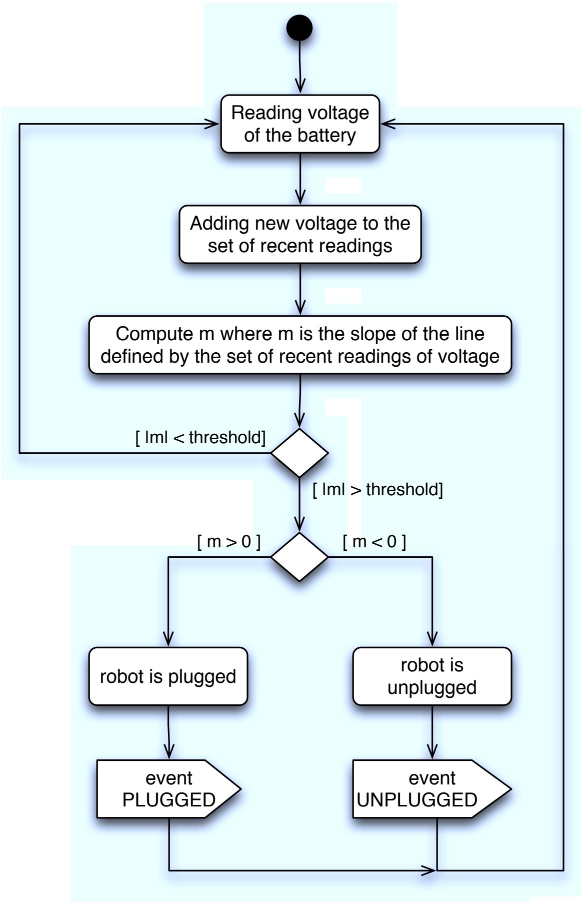

Reading voltage of the battery
Adding new voltage to the set of recent readings
Compute m where m is the slope of the line defined by the set of recent readings of voltage
[|m| < threshold]
[|m| > threshold]
[m > 0]
[m < 0]
robot is plugged
robot is unplugged
event PLUGGED
event UNPLUGGED

Figure 7.9: Activity diagram of the docking station sensor skill

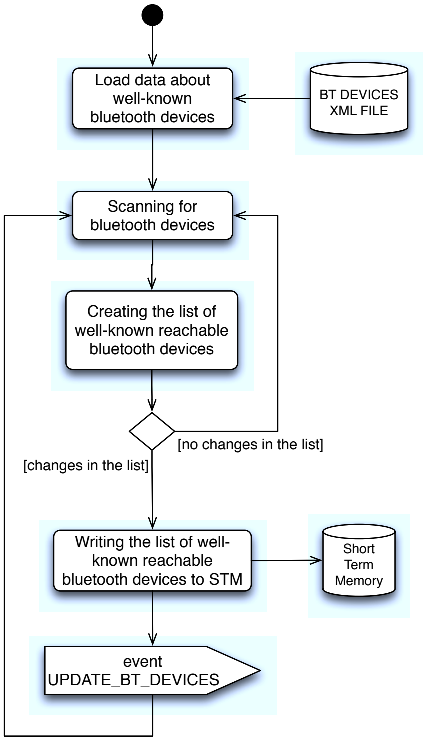

Load data about well-known bluetooth devices
BT DEVICES XML FILE
Scanning for bluetooth devices
Creating the list of well-known reachable bluetooth devices
[no changes in the list]
[changes in the list]
Writing the list of well-known reachable bluetooth devices to STM
Short Term Memory
event UPDATE_BT_DEVICES

Figure 7.10: Activity diagram of the bluetooth discoverer skill

This skill searches for bluetooth devices which are detailed in an XML file in LTM. The link between each bluetooth device and its user is defined in the XML file.

The control loop of the skill scans for near well-known bluetooth devices and writes to STM a string with the name of the users whose bluetooth devices have been detected. At the beginning of each iteration, the skill scans all available bluetooth devices. Then, it just filters the available bluetooth devices which are identified in the XML file. The resulting list of devices is compared with the list in the previous iteration. If differences appear, the new list of users is written to STM and the *UPDATE_BT_DEVICE* event is emitted. Otherwise, the loop starts again. The list of detected users is a string formatted as a space separated list of names. This string is stored in STM with the *ID_BT_DEVICES* id. Every time a user appears/disappears, the new list is written and the event *UPDATE_BT_DEVICES* is triggered. The list of users is updated in STM just when it changes. Figure 7.10 summarizes the whole process.

As mentioned, the information about users and their bluetooth devices are obtained from an XML file. An example is shown in Listing 7.2. The XML file must contain enough information for identifying the user and his bluetooth device. In order to undoubtedly identify a device, its bluetooth address (similar to the MAC address in network cards) or the device name are considered. Both fields are queried by the skill, but just one of them is required. The owner of the device is defined by the user’s name (the attribute *user*) in the XML element.

Code listing 7.2: XML file describing a user’s bluetooth device

```
<?xml version="1.0" encoding="UTF-8" ?>
<devices>
    <device>
        <!-- just the bluetooth address or the user-friendly name
            is enough -->

        <!-- bluetooth address -->
        <address>00:25:67:6F:2F:3D</address>

        <!-- user-friendly name -->
        <name>S8000</name>

        <!-- user’s name -->
        <user>alvaro</user>

    </device>
</devices>
```

In the experiments carried out in this work, two user’s bluetooth devices have been tracked.

7.4.5 The rfid discoverer

In the previous section, cellular phones were used to identify users 10 meters around the robot. This could be enough for some applications. However, when human-robot interaction is achieved in shorter distances (e.g. touching the robot, talking to the robot, etc), another technology is required.

The robot employed in this thesis interacts with users in very short distances: it plays board games, reacts to touch, or establishes dialogs. Then, other additional mechanism is required to distinguish between people really close to the robot from people in its vicinity.

In this work, Radio Frequency Identification (RFID) has been used as the short range identification technology. In this case, the user is given a personal RFID tag which can be placed at his pocket or wallet, and it will be sensed by the robot when he is closer than 1 meter.

In general, this skill can be used for identifying any object which has an RFID tag attached to it (also referred as RFID objects). Each RFID object is identified by a string which has been previously written in its RFID tag. This skill provides a high-level interface since it refers to the presence or absent of objects. However, it does not deal with low level operations. This is achieved by other skills running in the robot. It mediates between low-level operations and the user. The low level operations are performed by other skill which is in charge of reading the data from a new RFID tag, write them to STM, and finally notify it by emitting the *NEW_TAG_RFID* event (more details about low level RFID operations can be read in [196]). The *rfid discoverer* skill informs about the presence/absence of objects with an RFID tag. This makes very easy to extend the repository with more items, even if they were detected by other technology.

The *rfid discoverer* skill informs about the objects detected by the RFID antennas in the robot (one in the head, one in the chest, and one in the base). When a new RFID tag corresponding to an object has been detected, this is notified emitting the *DISCOVERED_RFID_OBJECT* event. An identifier is attached to this event, and it indicates which object has been sensed.

In the same way, once an RFID object has been detected and it is considered as *present*, a timeout is used for checking when it disappears: after certain time without sensing the RFID object again, the event *DISAPPEARED_RFID_OBJECT* is emitted and it is assumed that the object has disappeared (it is *absent*). The parameter attached to this event identifies the disappeared object. The time out has been set to 30 seconds ($timeout = 30$).

The list of RFID objects that the robot is able to detect is obtained from an XML file. The XML element named as *object* stores the data related to an RFID object. The value which is written in the RFID tag, and that is used to identify it, is stored at the child element *id*. Then, the name of the object and its description are the contents of the elements *name* and *description*. Finally, the child element *event_param* contains the number that identifies the object when it is discovered/disappeared (the parameter attached to the *DISCOVERED/DISAPPEARED_RFID_OBJECT* event). An XML example is presented in Listing 7.3

Code listing 7.3: Example XML file describing an RFID object

```
<?xml version="1.0" encoding="UTF-8" ?>
<rfid_objects>
    <object>
        <!-- rfid tag value -->
        <id>alvarocastro</id>

        <!-- name of object -->
        <name>alvaro</name>

        <!-- object description -->
        <description>phd candidate at roboticslab researching on
           social robotics</description>

    <!-- param attached to event emitted when the object is detected -->
    <event_param>1</event_param>
  </object>
</rfid_objects>
```

The main steps of this skill are described in the Figure 7.11. Initially, the data related to all RFID objects are loaded. Then, once an RFID tag is detected (this is notified by the low-level RFID skill emitting the *NEW_TAG_RFID* event), its value is read from STM and compared with the list of recently detected RFID objects. If it is a new one, it is added to this list and its timeout is reseted. If it already was in the list, the timeout is reseted too because this object has been perceived. In every iteration, the timeouts for all recently detected objects are updated. If any timeout reaches zero ($timeout = 0$), it is interpreted as that the object has disappeared, so it is deleted from the list of recently detected RFID objects, and it is notified by the emission of the corresponding event. The next time it appears, it will be considered as a new object and the corresponding event will be emitted.

The combination of the last two skills presented in this chapter, the *bluetooth discoverer* and the *rfid discoverer* skills, provides a reliable system to perfectly identify the presence of users. As a result, three different states are possible for a person:

- absent: the user is not perceived by either the bluetooth device or the RFID tag.
- present: the user is in the vicinity; his bluetooth device is detected, so he is about 10 meters around, but his personal RFID tag cannot be sensed.
- close: the user is closer than 1 meter to the robot; both bluetooth device and RFID tag are perceived.

Graphically, it can be seen as depicted in Figure 7.12 where the different ranges for both technologies are shown.

7.5 How the robot interacts with the objects

The robot's world consists of items. The robot interacts with these items by means of the execution of actions related to them. These actions are implemented as skills running within the robot's control architecture. This section details all the actions related to the items in the robot's world.

7.5.1 Charge the battery

This action is related to the docking station. Its task is to plug the robot into the docking station and to stay there until its battery is totally recharged.

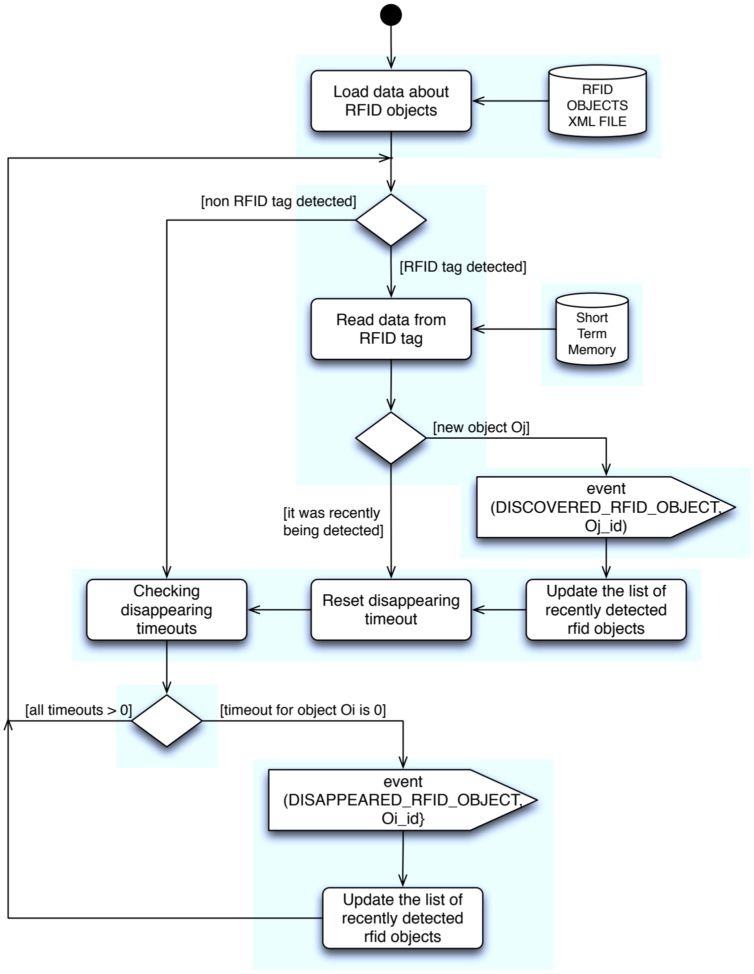

Load data about RFID objects
RFID OBJECTS XML FILE
[non RFID tag detected]
[RFID tag detected]
Read data from RFID tag
Short Term Memory
[new object Oj]
event (DISCOVERED_RFID_OBJECT, Oj_id)
[it was recently being detected]
Checking disappearing timeouts
Reset disappearing timeout
Update the list of recently detected rfid objects
[all timeouts > 0]
[timeout for object Oi is 0]
event (DISAPPEARED_RFID_OBJECT, Oi_id}
Update the list of recently detected rfid objects

Figure 7.11: Activity diagram of RFID discoverer skill

The skill implementing this action is called the *Charge* skill. The process for recharging the battery is shown in Figure 7.13. The first step is to determine if the robot is already plugged. If it is not, it approaches the docking station using the navigation system. The robot knows several well-known positions and the location monitor skill (Section 7.4.1) is in charge of moving it to the front of the docking station. Once the robot is facing the docking station, it has to accurately center its plug to the socket in the charger. This task is achieved by means of the laser telemeter which gives higher resolution than the geometric navigation. Then, Maggie moves back until the plug fits into the socket. This is detected by the *Docking Station Sensor* skill (Section 7.4.3). In the last step the robot remains plugged until its battery is totally recharged. Finally, the successful end of the action is pointed out by emitting the `CHARGED` event. In case an error occurs, an event, which indicates the type of error, is sent.

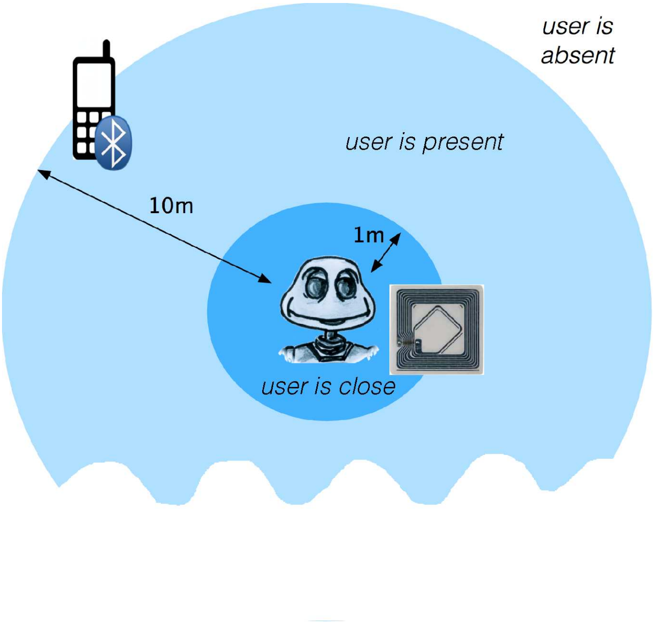


Figure 7.12: Sketch about the ranges of both technologies for identifying a user

Once the *Charge* skill has ended, the battery is full and the robot is still plugged. As seen before, this implies that the *survival* drive is satiated.

7.5.2 Staying plugged

This is the other available action in relation to the docking station. It makes the robot remains in the same situation, without moving, for a certain amount of time. There are two ways of operating:

- when the skill is activated, the robot waits for a specific time (30 seconds). After it, the event `STAYED` is emitted.
- when the event `WAITING` is received, the attached parameter defines the waiting time. After the time is over, the `WAITED` event is sent.

This skill guarantees that, during a fixed amount of time, the robot does not move at all.

7.5.3 Dancing

The Dancing skill requires that the robot is *listening* the music in order to execute its process. This skill makes the robot rhythmically moves its amrs and neck. It seems like the

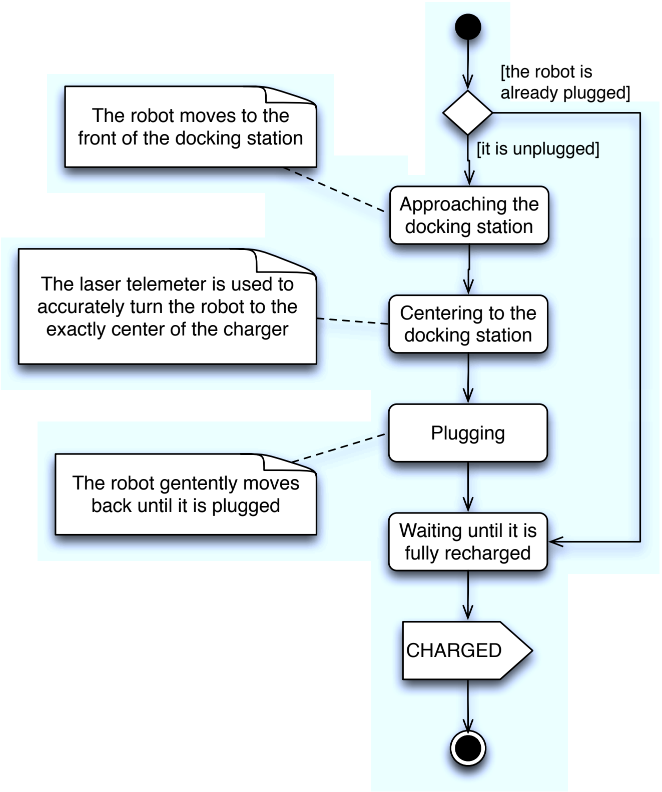


Figure 7.13: Activity diagram for the *Charge* skill

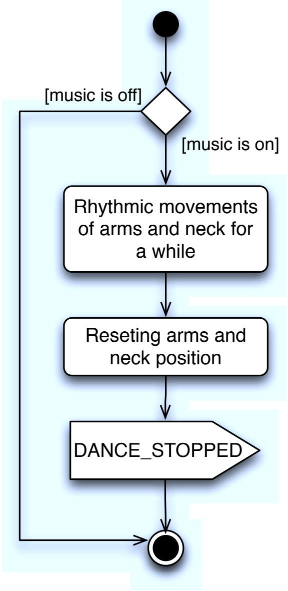

[music is off]
[music is on]
Rhythmic movements of arms and neck for a while
Reseting arms and neck position
DANCE_STOPPED

Figure 7.14: Activity diagram for the *Dancing* skill

robot is dancing. The process is presented in Figure 7.14. At the end of the action, the robot moves down its arms and its head is facing the front.

The end of the action is notified by the `DANCE_STOPPED` event.

7.5.4 Geometric move to

This skill moves the robot in the environment. It is employed by the actions which require a displacement in the environment. In this work these actions are related to the music player and the docking station. The class implementing this skill is the same class used for monitoring the robot's locations (Section 7.4.1). This section corresponds to the functionality mentioned in that section and labeled as *(a)*.

When the event `GEOMETRIC_MOVE_TO` is received by this skill, its parameter indicates the position to move to. The coordinates corresponding to this position are read from the matching entrance in the positions XML file (an example can be seen in Code listing 7.1). Once the coordinates are ready, they are sent to the *Go To Point* skill, which moves the robot to the desired position. Once the robot is there, the skill *Go To Point* notifies it by the event `I_AM_HERE` which is managed by the *Geometric Move To* skill. It checks that the reached goal corresponds to the desired one. If it is so, the `GEOMETRIC_GOAL_REACHED` event is sent and the location identifier is attached to it. This process is summarized in Figure 7.15.

The *Go To Point* skill [196] manages low level operations for moving the robot in the

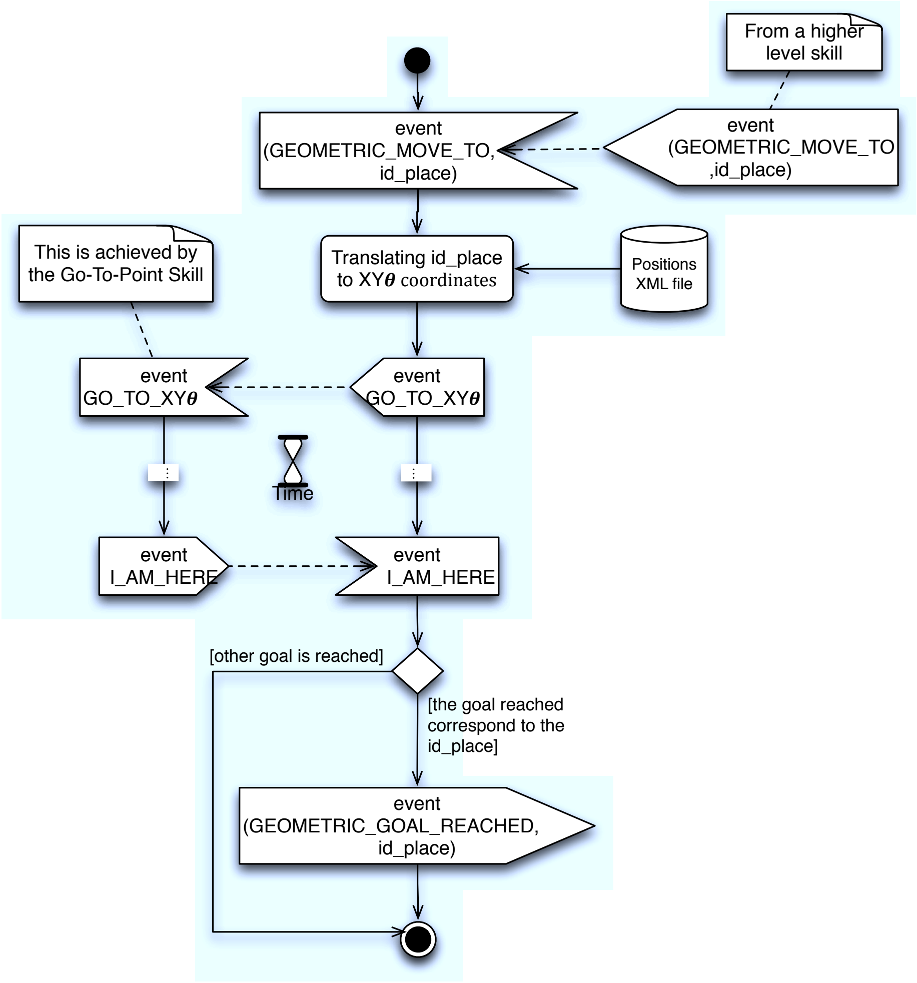

From a higher level skill
event (GEOMETRIC_MOVE_TO ,id_place)
event (GEOMETRIC_MOVE_TO, id_place)
Translating id_place to XYθ coordinates
Positions XML file
This is achieved by the Go-To-Point Skill
event GO_TO_XYθ
event GO_TO_XYθ
Time
event I_AM_HERE
event I_AM_HERE
[other goal is reached]
[the goal reached correspond to the id_place]
event (GEOMETRIC_GOAL_REACHED, id_place)

Figure 7.15: Activity diagram for the *Geometric Move To* skill

environment.

7.5.5 Staying

Once the robot is close to the music player, it can control it or stays there for a while. This last action makes the robot to remain close to the music player. The skill in charge of performing it is the same used for staying plugged (Section 7.5.2).

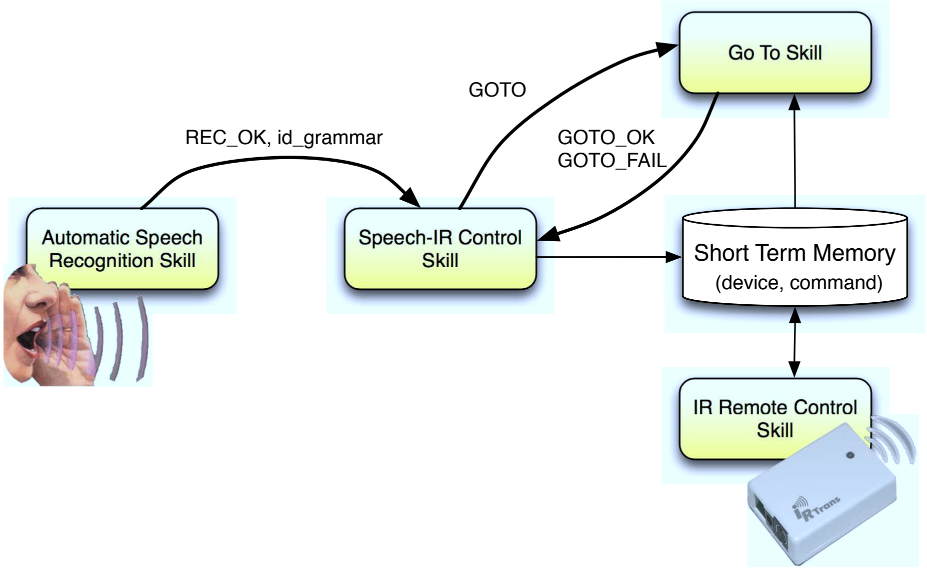


Figure 7.16: Communications among all the skills involve in the music player control

7.5.6 The music player control: turning it on/off

The robot Maggie is endowed with a voice operated infrared remote control. In general, the robot is able to control any home appliance equipped with an infrared interface. This is fulfilled by most of the regular appliances. Therefore, this means that no changes or adjustments have to be done in the appliances. In this particular skill, due to the limited number of home appliances in the lab, the skill is limited to operate a television. This television is employed, during the experiments, as a music player.

The required infrared commands for operating the music player (the television) are previously recorded from the original remote control. Subsequently, the commands can be sent by the robot upon request of a user. The human-robot interaction is achieved by means of the dialog system.

The music player is located in a certain position, so, when the robot has to control it, first, it has to approach it and face it.

In order to send commands to the infrared-operated appliances, Maggie is equipped with an infrared emitter/receiver. It has been placed inside Maggie's belly, behind a sphere which lets infrared signal to go through. Because of the nature of the infrared technology, it is essential that the robot is located close enough and facing the music player. Hence, a reliable navigation system is a fundamental element.

In order to successfully achieve the task, this skill is sustained by other skills running in the robot. The communication among all these skills is depicted on Figure 7.16. Following, the role of each skill is explained.

1. ***ASR Skill***: the Automatic Speech Recognition skill is in charge of informing about which grammar rule has been identified through the microphones. An event (*REC_-OK*) and the detected grammar rule identifier are sent to alert the rest of the architecture. This event will be catched by every skill subscribed to it; in particular, the skill named *Speech_IR_Control*.

2. ***Speech_IR_Control Skill***: it is a data processing skill which translates an incoming event from the ASR skill to a new one. The new event is based on the identified grammar rule which identifies the requested command. If the command is not related to the infrared system, the event is ignored. In other case, the required information is stored at STM. This information is the device to control and the command to send; for example, "turn the music player on". Then, the *Speech_IR_Control* skill indicates that Maggie has to move to the device's location by means of the *GOTO* event. If the position is reached, then the *GOTO_OK* event is received. Then, the robot is ready for emitting the appropriate command. Consequently, this is notified by sending the *CONTROL_IR* event. In case of any error, the operation is aborted.

3. ***GoTo Skill***: after the *GoTo* skill receives the *GOTO* event, the robot is intended to move to the position determined by the data stored in STM. Particularly, this skill takes the name of the device to be operated from the STM and it relates it to a pose (position and orientation) in an internal map of the world. If the desired position is reached, the *GOTO_OK* event will be sent. Other case, *GOTO_FAIL* is sent.

4. ***IR_Remote_Control Skill***: the *CONTROL_IR* event is captured by this skill. Once it is received, it accesses the data concerning to the corresponding command at STM. Then, the info is sent to the infrared server. The right ir coding to sent is obtained from the database where all the available coding commands are. Now, the infrared hardware emits this coding. Finally, it informs that everything has gone right.

The chronological evolution of the music player control skill is shown in the sequence diagram in Figure 7.17. When a user wants Maggie to operate an infrared appliance, he interacts with the robot by voice commands (message 1 Figure 7.17). The ASR skill identifies it and distributes the grammar rule joined to the user's command by means of the event *REC_OK* (message 2 Figure 7.17). Once the *speech_ir_control* skill receives it, it links the recognized grammar rule to a device and an instruction. Both parameters are stored and shared by means of STM. At this moment, Maggie has to change its position in order to face the appliance (message 3 Figure 7.17) which name is stored in STM. The name is linked to a pose in the internal map and the robot goes there. Whether it achieves it or not, it is notified by the *GOTO_OK* or *GOTO_FAIL* events (message 4 Figure 7.17). If Maggie is ready (i.e. facing the appliance) and the required data are available, the *CONTROL_IR* event is sent and it is received by the *ir_remote_control* skill. It asks the infrared server,

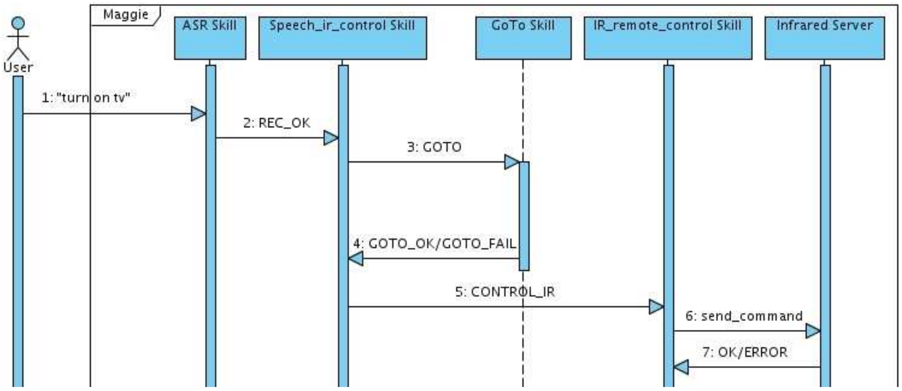


Figure 7.17: Sequence diagram of the music player control skill

which is directly connected to the hardware, for sending the requested command. The result of the operation is back-propagated.

7.5.7 Interacting

In this action, the robot does not move at all but asks a person for interacting with it. Then, the presence of a person is required for this action. The robot evaluates the action executed by the person in terms of the internal robot's wellbeing; i.e. how the person's action affects itself.

This action is implemented by the *Interact* skill. This skill detects the oral and tactile effects of the person's action. Roughly speaking, this skill distinguishes if the person says compliments to the robot or he offends it. Moreover, it evaluates the tactile interaction as a stroke, a damage, or neutral. Both interaction mechanisms run in parallel when the skill is activated. In order to easily understand how it works, each one is independently analyzed.

Insulting or paying it a compliment

The verbal actions of a user can be interpreted as positive or negative according to the meaning. In order to classify them, the *Interact* skill evaluates them.

The verbal communication is managed by a dialog system which is based on the Automatic Speech Recognition (ASR) and the Text To Speech (TTS) systems. The dialogs are formatted following the Voice XML standard (VXML) which defines the structure of the Dialogs. VXML converts speech to text by means of grammars. Grammars are a set of rules which define the sentences or words the robot is able to understand.

In this case, the required grammar is defined considering the possible insults or compliments the user says to the robot. Then, the rules of this grammar, somehow, describe the robot. Grammars allow the addition of semantic meanings. These semantic meanings are expressed in the following way:

```
<@attribute = value>
```

where *attribute* is a variable which will be set to *value*.

For example, the next grammatical rule represents the affirmative or negative answers:

```
public $yes\_no=("yes":yes|"no":no|"okey":yes|"affirmative":yes|"
   negative":no) {<@option $value>}
```

Then, the variable `@option` is set to the semantic value of the rule named `yes_no`. The possible values are `yes` or `no`. This variable is used inside the dialog too. Strings in quotation marks represent the possible words recognized by the ASR, that is, the possible words pronounced by the users.

The *Interact* skill uses a specific grammar for recognizing insults or compliments. This grammar is shown in Listing 7.4. The main rule (`root`) refers to the rule named `describing` (line 6). This rule accepts any sentence (this is represented by `GARBAGE`) before an insult or compliment (line 13). The repertory of understandable insults/compliments is defined by the rule `insults_compliments` (lines 8-11). This rule fixes the attribute `adjective` to the value `INSULT` or `COMPLIMENT`.

Code listing 7.4: The grammar for compliments and insults

```
1  #ABNF 1.0 ISO-8859-1;
2
3  language es-ES;
4  tag-format <loq-semantics/1.0>;
5
6  public $root = $describing;
7
8  $insults_compliments =
9     ("idiot":INSULT | "stupid":INSULT | "silly":INSULT | "clumsy"
        :INSULT | "ugly":INSULT | "bored":INSULT | "disgusting"
        :INSULT | "bastard":INSULT | "wicked":INSULT | "bitch"
        :INSULT | "incompetent":INSULT | "filthy":INSULT |
10     "clever":COMPLIMENT | "pretty":COMPLIMENT | "cute":COMPLIMENT
          | "you smell good":COMPLIMENT | "fun":COMPLIMENT | "funny
         ":COMPLIMENT | "fun-loving":COMPLIMENT | "charming"
         :COMPLIMENT | "scream":COMPLIMENT | "lovely":COMPLIMENT |
         "graceful":COMPLIMENT
11    ){<@adjective $value>};
12
```

```
$describing = [$GARBAGE] $insults_compliments;
```

VXML dialogs are based on forms which are filled according to the data provided by a user and processed by a grammar. In the mentioned application, just one form is necessary. This dialog is shown in Listing 7.5. Initially, the form is identified with the name `insults_compliments` (line 7) and several properties are defined: the `timeout` property specifies the default interval of silence allowed while waiting for a user input before a *noinput* event is thrown. In this case, it is set to 30 seconds (line 9). Then, the default language and grammar are set (lines 12 and 13 of Listing 7.5). The `prompt` tag controls the output of the dialog: it can be a synthesized sentence, the configuration of a property, emitting an event, etc.

The `field` element specifies an input item to be gathered from the user. In this dialog, this `field` is linked with the `adjective` attribute defined in the `insults_compliments` grammar. Once `adjective` is filled by a user's utterance, the action defined by the code in the `filled` element is executed (between lines 19 and 28). In this case, depending on the value of the `adjective` attribute, the user has paid Maggie a compliment or he has offended it. In the first case, the *COMPLIMENTED* event is sent and a happy sentence is said (lines 21-22). In the last case, the *OFFENDED* event is emitted and a sad sentence is said (lines 24-25).

In the case that the field is not filled (the user does not speak, or his speech does not fit the grammar), after 30 seconds, the `noinput` element is executed (lines 30-33): this is notified by the *IGNORED* event and the robot says *"you ignore me"*.

Code listing 7.5: The VXML dialog used by the *Interact* action.

```
1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <vxml xmlns="http://www.w3.org/2001/vxml"
3        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4        xsi:schemaLocation="http://www.w3.org/2001/vxml http://www.
            w3.org/TR/voicexml20/vxml.xsd"
5        version="2.0"
6       >
7          <form id="insults_compliments">
8  
9                  <property name="timeout" value="30s"/>
10 
11             <block>
12                 <prompt>#setLanguage$en</prompt>
13                 <prompt>#setGrammar$insults_compliments.gram</
                      prompt>
14             </block>
15 
```

```
            <!-- An insult or compliment is detected -->
            <field name = "adjective">

                <filled>
                    <if cond = "adjective == ’COMPLIMENT’">
                        <prompt>#emit$COMPLIMENTED</prompt>
                        <prompt>en:uuuii great! I really like it</prompt>
                    <elseif cond = "adjective == ’INSULT’" />
                        <prompt>#emit$OFFENDED</prompt>
                        <prompt>en:you are not very polite</prompt>
                    </if>
                                <clear/>
                </filled>

                <noinput>
                                <prompt>en: you just ignore me</prompt>
                                <prompt>#emit$IGNORED</prompt>
                        </noinput>

            </field>
        </form>
</vxml>
```

Several examples of possible Dialogs are presented on Figure 7.18. Figure 7.18(a) shows how a user offends Maggie. The next figure, Figure 7.18(b), displays the messages between the robot and the user and how he says a compliment to Maggie. In the last example, Figure 7.18(c), the users ignores Maggie and he does not say a word.

Stroking or beating the robot

In addition, apart from verbal interaction, the *Interact* skill evaluates the tactile communication. In order to achieve it, the sensitive “skin” of the robot is used. The capacitor sensors spread around the surface of the robot are read to determine where and how the robot is being touched.

In this skill, two kinds of tactile interactions are distinguished: strokes and hits. The first one is identified when the user strokes the robot’s head. In contrast, when the robot’s both shoulders are touched, it is considered as a hit. In short, this process is depicted on Figure 7.19. Every time a touch is detected on the surface of the robot, this is communicated to the rest of the architecture by the *Tactile Sensor* skill which emits the *TOUCHED* event. When

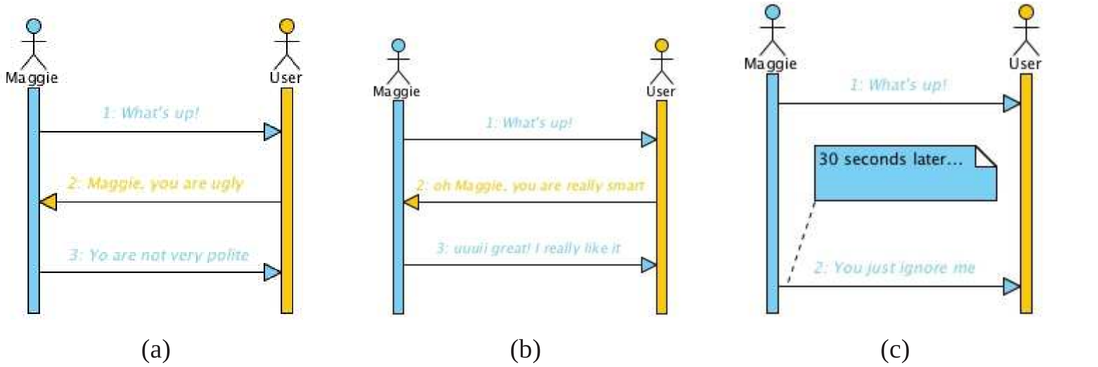


Figure 7.18: Three possible dialogues with a user

the *Interact* skill is running, this event is received and the tactile information is processed. The touch is classified as stroke, hit, or other. Strokes are considered similar to compliments and hits are similar to insults, thus the corresponding events are sent: *COMPLIMENTED* and *OFFENDED*.

The *Tactile Sensor* skill manages all the tactile sensors in the robot's skin and informs about what sensors have been touched.

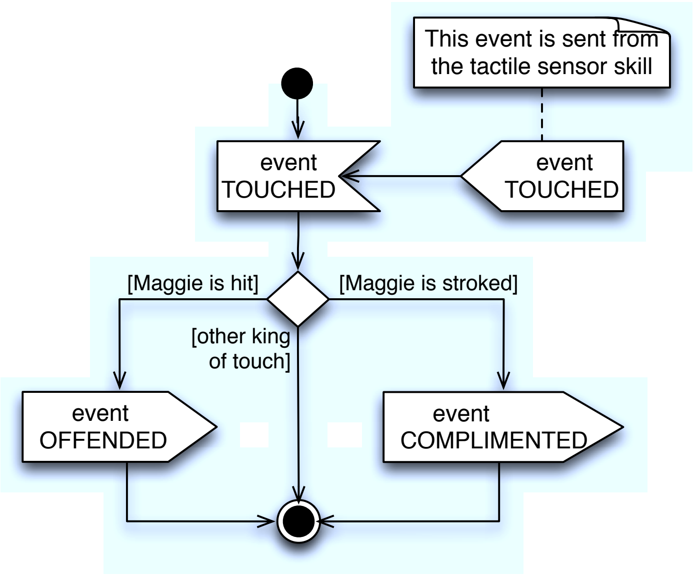


Figure 7.19: Activity diagram for tactile interaction

7.6 Summary

This chapter has shown how the whole system is technically designed and implemented. The relationship among different elements have been detailed.

Initially, the design of the DMS has been presented and its advantages has been justified. The system is fully configurable through the data stored in a database. The database design and the software design for managing all the data have been detailed.

Later, the skills running in the robot, either for detecting the external state, or for executing actions, have been analyzed and explained. The technical details have been shown in an attempt to clarify the ins and outs of the whole system.

The DMS pretends to be a fully customizable system. It has to be flexible enough to be effortless transferred to other robots. The proposed system perfectly can run in other platforms by just updating the data on the database. However, skills are more platform dependent. Some of them should be modified before they run in other platforms; for example the geometric navigation is dependent of the physical parameters of the mobile base. Other skills can perfectly work out-of-the-box, e.g. the *RFID discoverer* skill. But others absolutely cannot be adapted due to physical constrains of the robot; the *Dance* skill cannot be adapted to a robot without head and arms.

CHAPTER 8

Testing the experimental setup

8.1 Introduction

This chapter presents the scenario and conditions for the experiments. First, a general common setup of the experiments is given. Following, a fragment of an experiment is deeply analyzed in terms of motivations. This shows the interdependences between all elements and the operating of the DMS. Later, the use of the Object Q-Learning algorithm is justified and its benefits are exposed. Finally, the modifications to the learning algorithm are validated.

8.2 The arrangements for the experiments

In this thesis, the experiments consist of two phases: exploring and exploiting. First, the robot learns the proper behaviors in different situations, so it has to explore all possibilities. Maggie tries every action in order to learn the right policy to act. Second, the learned policy is exploited selecting the best action according to the world configuration in each moment or state. The behaviors are originated as a consequence of the string of these actions. During exploitation, learning is frozen and the best action is always selected.

All the experiments have been achieved by real robot-environment interactions, so the behaviors and actions imply to interact with the items in the robot's environment. An overview of the robot's environment was displayed in Figure 5.5 (Chapter 5). Learning has been also achieved by real robot-environment interaction, which means that the robot explores all available actions in every world configuration many times. As explained in

Section 4.4.1, each action will be evaluated according to its effect over the robot's wellbeing.

Reinforcement learning algorithms have been used during the experiments (Chapter 6). In particular, the Object Q-Learning algorithm (Section 6.2) has been implemented. Previous knowledge has not been given to the robot in advance, so it has learned from the ground up.

In the experiments, an iteration corresponds to the execution of an action by the robot. The robot decides at each iteration the action to carry out. The probability of an action to be selected is determined by the Q value associated to this action in the current state, and by the level of exploration. During the exploring sessions, all actions have the same probability of execution. This is required in order to guarantee that all actions are tried many times from all the possible states. At exploitation, the best action is always selected. The best action is the most convenient in terms of the robot's wellbeing. That is, action a is the best action when the robot is in state s if the $Q(s, a)$ value is the highest one among the Q values corresponding to the rest of the available actions.

The interactions between the robot and the environment where experiments are accomplished take a considerable amount of time. Hence, for most of the experiments, the learning phase has been established to last around 700 iterations which usually means a duration of more than seven hours.

As exposed in Section 5.4.3, the balance between exploration and exploitation will depend on the *temperature* factor. During learning, *delta* δ is set to 100 causing a high *temperature*, so the actions will be randomly selected to try all possible actions. Furthermore, initially, the learning rate α is set to 0.3 which means that the most recent data are quite relevant during learning. As justified in [49], at some point, exploring must stop and the learned values must be exploited. Considering this approach, after 500 iterations, the learning rate starts to continuously decrease until the learning rate reaches 0. After this point, the Q values will not change anymore and the experiments enter in the exploitation phase.

Since this work has been implemented on a social robot intended to interact with people, the *person* object has been considered as the only active object which shares the environment with Maggie and interacts with it. Consequently, the exogenous actions are those actions executed by the people around Maggie. Recalling, the exogenous actions affect the external state as well as the internal state of the robot. For example, when a person approaches Maggie, the state related to this person (the external state) has changed, and it is not due to the robot's actions. Moreover, the actions accomplished by a person may affect some robot's drives (the internal state): e.g. if a person hits the robot, the *social* drive soars, i.e. the need of a positive social interaction increases. Again, all these consequences are not caused by the robot but by the people's action (the active objects' action).

In these experiments, two people will interact with the robot: *Alvaro* and *Perico*. Both alternatively approach Maggie, one by one. *Perico* always interacts with positive actions:

he strokes the robot or he says compliments to Maggie. This results on the satisfaction of the social drive, which is set to 0. *Alvaro* generally acts in a positive way too. However, sometimes, he hits or offends Maggie. This is reflected in the robot's wellbeing through an increment of ten units in the *social* drive (Equation (8.1)). In general, both users benefit the robot but *Alvaro* occasionally causes harm to it.

$$\text{If the robot is harmed} \Rightarrow D_{social} = D_{social} + 10 \tag{8.1}$$

8.3 Analysis of the course of the motivations

First of all, in this section, the course of the motivations is detailed. That is, how motivation values change with time during Maggie's "lifetime". A ten-minutes period has been extracted and fully analyzed. This meticulous study proves the correct working of the system as well as clarifies how the internal and external stimuli are combined to compute the intensity of the motivations.

Motivations uniformly grow but, sometimes, their intensities suddenly change. These jumps occur due to the presence of external stimuli as well as the effects of the robot's actions on the drives. Figure 8.1 shows the evolution of the motivations during ten-minutes period and several of these jumps can be observed. The execution of different actions are identified by the letters between brackets located on the top of the figure. In order to clearly identify the iterations, i.e. the execution time for each action, the background of the plot is grey and white-striped. The multicolored band on the upper part of the figure represents the dominant motivation at each moment. Its colors match the colors of the motivations shown in the key of the graph.

Initially, the *charge* action (*c*) greatly reduces the *energy* drive. At the same time, as justified in Section 5.4.1, other drives are slowed down. Later, when the robot executes the *go to the player* action (*g*) and, consequently, it unplugs from the charger, the external stimulus of the *survival* motivation disappears and this motivation is reduced.

The influences of other external stimuli can be observed too. For instance, when the music player is turned on (action *play (p)*), the *fun* motivations increases; when the player is switched off (action *stop (s)*), the same motivation decreases; additionally, the presence of a person is reflected on the *social* motivation.

Satisfaction of several drives can be observed due to the execution of the correspondent consummatory actions. For example, the *fun*, *relax,* and *social* motivations jump down when their drives are satiated by means of the *dance (d)*, *stop (s)*, and *interact with Perico (iP)* actions respectively.

Focusing on the middle part of the graph, the *relax* motivation suddenly soars: at the same time the robot interacts with *Perico (iP)* and the need of socialize is satiated, the *relax*

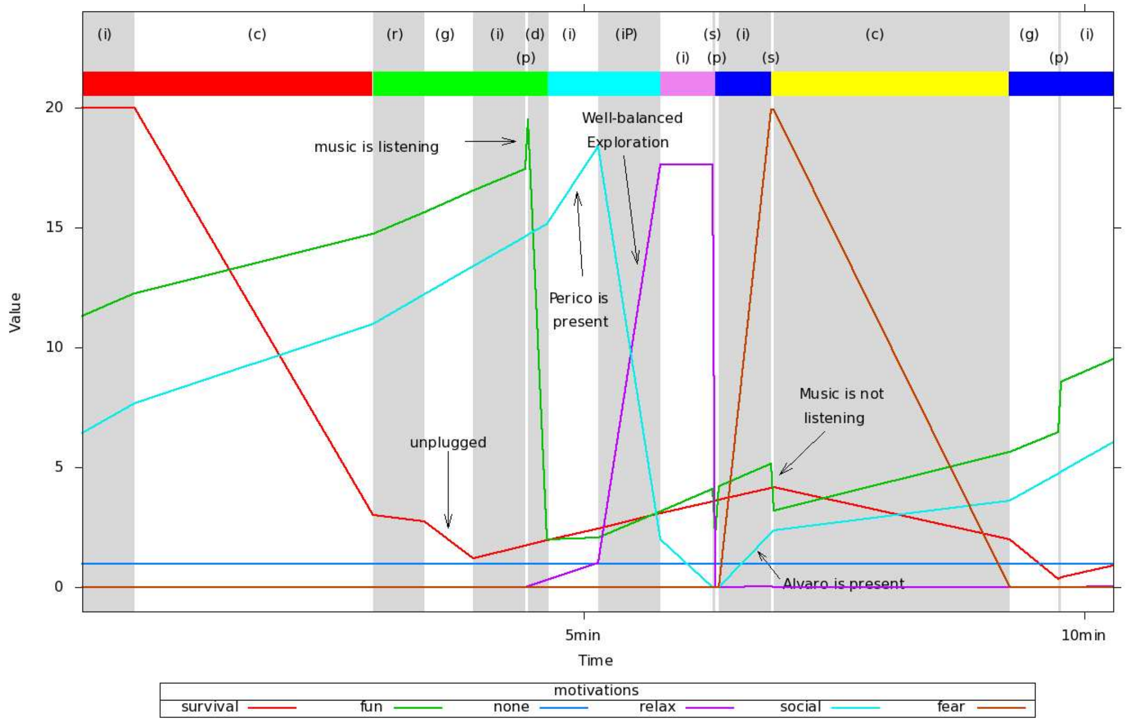


Figure 8.1: Temporal evolution of motivations. Numbers on top represent the executed actions: (i)idle, (c)charge, (r)remain plugged, (g)go to music player, (p)play, (d)dance, (iP)interact with Perico, and (s)stop. The vertical white-grey bands at the background correspond to the execution time of each action. The upper colored band indicates the dominant motivation. The effects of some actions and several changes of states are pointed.

motivation is saturated. Why is that? This is an example of how the Well-balance Exploration mechanism (Section 6.3.1) is applied. Recalling, this is an artificial modification of the robot's drives for a comparable level of exploration of all motivations. This iteration is not considered in the learning process because the variation of the robot's wellbeing is not "naturally" produced.

Looking at the right part of the graph, *Alvaro* approaches Maggie. This fact originates the emergence of the *fear* motivation. This motivation dissipates when the robot moves to other location, in this case to the charger (action *charge (c)*) and *Alvaro* is not present any more. Its causes and consequences are deeply examined in Section 9.2.

In the plot, the *none* motivation has a constant value of 1. Remembering Section 5.4.1, this motivation will be considered as the dominant one when no other competes. This occurs twice in the left part (blue parts in the multicolor band). This is the period of time when all the drives related to the other motivations are below their activation levels.

8.4 Testing the learning algorithm

The utilized learning algorithm, Object Q-Learning, was initially proposed due to a necessity of reducing the state space and, consequently, the learning time. Thus, firstly, this algorithm is compared with a traditional Q-Learning showing its advantages.

Later, when it is applied to a real robot, some improvements are required (Section 6.3). Then, the benefits of the modifications of the learning algorithm are shown by means of some experiments.

8.4.1 Object Q-Learning vs. Q-Learning

At this point, the use of Object Q-Learning is justified. Since the world is perceived in terms of objects and the robot's states in relation to these objects (Section 6.2.1), an agent using the traditional Q-Learning will learn the actions that satisfy the robot's needs in relation to just one object. However, it does not learn the related actions affecting other objects that are necessary. In other words, Q-Learning allows to learn when to execute the consummatory actions, but not the appetitive actions related to different objects.

Since objects have been considered as independents one from each other (Section 6.2.1), traditional Q-Learning will update the state-action value related to a particular object computing the previous value, the obtained reward, and the best value from the new state with that object. Using this approach, the effects of an action executed with an object but affecting other objects too are not considered. As presented in Section 6.2.3, this situation is not close to real life because any action can influence several objects. For example, when you feel tired, you go to bed. This action affects your need of rest, your location, and the bed (before it was free and now it is taken). Thinking of the experimental setup, if Maggie needs to get its battery recharged, it will move towards the charger, get plugged, and remain until its battery is high enough. This action alters, not just the state related to the docking station (from unplugged to plugged), but the states related to the music player (from close to far), and the people around the robot (from present to absent or vice versa).

However, by means of the Object Q-Learning and the collateral effects, the consequences of an action over all objects in the world are considered. As explained in Section 6.2.3, Object Q-Learning updates the Q-values based on previous Q-values; the reward after the action; the best value from the new state using the object the action has been executed with; and, finally, for the rest of the objects, the difference between the best action from the new state and the best action from the precedent state. Thus, the results of an action over all objects are considered in the Object Q-Learning algorithm.

The different results obtained by Object Q-Learning and Q-Learning can be seen in Figure 8.2. Both plots present the results obtained after learning the behavior when the dominant motivation is *fun*. That is, what the robot has to do to satisfy the need of entertainment. Figure 8.2(a) shows results obtained using Q-Learning. In Figure 8.2(b), the

Q values plotted have been learned by means of the Object Q-Learning algorithm. Both algorithms were run in parallel during the same execution of the robot.

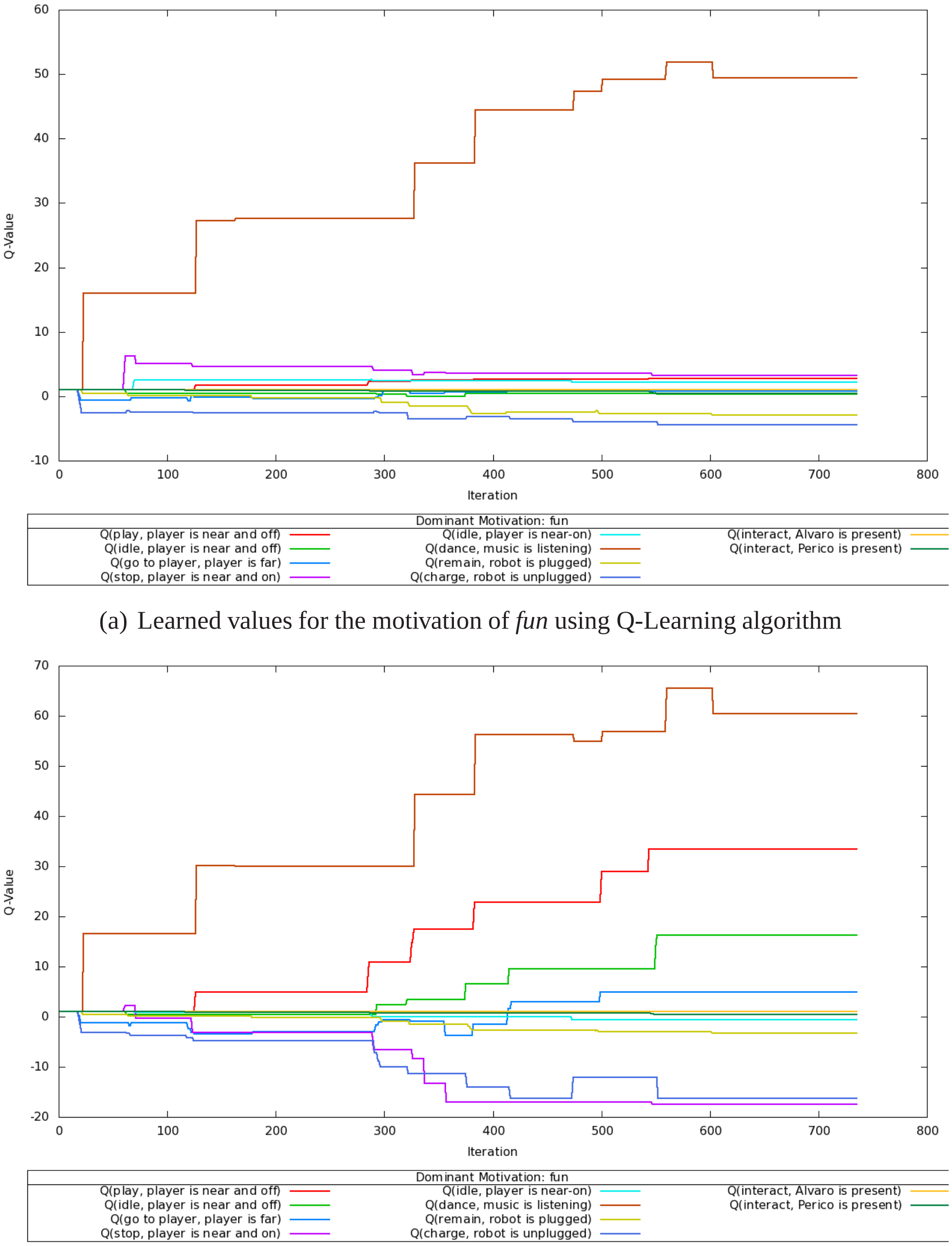
As expected, both methods learn that the best action to execute is *dance* (the consummatory action) because it satisfies the need of *fun*. However, in order to achieve this action, other objects are required: first, if the robot decides to dance, the music has to be on; and for turning the music on, the robot has to be close enough to the *music player*. This relationships among several objects and the states in relation to these objects cannot be learned by Q-Learning. Figure 8.2(a) shows how the rest of the actions have very low values. In fact, the next best action after *dance* is *stop music*. This is an incongruity since playing music is mandatory for dancing but, because of *stop music* satisfies the need of relax, when *fun* is the dominant motivation, there can be a little need of relax and then a low positive reward is assigned to *stop music* action.

On the other hand, the robot using the Object Q-Learning algorithm perfectly learns the correct relation among objects in order to expose the proper behavior when *fun* is the dominant motivation. In Figure 8.2(b) the most appropriate sequence of actions (consummatory and appetitive) can be extracted considering the highest values. As previously said, *dance* is the most valuable action and it corresponds with the highest value. Before this action can be executed, the *play music* action is required (it is the second highest value). Finally, the last required action is *go to player*, which is in charge of moving the robot close enough to the *music player*. Once there, the robot is able to *play music* and, then, to *dance*. The *go to player* action is the forth value and the last positive one.

There is one positive action left: the *idle with music off* action, which has a high value too. When this action is carried out, the robot is close to the *music player* and it is off. In this situation, the next best action is to *play music*, which has a very high value. For this reason, the value of the *idle with music off* action in this situation is high too. Actually, this is the third highest value.

The rest of actions are not relevant for the behavior exhibited when *entertainment* is the dominant motivation. However, some brief comments about the least valuable actions will help to clarify some ideas. Through Object Q-Learning just two really bad actions has been identified in relation to this motivation: *charge* and *stop music*. The *charge* action moves the robot far from the *cd player* so it cannot *play music* and, as result, it will not *dance*. In relation to the *stop music* action, which was deeply analyzed for regular Q-Learning, the reader can observe how this action has become the worst one (i.e. the lowest value). Looking into the Figure 8.2(b), it can be seen how *stop music* starts to abruptly decrease between iteration 300 and 400, which corresponds when the *play music* and the *dance* actions have relative high values. This makes sense because, as explained before, music is required to dance and, if music is suspended, it must be penalized.

Therefore, it has been proved that Object Q-Learning better performs in relation to the collateral effects. However, when there is just an object involved in a behavior, both algorithms are able to learn the proper skills to be activated. Figure 8.3 displays the Q



(a) Learned values for the motivation of *fun* using Q-Learning algorithm

(b) Learned values for the motivation of *fun* using Object Q-Learning algorithm

Figure 8.2: Comparison between traditional Q-Learning and Object Q-Learning when several objects are required for performing the behavior related to the motivation of *fun*

values learned when *relax* is the dominant motivation. Figure 8.3(a) represents the Q values determined by Q-Learning. In contrast, Figure 8.3(b) represents the results obtained by the Object Q-Learning algorithm. Now, in both cases, the learned values result in the proper behavior, which is formed by actions performed with the same object. The most important actions in order to *relax*, sorted by value, are: *stop music*, *idle with music on*, and *go to player*. All of them are related to the *music player* item and, therefore, both algorithms perfectly identify them.

The worst actions are analyzed as well. The least valuable action is *charge* for the two algorithms. Nevertheless, Object Q-Learning penalizes it in a greater manner because it considers the effects of this action over other items. *Charge* moves the robot to the docking station and plugs the robot for recharging its battery. Therefore, the robot is moved away from the music player. This fact is reflected by Object Q-Learning assigning a lower Q value to this action. In the case of the Q-Learning, it just considers that this action does not benefit the *relax* motivation, but it does not include the detriment.

Independently of the learning algorithm, from Figure 8.3, it is easy to describe the optimum behavior that the robot will exhibit when *relax* is the dominant motivation: if it is far from the music player, it will go towards it; then, it will stop music.

In conclusion, the robot using Q-Learning learns the direct action to deal with each motivation, i.e. the consummatory action, and the preceding actions (appetitive), all of them linked to the same object. However, this is not enough to behave in a proper way. Object Q-Learning provides a mechanism to acquire the required knowledge in order to exhibit behaviors that satisfy motivations involving several independent objects and their states. Then, the proper action with each object at each particular state will be carried out. Therefore, the robot learns consummatory as well as appetitive actions. This is the policy which will be exploited.

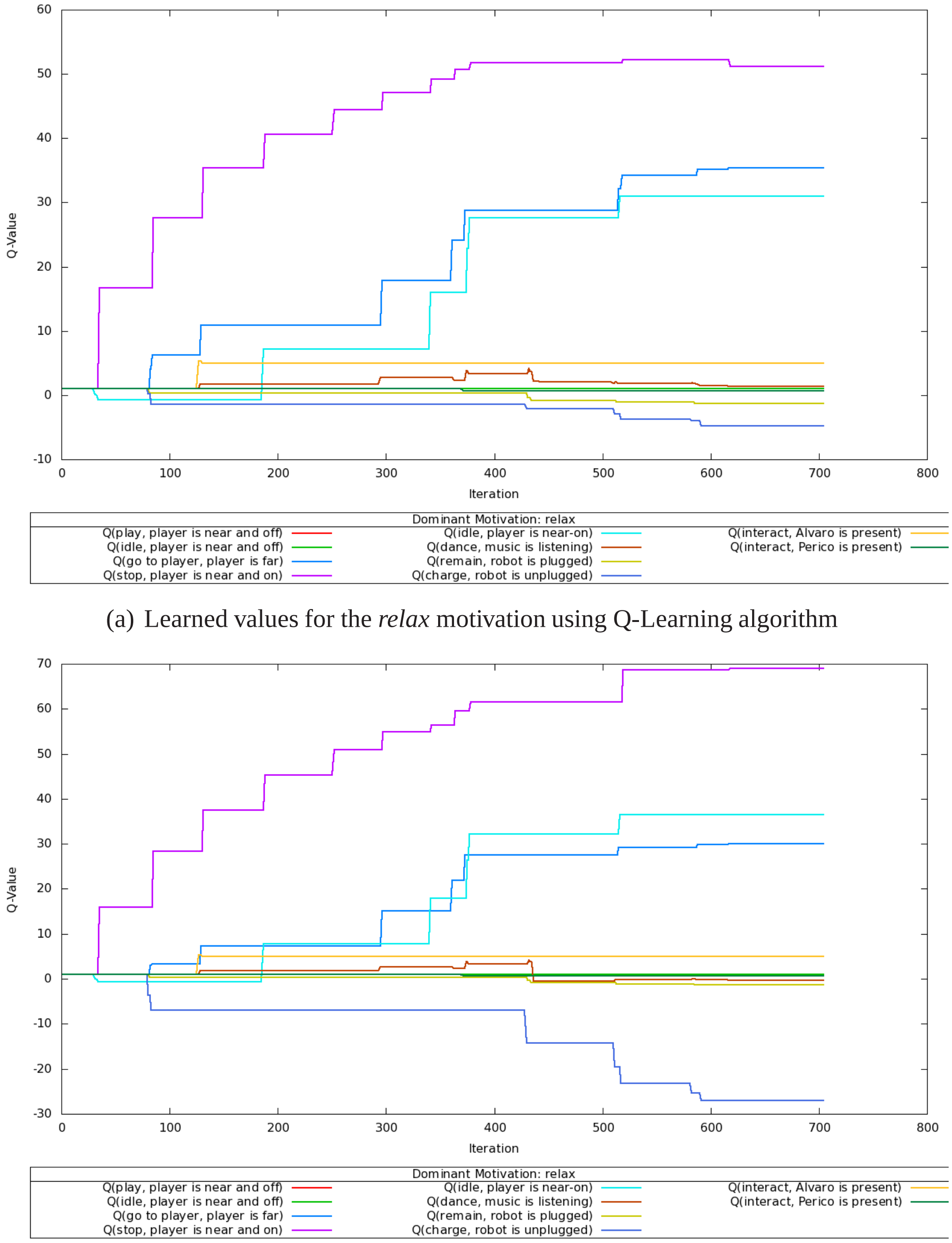
8.4.2 Validation of the improvements for learning behaviors

The benefits obtained by the mechanisms in charge of boosting learning process (Section 6.3) are exposed here. Both, the Amplified Reward and the Well-balanced Exploration, are analyzed comparing the results obtained with and without them in similar experiments.

Amplified Reward

In order to clearly demonstrate the advantages of using the Amplified Reward, this experiment has been focused in one dominant motivation: the *fun* motivation. In this case, a seven hundred iterations learning session has been performed. Two versions of the learning algorithm are concurrently running: a) an Object Q-Learning algorithm with Amplified Reward (Figure 8.4(a)), b) an Object Q-Learning without Amplified Reward (Figure 8.4(b)).

Looking into Figure 8.4, at first glance, both plots seem similar: despite the fact that



(a) Learned values for the *relax* motivation using Q-Learning algorithm

(b) Learned values for the *relax* motivation using Object Q-Learning algorithm

Figure 8.3: Comparison between traditional Q-Learning and Object Q-Learning when just one object is involved for performing the behavior related to the motivation of *relax*

the amplified one (Figure 8.4(a)) has higher values, the policy seems to be equal. However, focusing on the *going to the player* action, this is not equal. This action is required in order to satisfy the need of entertainment. In Figure 8.4(a), the Q value associated to this action is the forth highest positive value. In contrast, in Figure 8.4(b), this Q value is negative and other actions not related to the motivation of *fun* are over its value.

Using the Amplified Reward the learned values are higher, therefore, the back-propagation along all successive needed actions is stronger and it reaches farther actions faster.

Probably, longer experiments will end with a positive value of the *go to the player* action. However, by means of Amplified Reward this is achieved in a shorter period of time.

Well-balanced exploration

As expressed in Section 6.3.1, an exhausted exploration of all situations in order to correctly learn the proper behaviors is needed. Next, a situation where exploration is poorly achieved is shown. Figure 8.5 presents a four hundred iterations learning session where the Well-balanced Exploration has not been considered. It corresponds to the dominant motivation *relax* which associated drive is the slowest one (this has been explained in Section 5.4.1).

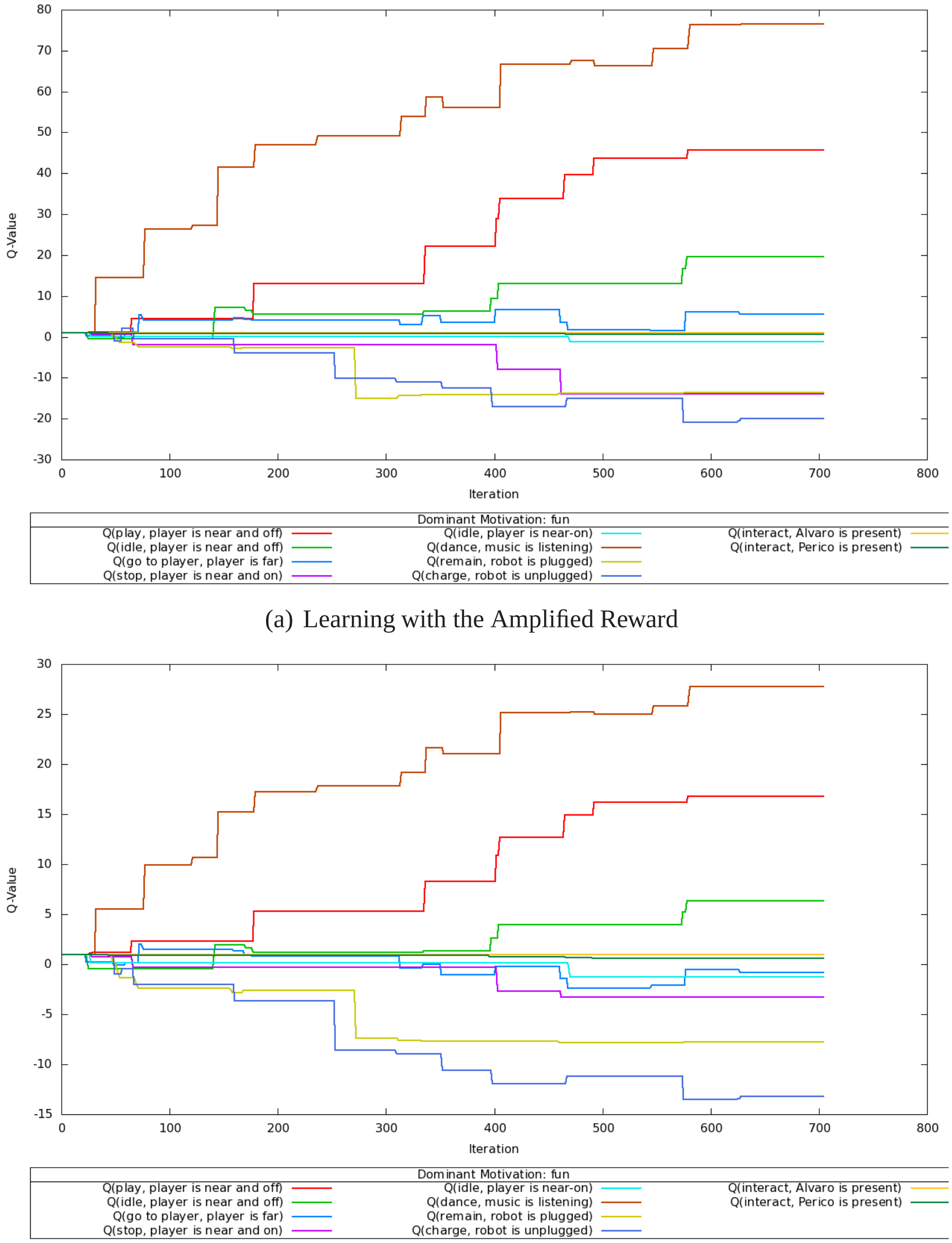
The remarkable issue extracted from Figure 8.5 is the long periods where non of the values are updated. Roughly, these periods correspond to the iterations ranges from 0 to 160 and from 250 to 390; this is about one hour and a half. These long lasting periods with stability of values during a learning session means that this motivation is not explored in these periods. In other words, *relax* does not frequently become the dominant motivation. These circumstances lead to a set of state-action pairs that are not enough explored and therefore they will not be properly learned in an acceptable amount of time.

The effects of the Well-balanced Exploration when *relax* is the dominant motivation can be observed in Figure 8.3(b). During the whole learning session, there is a frequent update of any state-action pair related to the *relax* motivation. There are not more of those long periods of undesired stability in a particular motivation.

8.5 Summary

At the beginning, this chapter introduces the structure of the experiments with two different phases: the exploring phase where learning is achieved, and the exploiting phase where the learned policy is employed. Moreover, the available active objects were introduced: the users; two people will share the robot's environment during the experiments: *Perico* (who always positively interacts) and *Alvaro* (he sporadically harms the robot).

This chapter has proved the correct working of the DMS. Initially, how the intensities of motivations are formed due to the interconnections with internal and external stimuli has been clarified and examined in a fragment of a real experiment.



(a) Learning with the Amplified Reward

(b) Learning without the Amplified Reward

Figure 8.4: Effects of Amplified Reward on the learning process when the dominant motivation is *fun*

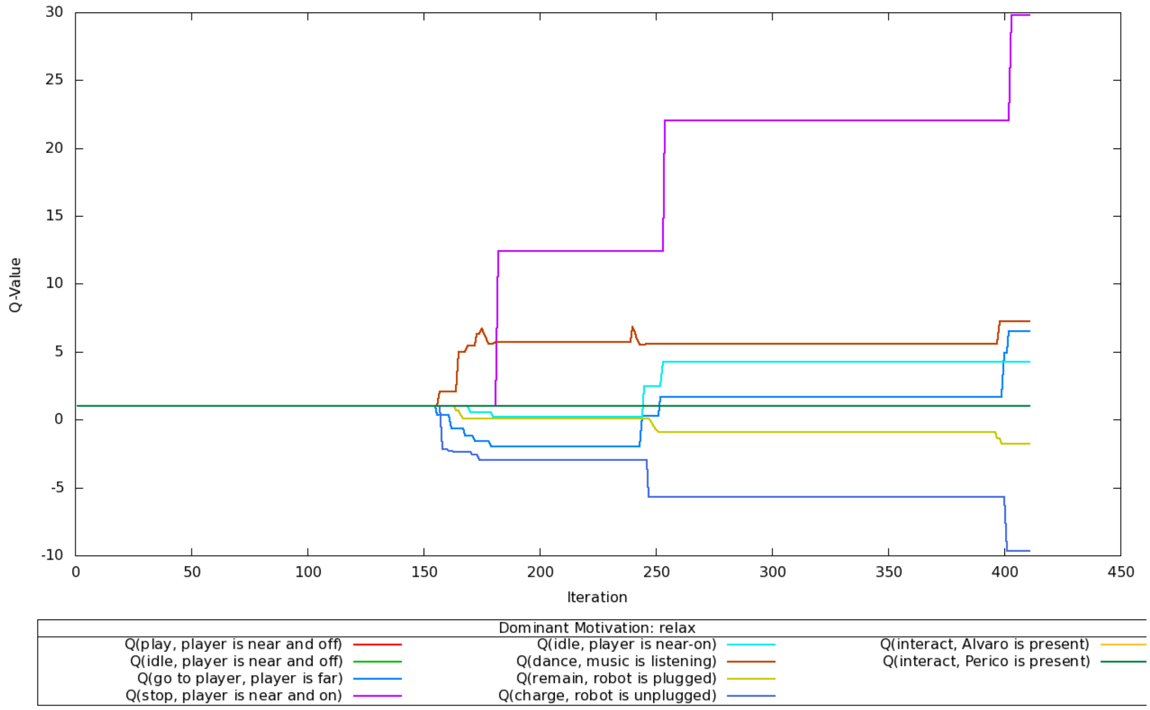


Figure 8.5: Learned Q values when dominant motivation is *relax* and Well-balanced Exploration is not included

Later, the benefits of Object Q-Learning were demonstrated since it considers the effects of the actions on all objects. Moreover, the improvements in the learning process were analyzed and their advantages shown: the Amplified Reward speeds up the learning of behaviors (especially those formed by appetitive actions which are related to several objects) and the Well-balanced Exploration enforces learning on states hardly tried. These two methods shorten the learning process.

The configuration for the experiments presented at the beginning of this chapter and the justified learning algorithm will be employed in the experiments exposed in Chapter 9.

CHAPTER 9

Experimental Results

9.1 Introduction

Once the DMS, its elements, the robot, and the experiment setup have been described, it is about time to put the robot to learn in the lab. In this chapter, the results obtained from several experiments prove the performance of the presented system.

At the beginning, considering that the application of fear is one of the most relevant contributions, the results of including the emotion of *fear* are firstly detailed.

Then, since the *happiness* and *sadness* emotions have been used as reinforcement during learning, the resulting policy is studied. The learned behaviors for each motivation are analyzed.

9.2 Fear results

This section validates and analyzes the use of fear in the social robot Maggie. More specifically, how *fear* improves the decision making process, and by extension the robot's autonomy, is exposed.

As previously said, fear has been considered as a motivation which incites the robot to behave. The experiment consists of comparing the performance of the robot with and without fear as a motivation in the same environment and conditions. Therefore, two different learning or exploring sessions have been performed: one including fear as a motivation, and other where fear does not exist. With the resulting policies, two different exploiting sessions are performed and the results are compared.

In this first section, the results of the appraisal of fear are analyzed. That is, the identification of new dangerous situations. Later, the adaptability of the proposed method is demonstrated by comparing different learned reactions to fear depending on the user's behavior. Finally, the usefulness of fear and its advantages are proved.

9.2.1 Results on the appraisal of fear

During the experiments, considering that the maximum "punishment" of a negative exogenous action corresponds to a penalty of ten points to the *social* drive (Equation (8.1)), and based on observations during trials, L_{danger} (the minimum of the Q_{worst} values of the actions in a state in order to consider it as a safe state, Section 4.4.2) has been set to -10 points. As a consequence, whenever the robot is in a state where there is a $Q_{worst}^{obj_i}$ value below this threshold, this is considered as a *dangerous state*. Therefore, the *fear* motivation suffers a drastically increment as shown in Equation (9.1).

$$\begin{array}{l}\text{If } s \text{ is a dangerous state} \Rightarrow \text{Fear} = 19.9\\ \text{If } s \text{ is a safe state} \Rightarrow \text{Fear} = 0\end{array} \tag{9.1}$$

where s is the state of the robot. This equation was already presented in Section 4.4.2 (Equation (4.13)).

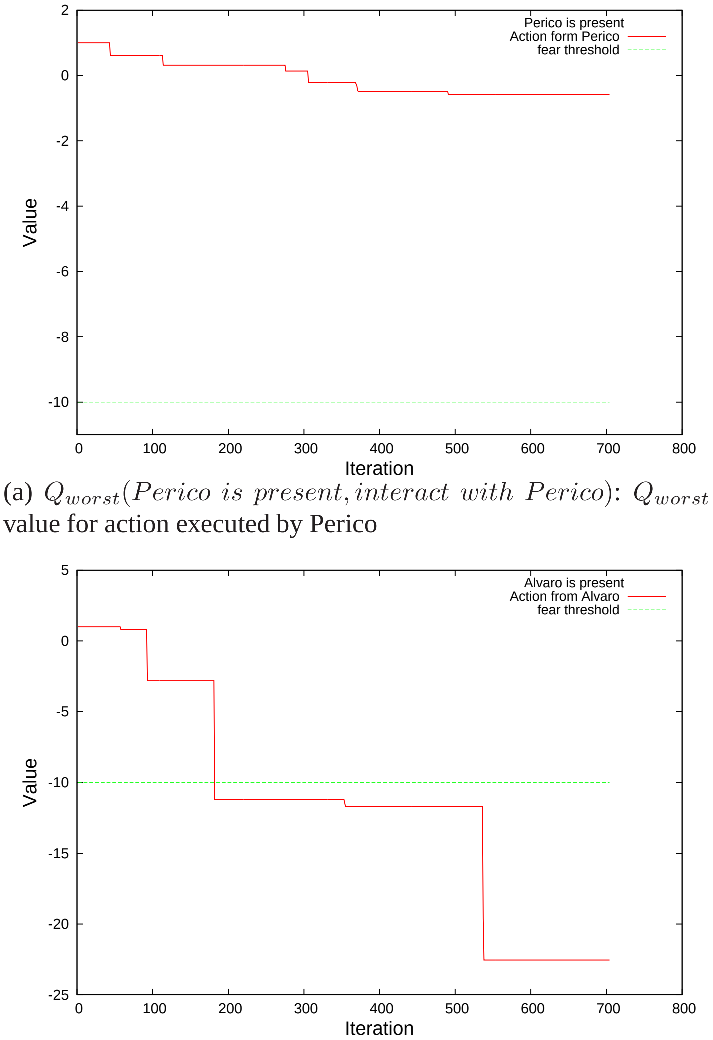
As already said, the consequences of the actions executed by both users (*Alvaro* and *Perico*) over the robot's wellbeing are perceived by Maggie. In order to do it, Maggie is endowed with the *Interact* action. This action does not have effects over the Maggie's drives or its external state; therefore, it is possible to evaluate how the exogenous actions affect the robot's wellbeing. Thus, translating Equation (4.12) into the experiment, it results on Equation (9.2).

$$\begin{array}{l}\text{If } Q_{worst}^{Alvaro}(s, interact) < -10 \Rightarrow \text{s is a dangerous state}; \forall s \in S_{Alvaro}\\ \text{If } Q_{worst}^{Perico}(s, interact) < -10 \Rightarrow \text{s is a dangerous state}; \forall s \in S_{Perico}\end{array} \tag{9.2}$$

Since there are two different users, there are two different instances of the same action which depend on who is interacting with the robot: *interact with Alvaro* and *interact with Perico*.

Note that the exogenous actions have been executed when a person is *present*. Therefore, considering Equation (9.2), the worst Q values are associated to the state s when $s = Alvaro\, is\, present$ or $s = Perico\, is\, present$. Naturally, if a person is absent, his actions do not interfere on the robot's "life". Therefore, potentially dangerous states are $Alvaro\, is\, present$ and $Perico\, is\, present$ because Maggie can be damaged from them.

Figure 9.1 depicts the evolution of the worst Q values associated to the exogenous actions. As can be seen in Figure 9.1(a), since all *Perico*-Maggie interactions are favorable from a robot's point of view, its Q_{worst}^{Perico} value slightly decreases from its initial value 1, and it remains stable around value 0. In contrast, the Q_{worst}^{Alvaro} value associated to the *Alvaro*'s interactions is significantly reduced (Figure 9.1(b)). This is due to the number of interactions where *Alvaro* has hit or offended Maggie. This number is low in comparison to the total amount of interactions: during the learning phase, *Alvaro* harmed Maggie five times of thirty-seven interactions ($13'5\%$).



(a) $Q_{worst}(Perico\ is\ present, interact\ with\ Perico)$: Q_{worst} value for action executed by Perico

(b) $Q_{worst}(Alvaro\ is\ present, interact\ with\ Alvaro)$: Q_{worst} value for action executed by Alvaro

Figure 9.1: Q_{worst} values of exogenous actions.

Looking into Figure 9.1, the robot does not know anything about dangerous states, or what to be afraid of, until iteration 182. At this point, *Alvaro* hits the robot one more time, and $Q^{alvaro}_{worst}(present, interact)$ reaches the value -11.2097. This value is under the selected threshold ($L_{danger} = -10$) and, therefore, the robot determines that being next to

Alvaro can be harmful. From this iteration on, if *Alvaro* is close to the robot, this is identified like a dangerous state and, as a result, the *fear* motivation is rocketed. Consequently, *fear* potentially becomes the dominant motivation, so it guides the robot's behavior. Therefore, the presence of *Alvaro* is the releaser of the *fear* emotion in this experiments.

9.2.2 Learned fear reactions: escaping

As previously shown, the proposed system is able to identify new dangerous situations which has not been previously defined. Moreover, by means of the learning mechanism of the DMS, the robot determines what behavior must be selected to avoid these situations.

The users (*Alvaro* and *Perico*) approach Maggie, one by one, and stay there. At that point, since Maggie is accompanied, it must decide to interact or to execute another action.

In this experiment, dangerous states are associate to the presence of *Alvaro* because of the few negative interactions (details about how the appraisal of fear is performed can be seen in Section 9.2.1). Then, the robot learns how to "escape" from *Alvaro*.

The actions which imply a displacement on the geometrical position of the robot are *go to player* and *charge*. The former moves to robot towards the *cd player* and the last gets the robot plugged to the docking station. Both actions make *Alvaro* disappears from the robot's scope or the robot moves away from *Alvaro*. Therefore, these two actions are the most appropriated actions when *fear* is the dominant motivation (Figure 9.2). When the robot is scared (i.e. *Alvaro* is beside Maggie), it will move to the *docking station* if it is close to the *cd player*, or to the *cd player* if it is plugged. This is a run-away behavior learned by the robot itself and it is similar to what animals do when they are afraid.

Just as a brief explanation, there are some actions which are always positive in all behaviors (for all dominant motivations). For example, in Figure 9.2, the *dance* action has a positive Q value in all circumstances. This is because this action satisfies the *boredom* drive which is one of the fastest ones. This means that whenever this action is executed, *boredom* is at its ideal value nearly never. Therefore, this action usually has a positive reward. The same explanation applies to the *interact* actions and *loneliness* drive. Therefore, their related Q values are positive for all behaviors. In general, consummatory actions satisfying fastest drives will be always positive in all behaviors.

9.2.3 Learned fear reactions: freezing

Since humans are unpredictable autonomous agents, different reactions to fear can be observed depending on the person involved in the situation.

In the results presented in Figure 9.2, both users alternatively approach Maggie with the intention of achieving some human-robot interaction. Recalling, *Perico* always achieves positive human-robot interactions, and *Alvaro*, once in a while, causes harm to Maggie. As

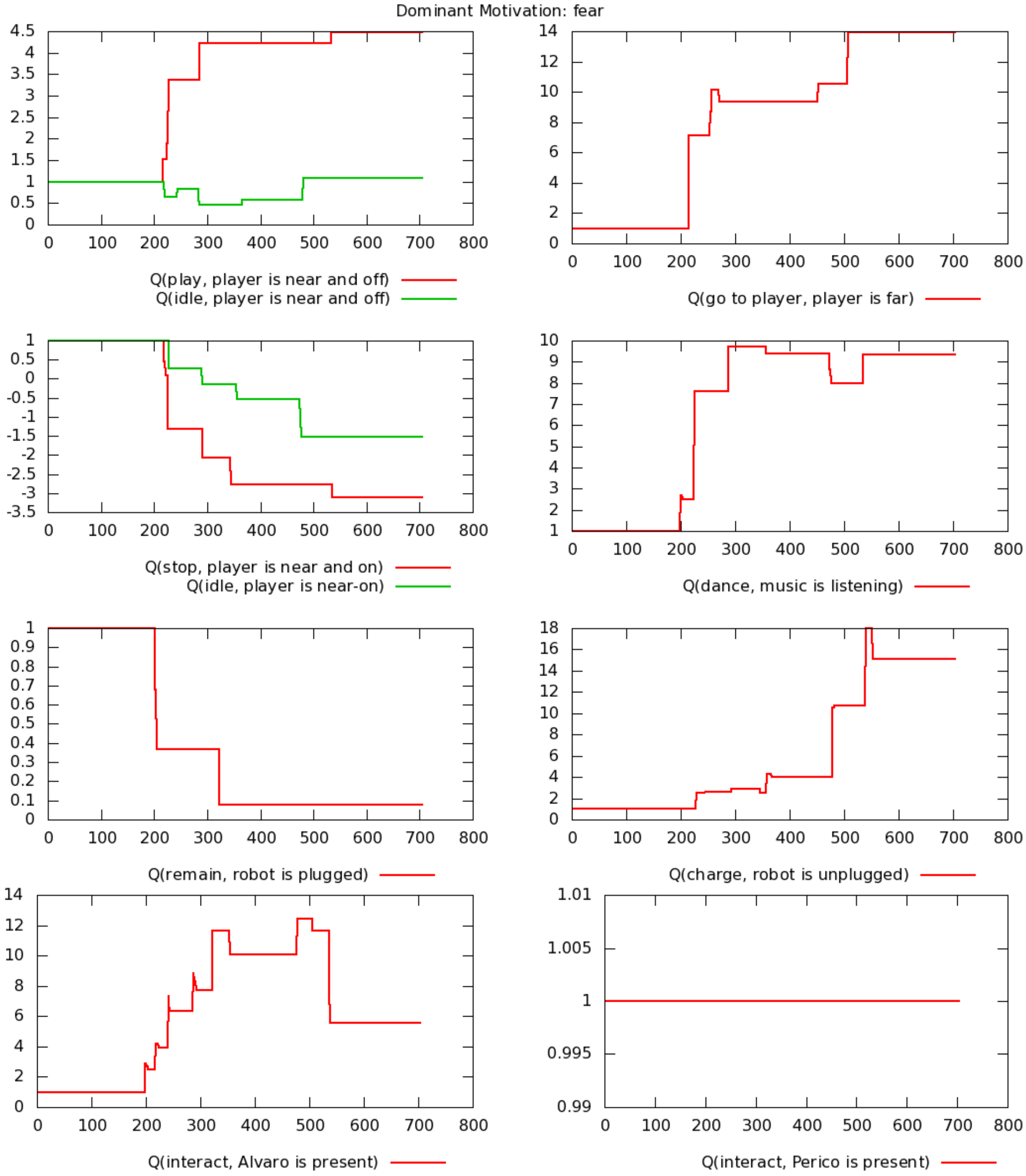


Figure 9.2: Learned Q-values when **fear** is the dominant motivation.

a consequence, Maggie is afraid of *Alvaro* and, as exposed in the previous section, it learns to escape from him.

However, the system is flexible enough to learn different behaviors according to diverse people's attitude. In this experiment, users have been trained to behave in a different way: now, *Alvaro* and *Perico* separately approach Maggie and they chase the robot. Users will leave when they get bored due to the robot's inactivity. Again, *Alvaro* occasionally damages Maggie. Considering these damages, *fear* comes out on Maggie when *Alvaro* is present.

A new learning session has been conducted, similar to the previous ones but with the new behaviors. The results can be observed in Figure 9.3. In this case, the behavior learned when *fear* is the dominant motivation is related to the *idle* action, when Maggie is close to the music player (both, with music on and off), and to the *remain* action, when it is plugged. This is because the Q values associated to these actions are the highest ones (upper three plots on the left column of Figure 9.3). These actions share that they cannot be externally perceived because they do not make any expression or movement, they give the impression of inactivity. Therefore, the robot bores *Alvaro* and he moves away from Maggie. After this happens, *fear* ceases resulting on the following benefit for the robot.

Summarizing, in this experiment the cause of fear (the releaser) has not been changed (the presence of *Alvaro*) and it has been perfectly identified again. However, the reaction to fear is totally different. As proved, the presented method nicely works with users conducting in diverse manners and the proper fear reaction is learned in each situation.

The new learned behavior dealing with fear can be biologically justified considering that some animals paralyze when facing a dangerous situation. It seems that they are "frozen" by fear.

9.2.4 Does Maggie need *fear*?

This section tries to justify the use of *fear* as a motivation. Here, the performance of the robot is measured and compared with the results obtained from experiments where fear does not exist. In this section, the same motivations considered in previous experiments are employed (all motivations introduced in Section 5.4.1).

Two different learning sessions have been realized, both using reinforcement learning algorithms. First, the robot learns to behave without considering *fear* as a motivation. In consequence, the motivations present on this session are: *survival*, *fun*, *relax*, and *social*. In the second session, the same four motivations are considered plus the *fear* motivation. Results from both learning sessions are compared.

During both sessions, the robot learns the right policy to satisfy its needs. However, the session considering *fear* learns an additional behavior in relation with this motivation. Each learned policy is used during an exploiting session. These exploiting sessions last around 80 minutes each one and the best action is always selected at each iteration.

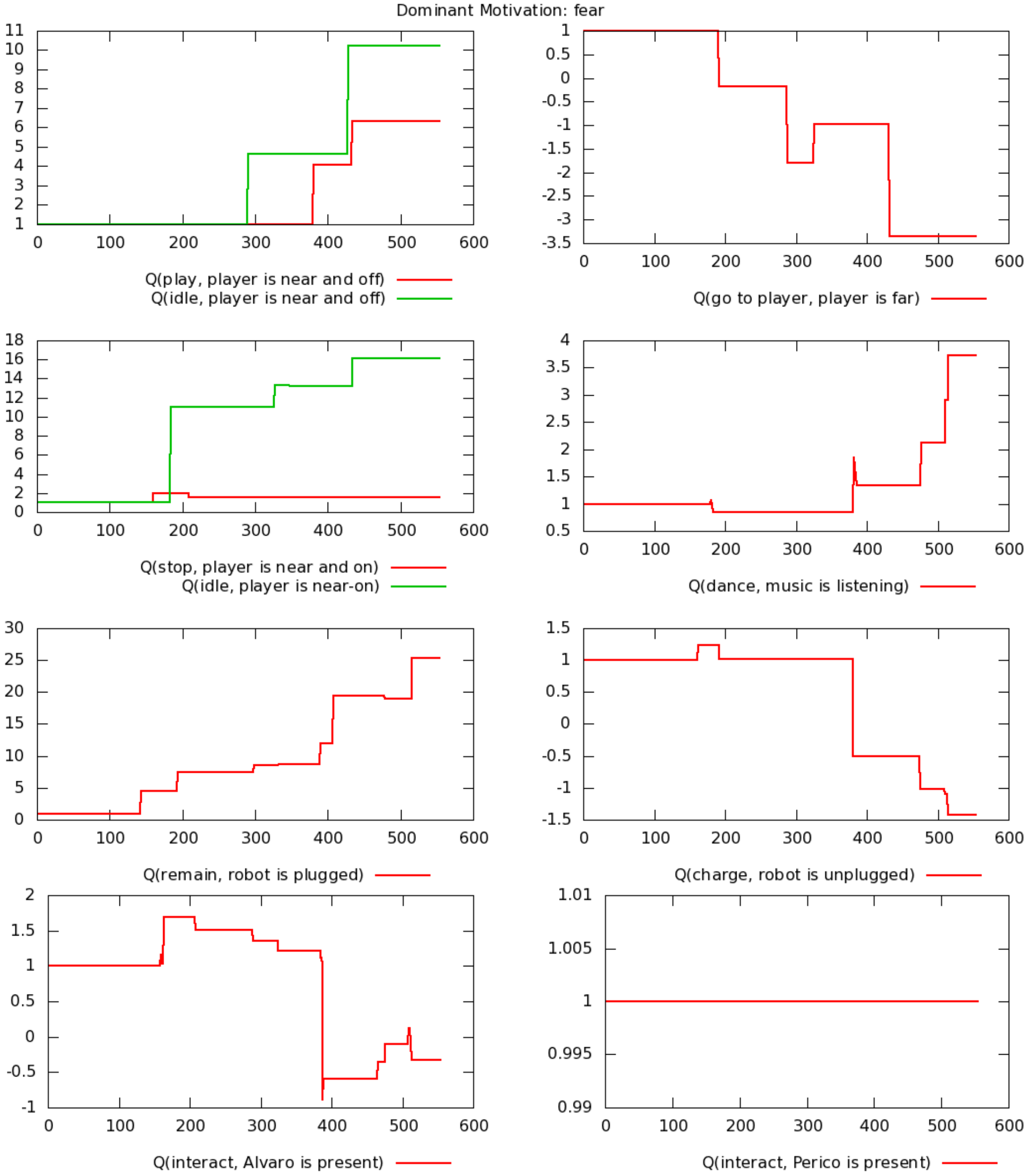


Figure 9.3: Learned Q values when **fear** is the dominant motivation. Alvaro chases the robot until getting bored or interacting with Maggie.

The learning and the exploiting sessions are performed in the same environment considering the two well-known users: *Alvaro* and *Perico*. In this case, the users individually approach Maggie and remain there until an interaction occurs and they move away, or Maggie moves away.

In order to evaluate both configurations, the results obtained during both exploiting sessions are compared. The next performance indicators have been employed: the average wellbeing and the percentage of permanence in a certain security zone. Besides, the results about the percentage of time the robot is without a dominant motivation (all drives are below the activation level L_d introduced in Section 4.2) are presented. This value gives an idea about how "comfortable" the robot is. Finally, the number of times the robot is harmed is also compared.

Average wellbeing

Since the variation of the wellbeing was used as the reward during the learning phase, the robot tends to maximize it. Table 9.1 presents the values corresponding to the average wellbeing with and without *fear* during the exploiting sessions. The average wellbeing when *fear* does not exist is slightly higher. This can be seen as a disadvantage of using *fear*. However, this is understandable considering that, when *fear* is included as motivation, the number of drives used to compute the wellbeing is bigger, so the wellbeing value is lower (the robot's wellbeing is computed as a function of the drives: $Wb = Wb_{ideal} - \sum_i \alpha_i \cdot D_i$, Equation (4.7)).

This drawback can be observed in nature too: a fearful person is not in a pleasant situation, his wellbeing decreases due to the anxiety suffered because of the fear. As a consequence, the person is distressed while he is afraid. However, other benefits can be obtained from *fear*.

Table 9.1: Average wellbeing during the exploiting sessions

without fear	with fear
87.77	86.72

Permanence in the secure area

These benefits are related to other reliable performance rate: the percentage of time the robot's wellbeing remains in a security zone. If the robot's wellbeing is within this area, it can be said that the robot is "fine" because its wellbeing is high. Thus, the percentage of time the wellbeing remains in this area gives an idea about how well the robot is performing.

In order to establish the limits of the secure area, the ideal wellbeing value ($Wb_{ideal} = 100$) and the activation levels for motivations ($L_d = 10$) are considered. Since all drives simultaneously evolve and several motivations can compete for the dominance, the security area width is set to 15. Consequently, it is considered that when the robot's wellbeing is between 100 and 85, it is within the secure area.

Table 9.2 shows the percentage of permanence within the secure area during the exploiting phase. As can be seen, when *fear* is included as a motivation, the wellbeing is almost the 70% of iterations within the secure area, which represents a 5% more than when *fear* is not used. This is coherent because *fear* is used to avoid dangerous states where the robot can be damaged. Once the robot is harmed, the wellbeing decreases enough to move out the secure area.

Table 9.2: Permanence at secure area during the exploiting sessions

without fear	with fear
65%	69.5%

Non dominant motivation

Moreover, if there is not a dominant motivation, it means that all the internal needs and external stimuli are not strong enough to induce a behavior. Hence, it can be considered that the robot is in a comfortable situation. The percentage of time during the exploiting sessions that a dominant motivation does not exist proves how pleasant the robot's "life" is. Table 9.3 shows that considering *fear*, the 78% of the time there is not dominant motivation. On the other hand, when the robot lives without *fear*, the percentage is reduced to 72%. Once again, these numbers show how *fear* provides a better quality of "life".

Table 9.3: Percentage without a dominant motivation during the exploiting sessions

without fear	with fear
72.22%	78%

Number of times the robot has been damaged

The differences of the previous percentage values could seem not very significant. However, it must be recalled that the number of negative interactions (the robot is hit or offended) is very low. During all experiments this only occurs for a low percentage of all interactions with *Alvaro*. Therefore, the impact of *fear* in this scenario can not represent a great improve in the average values. Nevertheless, the impact on the number of times the robot is damaged is outstanding.

Consequently, the most relevant result of using *fear* is related to the damage caused by *Alvaro* to the robot when it "lives" according to the learned policy of behavior. When *fear* is not implemented, the robot tries to interact with both users in order to satisfy its social need. This action leads Maggie to, some times, be harmed by *Alvaro* because it has not learned to identify that being next to *Alvaro* is dangerous. Consequently, it has not learned an avoidance behavior. As depicted in Table 9.4, this happens six times of twenty-three interactions between Maggie and *Alvaro*. Since damages heavily affect the *social* drive, these greatly affect the wellbeing results. For this reason, although the average wellbeing is better without fear, the rest of the performance indicators when *fear* does not exist are disturbed when Maggie is damaged and, as result, their values are worse.

Now, considering *fear* as a motivation in the system, once the presence of *Alvaro* is identified as dangerous, the robot does not interact with *Alvaro* at all so he could not hurt it. This is because, as shown in previous sections, the robot learned to avoid the interaction with *Alvaro*. Focusing again in Table 9.4, by means of *fear*, the dangerous situations are totally averted. In fact, the robot has not been damaged any more when *fear* is implemented. Therefore, *fear* improves the performance of the robot since it provides a safety mechanism to avoid situations where the robot can be damaged.

Table 9.4: Harm/interactions with Alvaro during the exploiting sessions

without fear	with fear
6/23	0/0

In conclusion, despite of the fact that the average wellbeing is hardly worse, *fear* provides significant benefits. Specially the fact that harm is totally avoided.

9.3 Learning behaviors

As presented in Section 4.4.1, *Happiness* and *sadness* are artificial emotions coming up from the variation of the robot's wellbeing. They are used as the reward function during the learning of the policy of behavior. Therefore, the robot's behavior in all circumstances is oriented towards increasing its wellbeing.

The robot Maggie has been learning in sessions which last more than seven hours in the laboratory. In this section, the learned behaviors are analyzed. During the learning, the robot has learned how to act according to its state (internal and external). As explained in Section 6.2.1, the internal state corresponds to the dominant motivation, and the external is related to different objects. Through learning, stable chains of actions have been formed and they can be considered as patterns of behavior corresponding to the motivations. In this section, the learned behaviors are independently presented motivation by motivation.

The behaviors exhibited when *fear* is the dominant motivation have been already shown in Sections 9.2.2 and 9.2.3. Therefore, they will not be included again in this section.

Moreover, the reaction of the robot when there is not a dominant motivation is also analyzed in the last part.

9.3.1 Survival motivation. *How do I get my batteries recharged?*

Figure 9.4 displays the Q values related to all the objects in the robot's world when survival is the dominant motivation. This means that the need of energy is high. The best action, this is the action with the highest Q value, is *charge* which is responsible for the totally recharging of the batteries. Consequently, the energy required is obtained. For that reason, after this action has finished, the *energy* drive is satiated. Then, this action is the most likely to be executed. It is the consummatory action for the *survival* motivation.

The *go to player* action is very high too because the next best action is the *charge* action. This action is executed when the robot is unplugged and far from the docking station. This situation results after the execution of the *go to player* action.

It is worth mentioning why remaining plugged is not a good strategy in this situation, although it would seem a contradiction. Since the *remain* action just can be executed when the robot is plugged and this is after the *charge* action, it implies that the robot's battery is likely full and, consequently, *remain* does not contribute anything because survival will not be the dominant motivation at that situation, so the robot's wellbeing does not augments. Moreover, the amount of time this action lasts is not enough for a significant contribution to the level of energy. Concurrently, other drives increase a bit and therefore the variation of the robot's wellbeing is negative. Then, the value of this action is not good. In fact, *remain* has been executed when *survival* is the dominant motivation just when, due to the Well-balanced Exploration mechanism (Section 6.3.1), *energy* has been artificially saturated and Maggie was plugged.

The rest of actions are slightly positive because they provide little benefits in other drives different than the *energy* drive which is the one related to survival motivation. The actions that reduce the *energy* drive have the highest values.

9.3.2 Fun motivation. *Let's enjoy!*

In this case, the dominant motivation is *fun*. Then, the robot needs to satisfy the need of entertainment through the *dance* action (the consummatory action), which is the best action (Figure 9.5). For dancing, music must be on, so *play* music is the second better action due to the collateral effects of this action. Moreover, the *idle* action when music is off and it is close to the *cd player* is good too because the next best action with the *cd player* is to

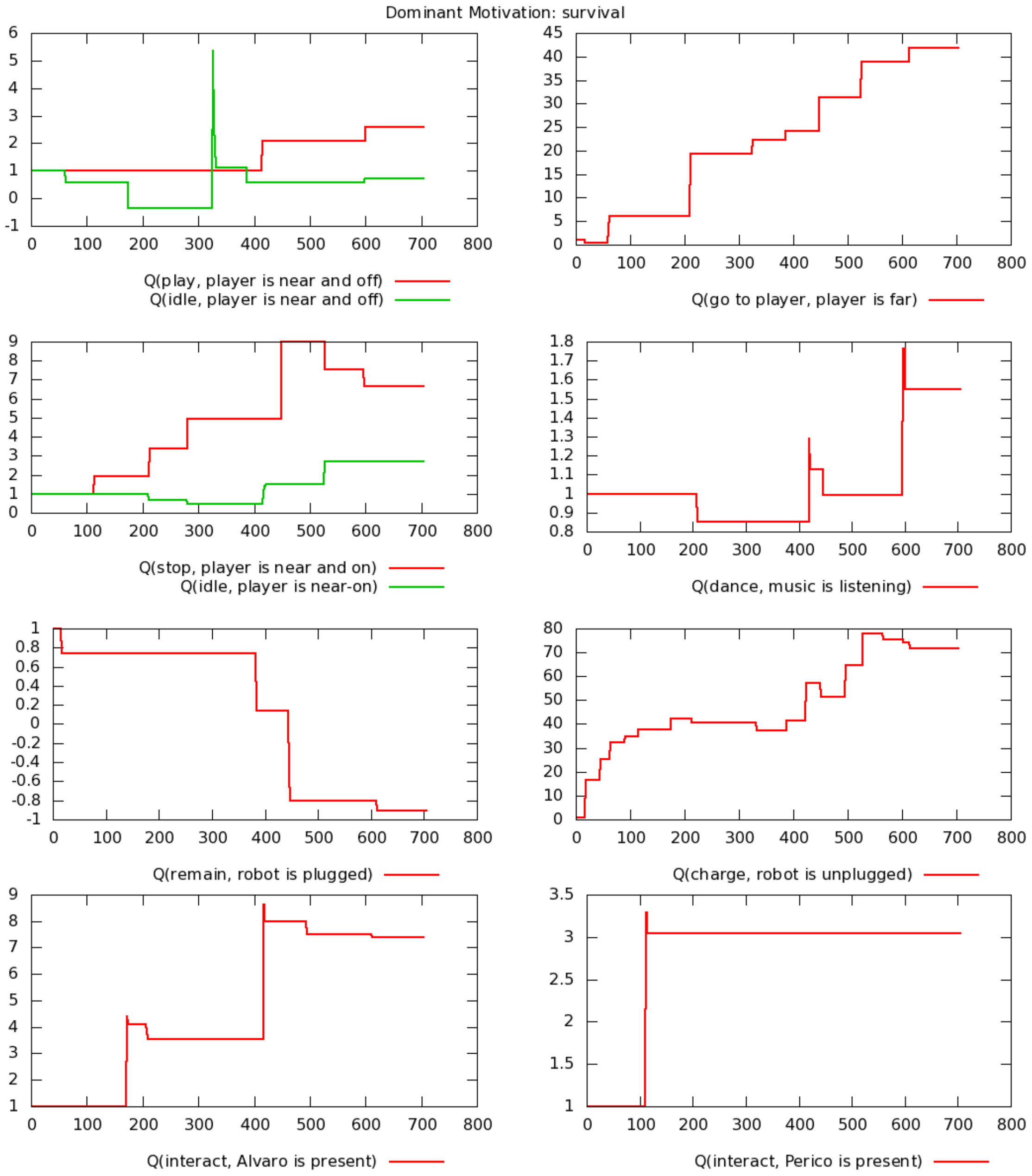


Figure 9.4: Learned Q-values when **survival** is the dominant motivation

play music, which is very good as well. In order to *play* music, Maggie must be close enough to the *cd player*, so, *go to player* is the next positive action. All these other actions are appetitive actions. This is a clear example about the advantages of Object Q-Learning algorithm and its collateral effects (Section 6.2).

The remaining actions are not suitable for this behavior. Therefore, their Q values are negative.

This motivation has already been extensively studied in Chapter 8 where details can be read.

9.3.3 Relax motivation. *I need calm!*

Now, the robot demands a quiet atmosphere, so the dominant motivation is *relax*.

Firstly, it must be emphasized that, if Maggie needs calm is because the music has being playing for some time. In other words, when the music is off, Maggie does not need to relax. Consequently, the Q values related to the actions executed when the *music player* is switched off and the robot is close to it (*play* and *idle*) does not change, so they remain at their initial value of 1 (top left Figure 9.6). This means that they have not been executed ever when the dominant motivation is *relax* because it is not possible.

After music is playing for a while, the robot feels the need of a peaceful environment. Then, it learns that it has to *stop* music (consummatory action). In consequence, this is the highest Q value. As it happens when *fun* is the dominant motivation, the robot must approach the *cd player* to operate it. In this case, this is necessary to *stop* music. Accordingly, *go to player* action (appetitive) is the next best action. Once the robot is in the proximity of the *cd player* (and the music is on), it can *stop* music or execute *idle* action. Since *stop* is the best action, *idle* value is very high as well. The reason is that when this action ends, the robot can *stop* music which is the highest Q value. All these Q values are plotted in Figure 9.6.

A significant negative value is assigned to the *charge* action. This action moves the robot far from the *music player*, which results in a very bad option because it cannot be switched off from far.

9.3.4 Social motivation. *Do you want to be my friend?*

As presented in Section 5.4.1, the *social* motivation is related to the need of positive human-robot interaction. Therefore, when the *social* motivation is the dominant one, the robot is encouraged to interact with the two users: *Alvaro* and *Perico*. Interactions with *Alvaro* and *Perico* have a great positive average effect over this motivation. Then, these actions are the most suitable skills to be executed: this is the reason because the highest Q values among all actions, when the dominant motivation is *social*, correspond to *interact-with-Alvaro*

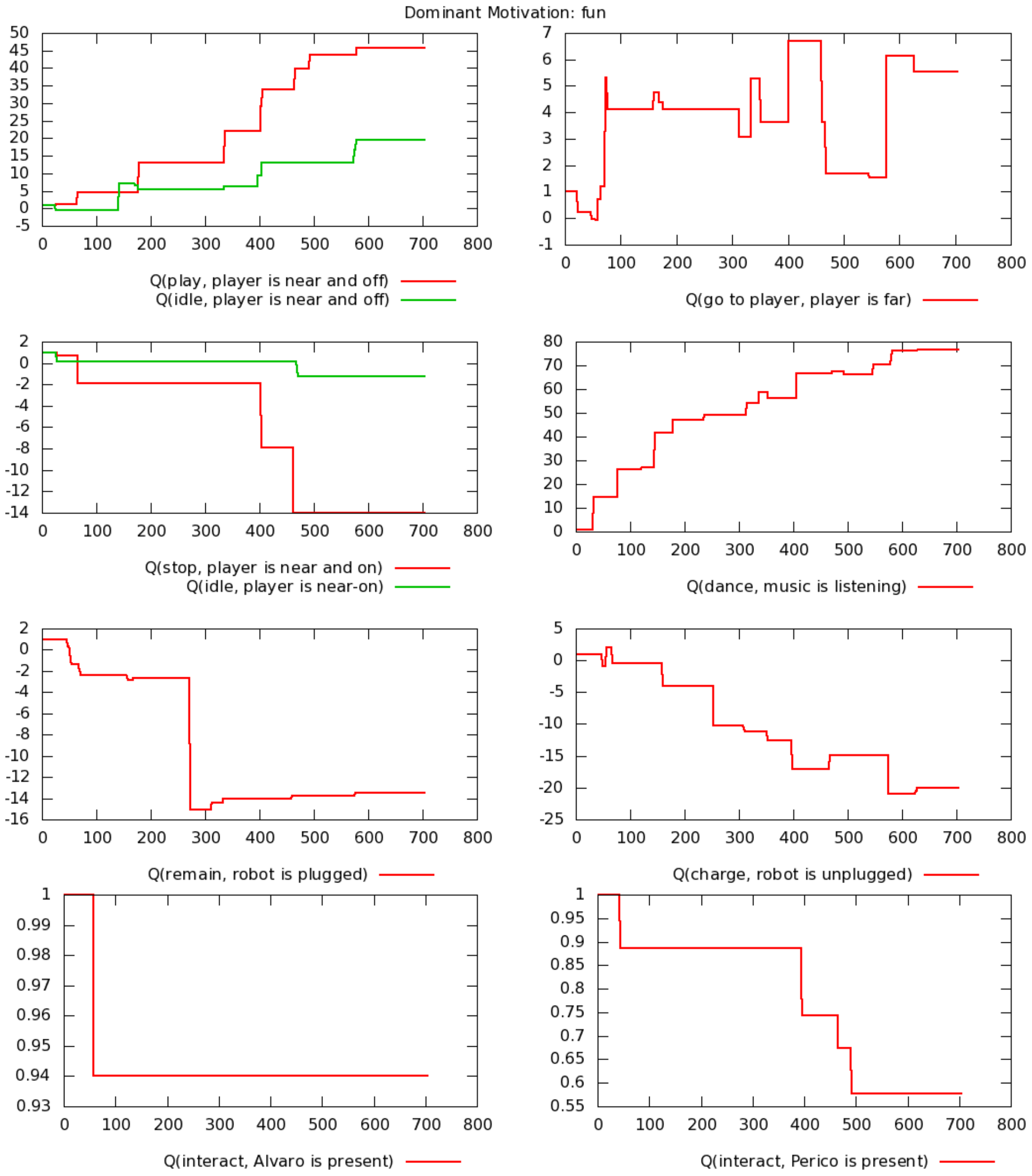


Figure 9.5: Learned Q-values when **fun** is the dominant motivation

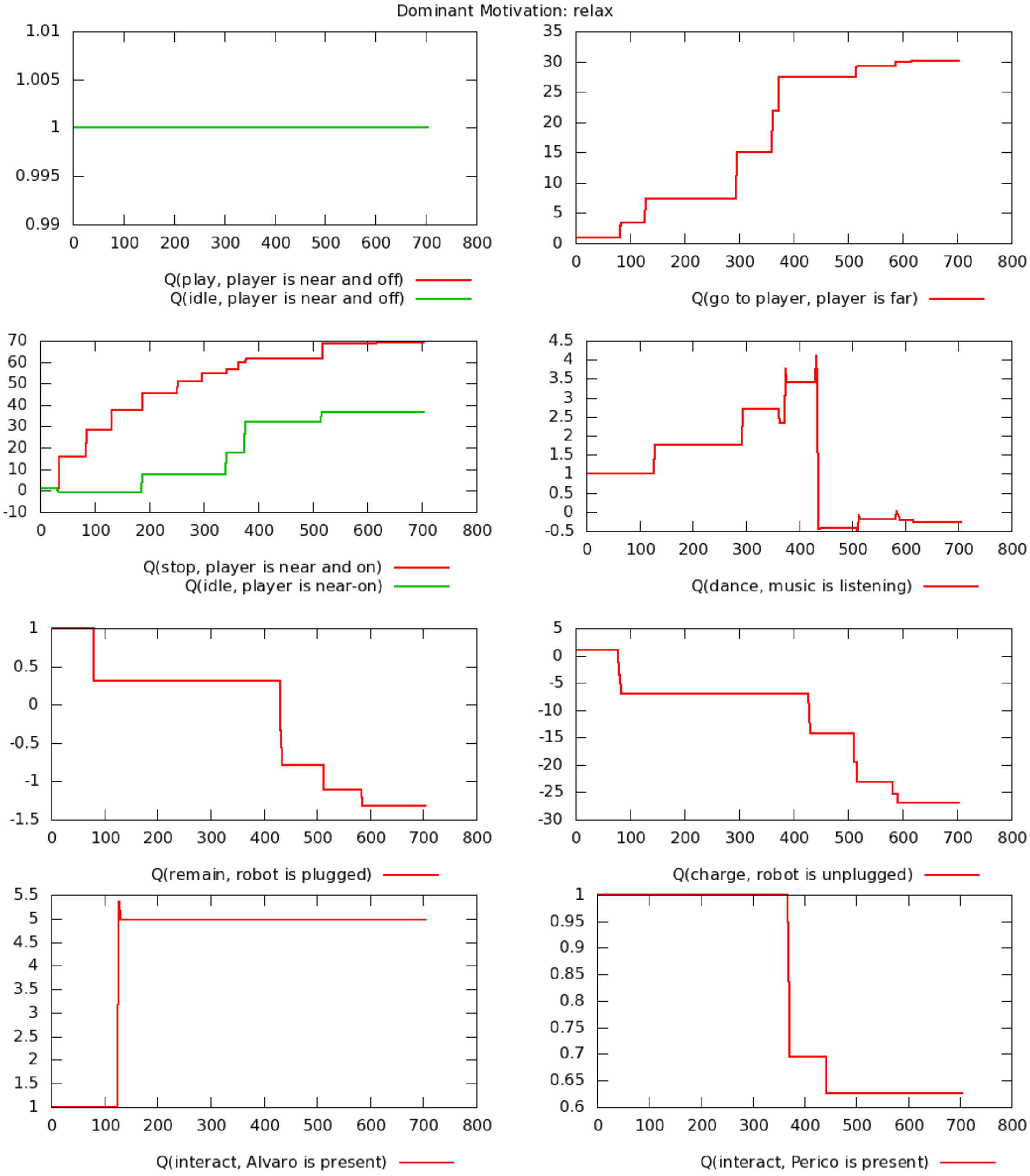


Figure 9.6: Learned Q-values when **relax** is the dominant motivation

and *interact-with-Perico* (see the highest values at bottom plots in Figure 9.7). These are consummatory actions.

The interaction with *Alvaro* must be detailed. *Alvaro*'s actions are, most of the times, favorable. However, he occasionally damages Maggie. Despite of the small percentage of hurting actions, the final Q value of *interaction-with-Alvaro* is quite high. However, the small number of hurting actions are enough to scare Maggie. Maggie is afraid of *Alvaro* because of the few negative interactions, which cause a 10 points penalization in its *social* drive (Equation (8.1)).

The plot in the bottom left corner in Figure 9.7 depicts the evolution of the Q_{social}*(Alvaro is present,interact with Alvaro)* value when *social* is the dominant motivation. Around the iterations 100 and 180, this value decreases because there has been an important decrement on the robot's wellbeing due to negative interactions. This is enough for Maggie to detect and remember the dangerous situation. Hereafter, whenever *Alvaro* is close to the robot, this situation is appraised as a dangerous state, and the *fear* motivation intensity exceeds the *social* motivation intensity. Therefore, whenever *Alvaro* is present, the *social* motivation will not be the dominant one again, and this Q value will not be updated anymore. This can be observed in the other motivations too (constant values of $Q(interact, Alvaro\ is\ present)$ after iteration 182), but not in *survival*. This is because the *survival* motivation was designed to guarantee that, in case it reaches its maximum level, it is always the highest motivation. This is considered as an inherited survival mechanism in nature: when animals are extremely hungry they can even risk their life for food. This is related to the saturation levels shown in Table 5.3.

How the robot reacts to *fear* has been detailed in Section 9.2.

Another issue worth mentioning is related to the rest of the actions when *social* is the dominant motivation. Users can approach Maggie at any time. From a social point of view, this exogenous action influences the robot's state and so the availability of endogenous actions; e.g. when a user is with the robot, it can interact with the user. However, it has been observed that users, most of the times, do not approach the robot when it is exhibiting a *lively* action like *dancing* or *going to player*. In contrast, they approach Maggie when it is doing other more *lethargic* actions. In particular, these *lethargic* actions are *idle* and *remain*. This is reflected on the Q values of these two actions (Figure 9.7): the Q values associated to these actions are the next highest actions after the two *interact* actions. This means, that when the robot needs to interact and there is no people around it, it will behave in a passive way by means of *idle* and *remain* actions (appetitive actions). It seems like users are reluctant to approximate Maggie as long as it is moving.

9.3.5 There is not dominant motivation. *I'm fine!*

An interesting result can be observed when there is no dominant motivation. This happens when the intensities of all drives are below their activation levels. This means that there is

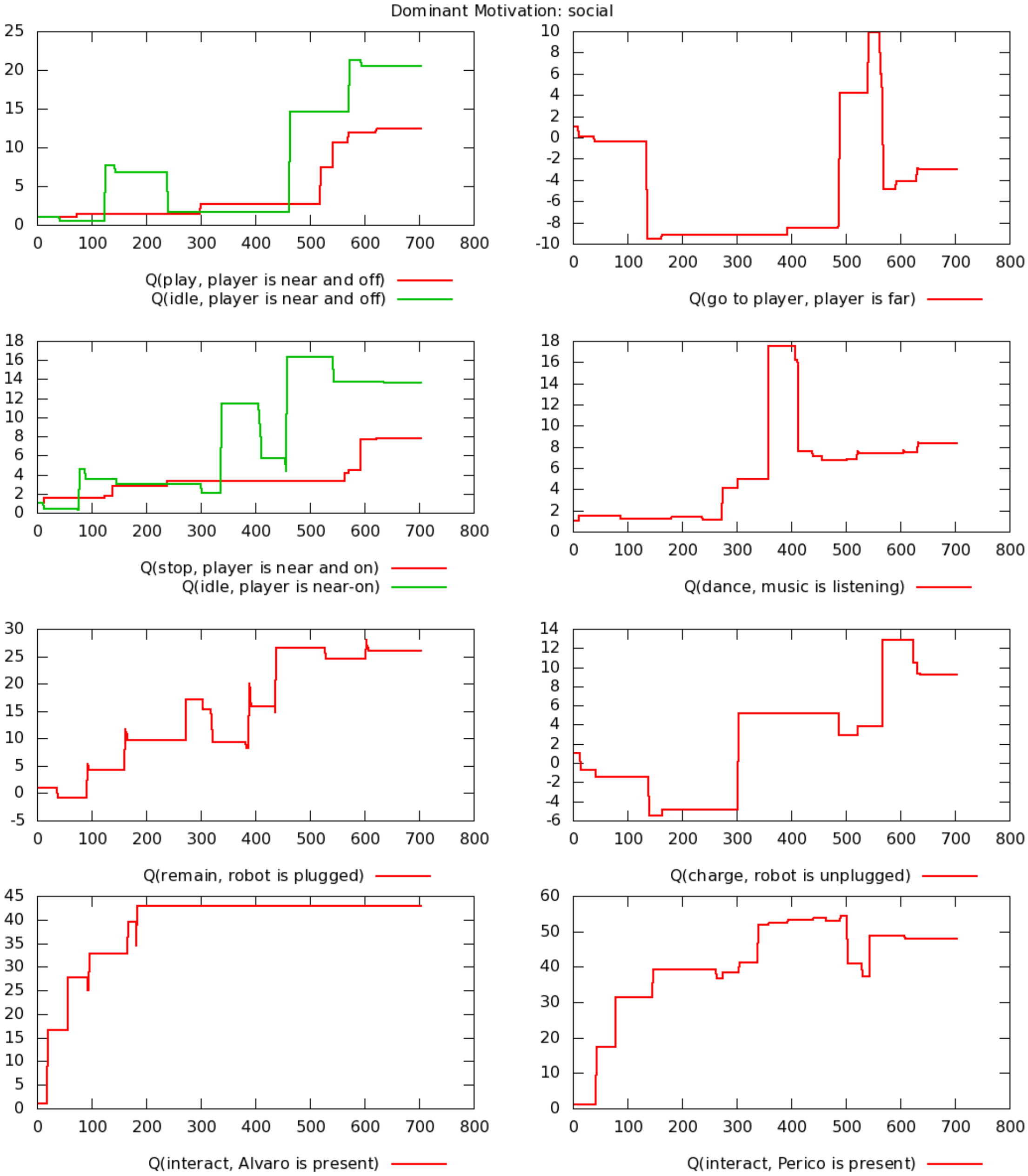


Figure 9.7: Learned Q-values when **social** is the dominant motivation

not any particular need that must be satisfied. Consequently, this situation corresponds to pleasant state. But, how does Maggie behave in this case? What does it do when there is not specific needs? The results are shown in Figure 9.8.

The values for all actions related to the need of fun are relatively high. This is because, as said before, *boredom* is one of the highest drives, so every time this action is executed it will likely receive positive reward. However, the most valuable action is the *charge* action. This produces a pattern of behavior where the robot is charging its battery or it turns the music player on and dances, even if it is plugged. This can be interpreted as the robot satisfies two basic needs even if they are not urgent. It is like if the robot foresees the most likely future needs and it gets ready in advance. These needs do not depend on other external elements and can be satisfied by the robot itself.

The rest of the actions are either slightly positive or negative, they are all around zero, but there are not really low or high values. This means that none of these actions play a crucial role in the absence of dominant motivation.

9.4 Summary

This chapter contains the results from the experiments where the robot's behaviors are learned. There are two sorts of experiments: the experiments related to the emotion of *fear*, and learned policy where *happiness* and *sadness* are used as the reinforcement function.

In the first section, the goodness of fear in Maggie has been exposed. The learning process of fear releasers endows the robot with a mechanism for identifying new dangerous states. Besides, these states are totally averted by means of the *fear* motivation. Different strategies can be learned to deal with these dangerous states according to how the environment reacts. For example, in the experiments, according to how people act when they are with the robot, the robot learns how to keep away from dangerous users. The final numerical results of fear certifies its benefit.

The second section describes the learned policy of behavior for each motivation. The robot has learned the correct behaviors to deal with each motivation in different situations. That is, Maggie has learned when to execute *appetitive actions* in order to enable the execution of *consummatory actions*.

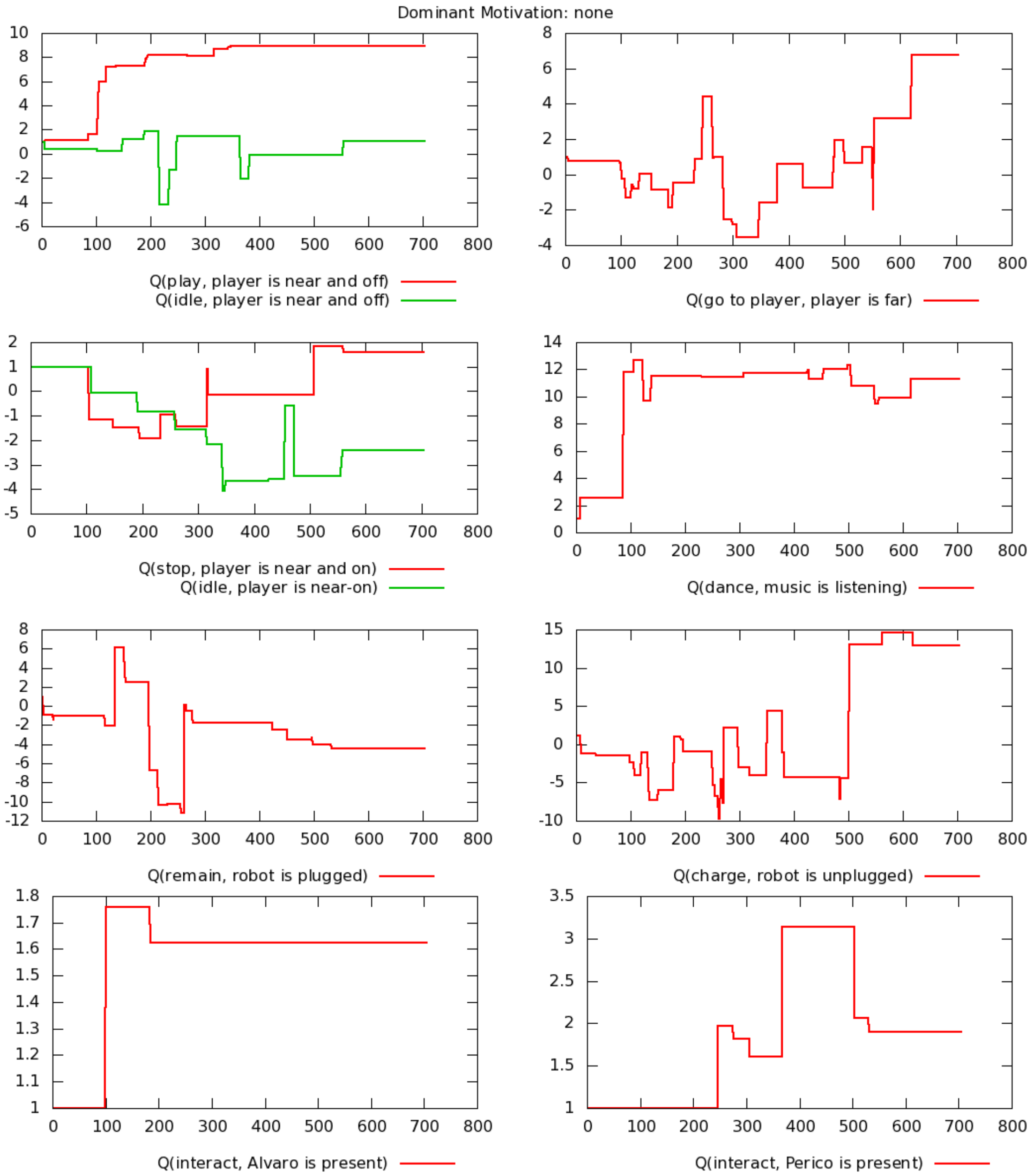


Figure 9.8: Learned Q-values when there is not a dominant motivation.

CHAPTER 10

Conclusions and Future Developments

10.1 Comments to the results

Since social robots move and interact with humans sharing the same areas, one of the main requirements for social robotics is a **natural behavior**. These natural robot's behaviors are in terms of similarity to humans' behaviors, or at least animals' behaviors (these are perfectly understandable and accepted by people). One of the advantages of using motivations and emotions in robots is that they allow animal-like responses to certain situations. In particular, **fear** has been successfully implemented in the robot Maggie in order to provide a natural mechanism of avoiding dangerous situations.

The presented thesis proposes a method which endows a robot with the capability to learn a policy of behavior autonomously, without any supervision, just by robot-environment interaction. Then, considering the **happiness** and **sadness** emotions, the robot learns what to do in every situation in order to survive and to maintain its needs satisfied.

The inclusion of motivations and drives in the DMS provides a flexible mechanism that leads the robot's behavior in every situation.

Moreover, the experiments and all parameters have been set considering that the robot lives in an environment with people, so, its behaviors should be *similar* to those exposed by its *world-mates* in an effort to make the robot's behavior understandable by people.

The resulting behaviors related to each motivation have been presented in Chapter 9. When the robot exploits the learned policy, **complex behaviors** are shown by series of simpler actions. For example, when the robot is motivated to have fun, it approaches the music player, turns it on, and then dances. In contrast, when the dominant motivation is relax, the robot approaches the music player and switch it off. In relation to social motivation, if the robot is alone, it decides to remain where it is until a person approaches and then they interact. Other behaviors look more elemental because just one single action is involved: when the battery are depleted the robot needs to survive so it gets its energy refilled by plugging to the docking station and remaining there. However, the mechanism under the hood is the same independently of the complexity of the consequent behaviors.

Behaviors are elicited due to the combination of the dominant motivation and the situation in the robot's world. But, if non of the drives exceeds its activation level, this results on a situation where there is not a dominant motivation. This means that there is not an urgent need so the robot is at a pleasant state. Learning has also been carried out in these cases, so the robot has also learned how to behave when it is *comfortable*.

In general, most of the resultant Q values when there is not a dominant motivation heavily fluctuate, so there is not a clear behavior. However, two state-action pairs are quite stable and have relative high Q values associated, what gives the idea that both actions will be likely selected. These state-action pairs are: the *play* action when it is close to the *player* and the music is off, and the *dance* action when the *music* is being listened. This implies that when dominant motivation does not exist, the robot will likely turn the music player on and dance. Why is so? Both actions are related to the behavior exhibited when fun is the dominant motivation. Since this motivation is one of the fastest one and due to the fact that it does not depend on external agents, it almost always gets a positive reward. Moreover, these two actions are relative short on time (specially the *play* action which takes around few seconds), and then the increment on drives is minimum. Therefore, the potential decrement in the robot's wellbeing is minimum. From other perspective, as just said, *fun* is one of the fastest motivation and, during learning, it was frequently the dominant motivation, i.e. the robot frequently needs to have fun. This reaction (dance when the dominant motivation does not exist) can be understood as a mechanism preventing from the most probable future need of entertainment.

During the exploiting session, observing the robot's behavior without a dominant motivation (this is most of the time) gives the impression of a "**dance-aholic**" robot. Recalling the experiments carried on by Olds and Milner in 1950s [61], rats rapidly became addictive to electrical self-stimulation into certain areas of their brains. This leaded to the discovery of the called pleasure centers. The behavior exhibited by the robot seems similar to how these rats acted: it is like the robot's pleasure center is being stimulated while dancing, so Maggie becomes addicted to dancing.

In relation to the emotion of **fear**, it has been successfully implemented in the robot Maggie in order to guide its behavior providing a natural mechanism for avoiding dangerous situations. Fear is treated as a motivation which moves to behave. In addition, an original appraisal mechanism of fear has been implemented and it allows to identify non-predefined dangerous situations. The fear motivation is elicited when a dangerous situation is detected. These circumstances are not predefined, but they are **appraised** by the robot through interaction. Therefore, the robot is able to identify by itself the conditions which cause fear.

Permanent harmful exogenous actions can be easily avoided by traditional reinforcement learning algorithms. However, when few negative experiences in relation to exogenous actions have been suffered in a specific situation, it is not easy to identify it as a potential dangerous situation. Nevertheless, the presented method is able to assess them as a dangerous situations too. The proposed appraisal of fear nicely works with states where the robot is sporadically harmed as well as states where it is constantly damaged. Once the dangerous states are recognized, the robot is able to learn what to do for avoiding them. This is achieved when fear becomes the dominant motivation.

Remembering the experiments achieved by Klüver and Bucy (Section 2.4.7), monkeys' behavior were studied in relation with fear. Normal monkeys are afraid of people, but the suppression of the amygdala causes some kind of fearlessness in monkeys: people touched them, stroke them, and even picked them up. Therefore, fear provides monkeys, and animals in general, with the required behavior at certain situations to **survive**. This same kind of behavior has been exhibited by the robot during the experiments where the *fear* emotion is not considered. Maggie has learned that when a certain situation is dangerous, it moves to other place far from where the danger is. When fear is not included as a motivation, Maggie's behavior corresponds to the same one exhibited by an animal suffering an amygdalectomy, similar to Klüver and Bucy's monkeys: it is not able to perfectly identify the dangerous situations when fear does not exist (i.e. like if the "robot's amygdala" has been removed).

In fact, Maggie learns the proper behavior to avoid dangers. As presented on the experiments (Sections 9.2.2 and 9.2.3), depending on different people attitudes, the danger-avoidance behavior could differ: as exposed in the previous paragraph, one behavior is to **run away** from where the danger is, but the other is to **remain still** until the threatening person gets bored and goes. This is also a common human behavior observed in terrified people: some people are stunned when they face a great danger. Other example can be observed in some chickens: after a chicken is frighten, it crouches down and trembles with fear.

However, the origin of this behavior differs: in animals, this is an unconscious, bodily reaction which makes muscles tensed. In the robot, the reaction is provoked because the learned values indicate that the danger will disappear after. Nevertheless, both responses, in animals and in Maggie, are **automatic** because the exhibited fear behavior is formed without any perspective into the future, just by executing the best action at each moment. The decision making process selects the next action considering the current available information. Then, there is not any planing looking into the future, thus, there is not deliberation.

In this work, reactions to fear (similarly to the reactions to the other motivations) have been learned by the robot through interaction with its world. In animals, some reactions to fear are inherited, this is, they are instinctive. Instincts are innate behaviors that are not highly dependent on specific learning experiences performed by the individual [50]. In fact, instinctive behaviors have been learned by the species through evolution. The experiments have shown that the results obtained from evolution and from the proposed mechanism are similar: escaping or freezing reactions are observed in both. This can be seen as another proof of the good performance of the proposed system because the behaviors exhibited as consequence of fear are analogous: the reactions to fear learned by the robot are comparable to those innate reactions exhibited by animals.

Besides escape and freeze, in nature, there is another well-known reaction to fear: fight. Due to the possible ethical problems, the robot has not been endowed with actions related to fighting and, consequently, Maggie cannot exhibit this kind of reactions.

Fear in animals is related to **anxiety**. Anxiety and its bodily reactions are proportional to the intensity of danger and, by extension, to the intensity of fear. One does not feel the same level of fear when one takes a ride on the roller coaster than when a criminal points you with a gun. However, in this work this is not considered, and fear is a binary variable: it is afraid or it is not. Therefore, the level of fear perceived by Maggie is constant for all the circumstances that evoke fear.

Moreover, in the proposed system, once a dangerous state is identified, this is not forgotten ever. This is based on the theory that memories associated with fear are quickly formed and **long-lasting** [52]. However, this situation could lead a robot to suffer some kind of anxiety disorders typical of humans beings. Imagine a long-lasting experiment which takes several days. At the beginning, during the first hour, the robot identifies the presence of person *A* as dangerous because *A* has hit the robot few times. Despite the fact that all the rest of actions carried out by *A* during the rest of the days were always positive, the system remembers always the painful initial interactions between Maggie and *A*. Consequently, if *A* is present, then fear emerges on Maggie during the rest of the experiment.

From a psychological perspective, this can be seen as an inappropriate experience of fear which is related to **anxiety disorders**. There are some points in common with Post-traumatic Stress Disorder (PSD). Approximately, PSD is related to intense or unrealistic worries suffered when the stimuli related to a past trauma are present. Even if the person *A* damaged Maggie at the very beginning, and he has not done it again in several days, which suggests that this behavior hardly will be repeated, fear arises in the presence of *A*. Also, similarities with a phobia provoked by exposure to situations leading to avoidance behaviors can be found. In particular, a sort of social phobia can be identified because any social interaction with a specific person is avoided, even if it seems that he will not induce any damage. At this point, it seems that traumas on humans are very hard to re-program. This is exactly what happens to the robot as well.

As proved in Section 9.2.4, the average wellbeing does not improve when fear is considered, actually, it is slightly lower. People in fear live distressed, and this fact is shown in Maggie as well. However, some other benefits justify the use of fear. First of all, by means of fear the robot has avoided all harmful exogenous actions: Maggie has not been hurt anymore by a user. Moreover, the permanence within comfortable levels of wellbeing is better when fear is present since it is not hit anymore. Additionally, the quality of life can be also measured as the amount of time that a particular behavior is not required, i.e. there is not dominant motivation. Also in this case, the experiment which considers fear outperforms.

From the point of view of human-robot interaction, the behaviors displayed by the robot are rather **animal-like**. This helps to improve the interaction when the robot is *living* with people and validates the followed approach.

10.2 Contributions and achievements

As mentioned in Section 1.2, the main goal of this dissertation is to improve the autonomy of a real robotic platform. This has been achieved by extending its control architecture with a bio-inspired DMS.

This DMS has several drives, motivations, and emotions which shape the robot's behavior. The followed approach of using the *happiness*, the *sadness*, and the *fear* emotions in a social robot is one of the novelties of this work.

In particular, *fear* has shown promising results. The implementation of a method for learning the appraisal of new fear elicitors, as well as the reactions to fear, by the social robot Maggie provides a powerful adaptive method which increases the possibilities of a better quality of "life" for the robot.

Moreover, the design of the DMS proposed by the author allows to apply the same model to different robots independently of the control architecture.

In relation to the learning process, the Object Q-Learning algorithm proposed in [49] has been improved by adding two modifications which make it possible to learn a correct policy of behavior in an acceptable amount of time.

This research has ended up to a lively robot whose behavior is defined by the robot itself, so it provides the illusion of life. This is because the emergent behaviors observed in the robot are comparable to those observed in living beings. This is the validation of the goodness of the motivational and emotional mechanisms involved in the DMS.

10.3 Fulfillment of the objectives

In Chapter 1, a set of objectives were listed as sub-goals that must be achieved in order to reach the main goals. Following, the level of achievement of each one of these objectives is detailed.

- The Object Q-Learning (Section 6.2) algorithm has been successfully implemented in the robot. By means of it, the robot has learned the proper sequence of actions (behavior) with different objects according to the highest motivation and the robot's world configuration. In addition, the learning algorithm has been modified to speed up the learning process. Two new mechanisms has been integrated: the Amplified Reward and the Well-balanced Exploration. In short, both allow to learn the policy of behavior faster.

- The robot has been endowed with a set of skills which allow it to interact with several objects. Some of these skills perceive the different objects and define the state of this object in relation to the robot; people are perceived by blue-tooth and RFID technology, the location of the robot and the other static objects are determined by means of the robot's navigation system, the charger is detected using a data acquisition board, etc. Other skills perform some actions with the objects: go to the music player and turn it on, dance with the music, recharge the battery, etc.

- The decision making model proposed by Malfaz in [49] has been adapted to and implemented in a real robot. It has been successfully integrated into the AD architecture which controls the robot. The elements of this architecture has not been modified at all, but the decision making module has been added as an extension. Actually, this model can be easily integrated in other control architectures or robots with minimum effort.

- The implementation of the DMS has been designed following the principle of flexibility. A database has been designed were all required information for the DMS is stored as tables. The inclusion of new parameters, new drives, new motivations, new effects, etc, is as easy as include new entries in the corresponding table.

- The emotions of happiness, sadness, and fear have been analyzed from a functional perspective. After defining its potential applications to robots, they have been integrated in the system according to the required functions. Particularly, the artificial emotion of fear helps to improve the robot's "quality of life" and provides a mechanism to "live" more secure.

- Humans have been considered as a sort of "objects" that the robot can make use of them for its own goals. However, human reactions can not be easily predicted; therefore, the robot has been endowed with mechanisms for evaluate the human actions and, accordingly, react.

- Focusing on the results observed in relation to the artificial emotion of fear, it seems clear that its utility is relevant to the performance of the robot. Moreover, its inclusion in the robot's DMS has shown animal-like behaviors learned by the robot itself. This is probably one of the main achievements of this thesis.

- The implemented DMS works in an automatic manner: the system just considers the available information at an instant, there is not a model used to predict the effects in the future. Then, there is not reasoning behind the proposed decision making system. The behavior is formed by selecting the most appropriate actions at each moment. Therefore, the whole process is an automatic process where deliberation is not involved. In the next versions of the DMS, it is planned to build a model to predict the consequences of the robot's actions. Its results will be compared with the current model-free approach in order to come up with the pros and cons of both approaches.

10.4 Future works and limitations

This work presents some challenges to be accomplished in the future:

- So far, the system makes decisions just after the previous action has finished. However, it could be more realistic (animal-like) to add the possibility of the interruption of the current action in case of a relevant event. For example, if the robot is interacting with a person but, in the meantime, the energy reaches a low level, then the robot should be able to interrupt the *interact* action and to recharge its battery. This requires a safe mechanism to interrupt the control loop of a skill in the AD architecture. Then, the evaluation of the convenience of executing a new action could be triggered each time a new event happens, after certain time without any update, or after an action has finished.

- During the experiments, the effects of the exogenous actions are considered during the robot's actions which do not cause any effect in the robot or the environment (the robot's *interact* actions). Therefore, all the variations in the robot's wellbeing during these effect-less actions are due to the exogenous actions. This is an unrealistic approach because the exogenous actions can be executed whenever the other agent decides it (independently of what the robot is executing). In the future, a probabilistic method should be proposed in order to forecast the execution of exogenous actions and asses their effects, considering that they follow a stochastic process.

- The final robot's behavior heavily depends on the parameters assigned to the elements in the DMS. Different configurations of these parameters may lead to undesired behaviors or behaviors that are far from the biological approach followed in this thesis. For example, if the satisfaction times are very small and drives increase very fast, the learning process is very difficult due to an abnormal number of motivations competing at the same time. Moreover, if the effects of the actions over the drives are not fine tuned, the learned behavior can be different from the expected. It could happen that if the robot needs to relax, and it is plugged to the docking station, it "prefers" to dance plugged to approach to the music player for turning it off. This results in an unsatisfied need. Future studies of these parameters will clarify how they influence in the robot's performance and its "personality". These robot's "personalities" will be studied in relation to their influences in different users.

- In the near future, as the functionality of the robot and its environment becomes more and more complex, it will have to cope with new situations. This could lead to the inclusion of new drives, motivations, or emotions, or a redefinition of the existing ones. Moreover, more complex functions may require more complex relations between the DMS elements. For example, a motivation may be related to several drives (e.g. the motivation to have fun could be related to the boredom but also to the energy). Furthermore, several drives could be altered by the same action and several actions could satiate the same drive.

- The presented experiments have been carried out in a controlled scenario, the lab, where possibilities are limited. In the next future, robots will be moved closer to users' environment (houses, hospitals, or schools) where they interact with people without previous knowledge about robotics. The aim will be to improve people's quality of life acting as a game-partner, study-partner or companion. The proposed DMS will be applied to these robots which will coexist with elders and children at their homes or hospitals. Moreover, seeing that robots can make their own decisions, they will be able to initiate human-robot interaction showing proactive behaviors. This is a really interesting capacity when dealing with people suffering social diseases which can be studied.

- In this thesis, two different phases during the experiments have been presented: exploration and exploitation. These phases are differentiated according to the *temperature* parameter which balances both phases. The tuning of this parameter is made at design time (hand-coded) which results in a very steady system: first the robot learns, and, at some point, it does not learn any more and exploits the learned policy. However, if new situations emerge later, the proper behavior in that new situations will not be learned. In order to tackle this problem, a new formal method could be based on the variability of the data which could be related to the cognitive concept of *curiosity*. For example, Breazeal [4] proposed a *curiosity* drive for balancing exploration verses exploitation during robot's learning, so it correlates the amount of novelty over time; e.g. if the robot's environment is too predictable, this drive could lead it to novel contexts. This opens an interesting new research line.

- The motivations considered in this thesis take into account physiological and psychological needs, i.e. they are related to deficits on drives. However, as presented in Chapter 2, motivations in humans can also be related to hedonic factors. These hedonic factors in motivations form a novel research field in robots which look for *pleasure*.

- Since the DMS is applied to social robots, it considers the state of the person who is interacting with it. In this work, just the position of the users in relation to the robot is considered. However, the robot's behavior can be altered depending on the person humor (happy, sad, angry, etc.), how the person feels (tired, bored, etc.), where he is (far or close to the robot, at the kitchen or at the toilet, etc.), or what the person is doing (eating, sleeping, working, etc.). All these perceived states would endow the robot with a kind of empathy, which will improve the social interaction quality.

- In this work, the fear is related to dangerous states where the robot can be harmed due to the other agent's action (the exogenous action of active objects), i.e. the robot is afraid if it is in a situation where it can be potentially damaged. Nevertheless, the action performed by the individual itself can also be harmful (imaging you walk a tightrope). In this case, these are risky actions and fear also comes up because of them (e.g. you are afraid of walking a tightrope). Risky actions have already been studied in virtual agents [208] and they will be considered in the robot in future works.

- In this implementation of fear, dangerous states are learned and never forgotten. In future works, fear will be enhanced with mechanisms to take into account the dynamic aspects of fear making it more flexible. Fear will be able to be reprogrammed in order to "forget" the old dangerous states under certain conditions.

- Cañamero proposes in [149] the study of emotional disorders by simulating maladaptive artificial emotions. The proposed system can be configured for analyzing the consequences of maladaptive artificial emotions. The ever lasting memory of dangerous states is an example. This could be a promising line of research considering the benefits of studying this kind of disorders in artificial creatures, comparing to the potential ethical problems of experimenting with living beings.

- This thesis has been focused on the internal component of emotions, the experience of emotions. However, if a more realistic use of bio-inspired emotions in robots is desired, the external component is a must. Consequently, the expression of emotions according to the emotional state of the robot is one of the coming steps.

10.5 Final comments

The existing approaches to use artificial emotions in robot (including this thesis) make strong simplifications about *natural* emotions. Despite of these significant simplifications, the observed results envision promising applications. Thus far, these applications might seem simple considering just the external appearance. However, under the hood, the ins and outs of the bio-inspired mechanisms move us a step forward towards the full understanding of the existing processes in the brain. The collaboration of interdisciplinary researchers (neuroscientist, biologist, bio-engineers, and many other specialists) is probably the only way to achieve it.

In this dissertation, the applied artificial emotions just implement few of the functions of their counterparts in living beings. Due to the amount of functions assigned to emotions according to the last investigations, it is rather difficult to create artificial *creatures* endowed with emotions covering all of them. Actually, it is possible that not all these functions are required or even not all emotions are desired. For example, *loathing* apparently is not a desire emotional state in a social robot. Accordingly, robots must be endowed just with the required emotions and functions that they need for achieving their tasks.

Moreover, “machines” making decisions by themselves terrifies many people mainly due to the science fiction films where robots rule the world. This catastrophic view of robots is rather far from reality. Nowadays, robots just can execute actions that they have been intended for. Therefore, similarly to the robot Maggie, if it is not designed for fighting, it will not develop fighting skills.

A different aspect is related to the responsiveness of the robot’s actions. Since researchers are working on robots making their own decisions, who is responsible of those decisions? The designer? The owner? The robot itself? Currently, there is not a clear agreement in the scientific community either this issues are under the cover of new laws. However, researchers are already working on this topic and its ethical implications.

Bibliography

[1] J. D. Velásquez, "When Robots Weep : Emotional Memories and Decision-Making," in *Artificial Intelligence*, pp. 70–75, JOHN WILEY & SONS LTD, 1998.

[2] O. Avila-García and L. Cañamero, "Hormonal modulation of perception in motivation-based action selection architectures," in *Proceedings of the Symposium on Agents that Want and Like: Motivational and Emotional Roots of Cognition and Action*, pp. 9–16, SSAISB, 2005.

[3] S. C. Gadanho, "Learning Behavior-Selection by Emotions and Cognition in a Multi-Goal Robot Task," *Journal of Machine Learning Research*, vol. 4, no. 3, pp. 385–412, 2003.

[4] C. L. Breazeal, *Sociable machines: Expressive social exchange between humans and robots*. PhD thesis, Massachusetts Institute of Technology, 2000.

[5] L. Moshkina, S. Park, R. C. Arkin, J. K. Lee, and H. Jung, "TAME: Time-Varying Affective Response for Humanoid Robots," *International Journal of Social Robotics*, vol. 3, pp. 207–221, Feb. 2011.

[6] N. Esau and L. Kleinjohann, "Emotional Robot Competence and Its Use in Robot Behavior Control," in *Emotional Ingineering* (S. Fukuda, ed.), ch. Emotional, pp. 119–142, Springer London, 1st ed., 2011.

[7] Z. Kowalczuk and M. Czubenko, "Intelligent decision-making system for autonomous robots," *International Journal of Applied Mathematics and Computer Science*, vol. 21, pp. 671–684, Dec. 2011.

[8] M. A. Martínez Vidal and S. Muriel de la Riva, “INE-Spain strategy on population estimates and projections facing the challenge.” 2010.

[9] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, “Experiences with an interactive museum tour-guide robot,” *Artificial Intelligence*, vol. 114, no. 1-2, pp. 3–55, 1999.

[10] F. Littmann and J. Riviere, “A remote-operated system for interventions on explosives,” in *Proceedings of the ANS Seventh Topical Meeting on Robotics and Remote Systems*, vol. 2, pp. 1038–42, 1997.

[11] S. Tetsudo, N. Hisato, N. Daisuke, U. Hiroyuki, and K. Yukihiko, “Autonomous mobile robot system for delivery in hospital,” *MEW Technical Report*, vol. 53, no. 2, pp. 62–67, 2005.

[12] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, “A Communication Robot in a Shopping Mall,” 2010.

[13] A. Casals, R. Merchan, E. Portell, X. Cuf, and J. Contijoch, “Capdi: a robotized kitchen for the disabled and elderly,” in *Proceedings of the Assistive Technology an the Threshold of the New Millennium. AAATE 99*, (Dusseldorf, Germany), pp. 346–351, 1999.

[14] K. Schilling, M. Mellado, J. Garbajosa, and R. Mayerhofer, “Design of flexible autonomous transport robots for industrial production,” in *Proceedings of the IEEE International Symposium on Industrial Electronics*, vol. 3, p. ISIE ’97, 1997.

[15] A. Bicchi, A. Fagiolini, and L. Pallottino, “Toward a Society of Robots Behaviors, Misbehaviors, and Security,” *IEEE Robotics and Automation Magazine*, vol. 17, no. 4, pp. 26–36, 2010.

[16] N. Kubota, Y. Nojima, N. Baba, F. Kojima, and T. Fukuda, “Evolving pet robot with emotional model,” Proceedings of the 2000 Congress on Evolutionary computation, 2000.

[17] R. A. Brooks, “From earwigs to humans,” *Robotics and Autonomous Systems*, vol. 20, no. 2-4, pp. 291–304, 1997.

[18] W. Maier and E. Steinbach, “A probabilistic appearance representation and its application to surprise detection in cognitive robots,” *Autonomous Mental Development, IEEE Transactions on*, vol. 2, pp. 267 –281, December 2010.

[19] Y. Zhang and J. Weng, “Spatio-temporal multimodal developmental learning,” *Autonomous Mental Development, IEEE Transactions on*, vol. 2, pp. 149 –166, September 2010.

[20] R. C. Arkin, “Robots that Need to Mislead: Biologically-inspired Machine Deception,” *IEEE Intelligent Systems*, 2012.

[21] J. LeDoux, *El cerebro emocional*. Ariel/Planeta, 1996.

[22] A. Damasio, *Descartes’ Error - Emotion, reason and human brain*. Picador, London, 1994.

[23] S. C. Lewis, *Computational Models of Emotion and Affect*. PhD thesis, University of Hull, 2004.

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

[25] R. W. Picard, *Los ordenadores emocionales*. Ed. Ariel S.A., 1998.

[26] E. Rolls, *Emotion Explained*. Oxford University Press, 2005.

[27] R. C. Arkin, *Who needs emotions? The brain meets the robots*, ch. Moving up the food chain: Motivation and Emotion in behavior-based robots. Oxford University Press, 2004.

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

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

[30] T. Ziemke and R. Lowe, “On the role of emotion in embodied cognitive architectures: From organisms to robots,” *Cognitive Computation*, vol. 1, no. 1, pp. 104–117, 2009.

[31] C. Breazeal, *Designing Sociable Robots*. The MIT Press, 2002.

[32] S. M. Veres, L. Molnar, N. Lincoln, and C. Morice, “Autonomous vehicle control systems – a review of decision making,” *Control Engineering*, vol. 225, no. I2, pp. 155–195, 2011.

[33] J. Gancet and S. Lacroix, “Embedding heterogeneous levels of decisional autonomy in multi-robot systems,” *Distributed Autonomous Robotic Systems*, vol. 6, pp. 263—-272, 2007.

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

[35] B. Hardy-Vallée, “Decision-making in robotics and psychology: A distributed account,” *New Ideas in Psychology*, pp. 1–14, octubre 2009.

[36] M. J. Matarić, *The Robotics Primer*. The MIT Press, Sept. 2007.

[37] M. Mataric, “Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior,” *Trends in Cognitive Science*, vol. 2(3), pp. 82–87, 1998.

[38] A. Bechara, H. Damasio, and A. R. Damasio, “Emotion, decision making and the orbitofrontal cortex.,” *Cerebral cortex New York NY 1991*, vol. 10, no. 3, pp. 295–307, 2000.

[39] J. Velsquez, “When robots weep: Emotional memories and decision making,” in *Proceedings of AAAI-98*, 1998.

[40] L. Cañamero, “Designing emotions for activity selection,” tech. rep., Dept. of Computer Science Technical Report DAIMI PB 545, University of Aarhus, Denmark, 2000.

[41] S. Gadanho, “Learning behavior-selection by emotions and cognition in a multi-goal robot task,” *The Journal of Machine Learning Research. MIT Press Cambridge, MA, USA*, no. 4, pp. 385–412, 2003.

[42] M. Malfaz and M. Salichs, “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*, 2009.

[43] F. Michaud, F. Ferland, D. Létourneau, M.-A. Legault, and M. Lauria, “Toward autonomous, compliant, omnidirectional humanoid robots for natural interaction in real-life settings,” *Paladyn*, vol. 1, pp. 57–65, marzo 2010.

[44] T. Ziemke, “On the role of emotion in biological and robotic autonomy,” *Biosystems*, vol. 91, no. 2, pp. 401–408, 2008.

[45] J. J. Bryson, “Robots should be slaves,” in *Close Engagements with Artificial Companions Key social psychological ethical and design issues* (Yorick Wilks, ed.), pp. 1–12, John Benjamins, 2010.

[46] K. Lorenz, *Behind the Mirror*. 1977.

[47] K. Doya, “What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?,” *Neural networks*, vol. 12, no. 7-8, pp. 961–974, 1999.

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

[49] M. Malfaz, *Decision Making System for Autonomous Social Agents Based on Emotions and Self-learning.* PhD thesis, Carlos III University of Madrid, 2007.

[50] E. Kandel, J. Schwartz, and T. Jessell, *Principles of Neural Science*. Elsevier, 1991.

[51] A. Veldhuis, *Reviewing Decision Making: from awareness to social decision making*. Master thesis, University Utrech, 2011.

[52] M. Bear, B. Connors, and M. Paradiso, *Neuroscience: Exploring the brain*. Lippincott Williams & Wilkins, 2001.

[53] K. C. Berridge, “Motivation concepts in behav-ioural neuroscience,” *Physiology and Behaviour*, no. 81, pp. 179–209, 2004.

[54] B. Baars and N. Gage, “Cognition, brain, and consciousness: Introduction to cognitive neuroscience,” 2010.

[55] C. L. Hull, *Principles of Behavior: An Introduction to Behavior Theory*, vol. 25 of *The Century psychology series*. Appleton-Century, 1943.

[56] K. Cherry, “Drive-Reduction Theory. Hull’s Drive-Reduction Theory of Motivation.”

[57] C. Hull, “The conflicting psychologies of learning—a way out.,” *Psychological Review*, 1935.

[58] D. P. Schultz and S. E. Schultz, *A history of modern psychology*. Thomson/Wadsworth, 2005.

[59] J. Santa-Cruz, J. M. Tobal, A. C. Vindel, and E. G. Fernndez, “Introduccin a la psicologa.” Facultad de Psicologa. Universidad Complutense de Madrid, 1989.

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

[61] J. Olds and P. Milner, “Positive reinforcement produced by electrical stimulation of septal area and other regions of rat brain.,” *Journal of Comparative and Physiological Psychology; Journal of Comparative and Physiological Psychology*, vol. 47, no. 6, p. 419, 1954.

[62] J. Deutsch and C. Howarth, “Some tests of a theory of intracranial self-stimulation.,” *Psychological Review*, vol. 70, no. 5, p. 444, 1963.

[63] J. LeDoux, “The emotional brain,” *New York*, vol. 94, no. 4, pp. 1–29, 1996.

[64] C. Darwin, *The Expression of the Emotions in Man and Animals*, vol. 232. John Murray, 1872.

[65] T. Rumbell, J. Barnden, S. Denham, and T. Wennekers, “Emotions in autonomous agents: comparative analysis of mechanisms and functions,” *Autonomous Agents and MultiAgent Systems*, vol. 25, no. 1, pp. 1–45, 2011.

[66] D. Cañamero, “A Hormonal Model of Emotions for Behavior Control,” 1997.

[67] F. Gordillo, J. Arana, L. Mestas, and J. Salvador, “Entre la razón y el corazón: La importancia de la emoción en la toma de decisiones,” *Ciencia Cognitiva: Revista Electrónica de Divulgación*, vol. 5, no. 1, pp. 25–27, 2011.

[68] C. Castelfranchi, “Affective appraisal versus cognitive evaluation in social emotions and interactions,” *Affective interactions*, pp. 76–106, 2000.

[69] F. D. Rosis, C. Castelfranchi, P. Goldie, and V. Carofiglio, “Cognitive Evaluations And Intuitive Appraisals: Can Emotion Models Handle Them Both?,” in *Humaine Handbook*, vol. 32, pp. 845–863, Springer, 2005.

[70] A. Ortony, “On making believable emotional agents believable,” in *Emotions in humans and artifacts* (R. Trappl, P. Petta, and S. Payr, eds.), Emotions in Human and Artifacts, ch. 6, pp. 189–212, MIT Press, 2003.

[71] N. H. Frijda, “The laws of emotion,” *ACADEMY OF MANAGEMENT REVIEW*, vol. 32, no. 3, pp. 995–998, 2007.

[72] N. H. Frijda, “The Empirical Status of the Laws of Emotion,” *Cognition & Emotion*, vol. 6, pp. 467–477, Nov. 1992.

[73] T. Dalgleish, “The emotional brain,” *Nature Reviews Neuroscience*, 2004.

[74] P. Ekman, “An argument for basic emotions,” *Cognition and Emotion*, vol. 6(3/4), pp. 169–200, 1992.

[75] A. R. Damasio, T. J. Grabowski, A. Bechara, H. Damasio, L. L. B. Ponto, J. Parvizi, and R. D. Hichwa, “Subcortical and cortical brain activity during the feeling of self-generated emotions,” *Nature Neuroscience*, vol. 3, no. 10, pp. 1049–1056, 2000.

[76] J. E. Ledoux, “Cognitive-emotional interactions in the brain,” *Cognition & Emotion*, vol. 3, no. 4, pp. 267–289, 1989.

[77] H. Damasio, T. Grabowski, R. Frank, A. Galaburda, and A. Damasio, “The return of Phineas Gage: clues about the brain from the skull of a famous patient,” *Science*, vol. 264, pp. 1102–1105, May 1994.

[78] A. Olteanu, I. Simion, A. Purcăreanu, and N. Bîzdoacă, “Robotic Architecture for Experiments on Emotional Behavior,” *ace.ucv.ro*.

[79] M. B. Arnold, *Emotion and personality*, vol. 1. Columbia University Press, 1960.

[80] I. Roseman and C. Smith, “Appraisal Theory: Overview, Assumptions, Varieties, Controversies,” in *Appraisal Processes in Emotion Theory Methods Research* (K. R. Scherer, A. Schorr, and T. Johnstone, eds.), ch. 1, pp. 3–19, Oxford University Press, 2001.

[81] B. Parkinson, “Putting appraisal in context,” in *Appraisal Processes in Emotion Theory Methods Research* (K. R. Scherer, A. Schorr, and T. Johnstone, eds.), ch. 9, pp. 173–186, Oxford University Press, 2001.

[82] K. R. Scherer, “The Nature and Study of Appraisal. A Review of the Issues,” in *Appraisal Processes in Emotion Theory Methods Research* (K. R. Scherer, A. Schorr, and T. Johnstone, eds.), ch. 21, pp. 369–391, Oxford University Press, 2001.

[83] A. Ortony, *Emotions in Humans and Artifacts*, ch. On making Believable Emotional Agents Believable, pp. 188–211. MIT Press, 2003.

[84] A. Sloman, “Architectural Requirements for Human-like Agents Both Natural and Artificial. (What sorts of machines can love?),” in *Science* (K. Dautenhahn, ed.), ch. 7, pp. 163–195, John Benjamins, 2000.

[85] A. Bechara, H. Damasio, D. Tranel, and A. R. Damasio, “Deciding advantageously before knowing the advantageous strategy.,” *Science*, vol. 275, no. 5304, pp. 1293–1295, 1997.

[86] M. Davis, “The role of the amygdala in fear and anxiety.,” *Annual review of neuroscience*, vol. 15, no. Table 1, pp. 353–375, 1992.

[87] J. LeDoux, “The emotional brain, fear, and the amygdala,” *Cellular and molecular neurobiology*, 2003.

[88] H. Kluver and P. Bucy, “Preliminary analysis of functions of the temporal lobes in monkeys,” *Archives of Neurology & Psychiatry*, vol. 42, no. 6, p. 979, 1939.

[89] A. R. Damasio, *The Feeling of What Happens*. Harcourt Brace, 1999.

[90] C. Bartneck and J. Forlizzi, “A design-centred framework for social human-robot interaction,” in *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No.04TH8759)*, vol. Kurashiki, pp. 591–594, IEEE, IEEE, 2004.

[91] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu, “Robovie: an interactive humanoid robot,” *Industrial Robot: An International Journal*, vol. 28, no. 6, pp. 498–504, 2001.

[92] T. Kanda, H. Ishiguro, T. Ono, M. Imai, and K. Mase, “Multi-robot cooperation for human-robot communication,” in *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*, pp. 271–276, Ieee, 2002.

[93] R. Matsumura, “Communication Robot Robovie-mR2,” 2009.

[94] C. Breazeal, A. Brooks, D. Chilongo, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, and A. Lockerd, “Working collaboratively with humanoid robots,” in *Framework*, vol. 1, pp. 253–272, Ieee, 2004.

[95] C. Breazeal, M. Siegel, and M. Berlin, “Mobile, dexterous, social robots for mobile manipulation and human-robot interaction,” *SIGGRAPH’08: ACM SIGGRAPH 2008 new tech demos*, 2008.

[96] M. Fujita, “AIBO: Toward the Era of Digital Creatures,” *The International Journal of Robotics Research*, vol. 20, no. 10, pp. 781–794, 2001.

[97] L. Geppert, “QRIO the robot that could,” *Ieee Spectrum*, vol. 41, no. 5, pp. 34–37, 2004.

[98] Sony, “Sony History - Robots.”

[99] E. Libin, “Exploring the potentials of robotic psychology and robotherapy,” *Annual Review of CyberTherapy and Telemedicine*, vol. 1, 2003.

[100] Omron Corporation, “"Is this a real cat?" A robot cat you can bond with like a real pet – NeCoRo is born,” 2001.

[101] K. Wada and T. Shibata, “Social Effects of Robot Therapy in a Care House - Change of Social Network of the Residents for Two Months,” *Systems Research*, vol. 13, no. April, pp. 10–14, 2007.

[102] K. Wada and T. Shibata, “Robot Therapy in a Care House - Its Sociopsychological and Physiological Effects on the Residents,” *Science And Technology*, no. May, pp. 3966–3971, 2006.

[103] K. Wada and T. Shibata, “Living With Seal Robot-Its Sociopsychological and Physiological Influences on the Elderly at a Care House,” 2007.

[104] A. Billard, “Robota: Clever toy and educational tool,” *Robotics and Autonomous Systems*, vol. 42, pp. 259–269, Mar. 2003.

[105] B. Robins, K. Dautenhahn, R. Boekhorst, and A. Billard, “Effects of repeated exposure to a humanoid robot on children with autism,” *Assistive Technology*, vol. 22, no. March, pp. 22–24, 2004.

[106] K. Dautenhahn, C. L. Nehaniv, M. L. Walters, B. Robins, H. Kose-Bagci, N. A. Mirza, and M. Blow, “KASPAR – a minimally expressive humanoid robot for human–robot interaction research,” *Applied Bionics and Biomechanics*, vol. 6, pp. 369–397, Dec. 2009.

[107] J. Wainer, K. Dautenhahn, B. Robins, and F. Amirabdollahian, “Collaborating with Kaspar: Using an autonomous humanoid robot to foster cooperative dyadic play among children with autism,” 2010.

[108] H. Kozima, C. Nakagawa, and H. Yano, “Can a robot empathize with people?,” *Artificial Life and Robotics*, vol. 8, pp. 83–88, Sept. 2004.

[109] H. Kozima, M. P. Michalowski, and C. Nakagawa, “Keepon,” *International Journal of Social Robotics*, vol. 1, pp. 3–18, Nov. 2008.

[110] H. Kozima and C. Nakagawa, “Interactive Robots as Facilitators of Children ’ s Social Development,” *Social Development*, no. December, pp. 269–286, 2006.

[111] M. P. Michalowski, S. Sabanovic, and H. Kozima, “A dancing robot for rhythmic social interaction,” *Proceeding of the ACMIEEE international conference on Humanrobot interaction HRI 07*, p. 89, 2007.

[112] A. van Breemen, X. Yan, and B. Meerbeek, “iCat: an animated user-interface robot with personality,” in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems - AAMAS ’05*, (New York, New York, USA), p. 143, ACM Press, July 2005.

[113] K. P. Tee, R. Yan, Y. Chua, and Z. Huang, “Singularity-robust modular inverse kinematics for robotic gesture imitation,” in *2010 IEEE International Conference on Robotics and Biomimetics*, pp. 920–925, IEEE, Dec. 2010.

[114] A. Robotics, “AiSoy1.”

[115] B. Graf, U. Reiser, M. Hägele, K. Mauz, and P. Klein, “Robotic Home Assistant Care-O-bot ® 3 - Product Vision and Innovation Platform,” *Components*, pp. 312–320, 2008.

[116] K. Ogawa, S. Nishio, K. Koda, K. Taura, T. Minato, C. T. Ishii, and H. Ishiguro, “Telenoid: Tele-presence android for communication,” in *ACM SIGGRAPH 2011 Emerging Technologies on - SIGGRAPH ’11*, (New York, New York, USA), pp. 1–1, ACM Press, Aug. 2011.

[117] V. RED’KOa and A. KOVAL, “Evolutionary Approach to Investigations of Cognitive Systems,” *bicasociety.org*, 2011.

[118] J. Hirth, N. Schmitz, and K. Berns, “Towards Social Robots: Designing an Emotion-Based Architecture,” *International Journal of Social Robotics*, 2011.

[119] A. Neto and F. da Silva, “A Computer Architecture for Intelligent Agents with Personality and Emotions,” *Human-Computer Interaction: The Agency Perspective*, vol. 396, pp. 263—-285, 2012.

[120] A. Matsuda, H. Misawa, and K. Horio, “Decision making based on reinforcement learning and emotion learning for social behavior,” 2011.

[121] J. D. Velásquez and P. Maes, “Cathexis: a computational model of emotions,” in *Proceedings of the first international conference on Autonomous agents - AGENTS ’97*, (New York, New York, USA), pp. 518–519, ACM Press, Feb. 1997.

[122] J. D. Velásquez, M. Fujita, and H. Kitano, “An Open Architecture for Emotion and Behavior Control of Autonomous Agents,” in *Proceedings of the second international conference on Autonomous agents*, pp. 473–474, ACM, 1998.

[123] J. D. Velásquez, “Modeling Emotions and Other Motivations in Synthetic Agents,” *Artificial Intelligence*, pp. 10–15, 1997.

[124] L. Cañamero, “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.*, 1997.

[125] O. Avila-García and L. Cañamero, “Comparing a Voting-Based Policy with Winner-Takes-All to Perform Action Selection in Motivational Agents,” pp. 855–864, Nov. 2002.

[126] D. Cañamero, “Designing Emotions for Activity Selection,” 2000.

[127] L. Cañamero, “Emotion understanding from the perspective of autonomous robots research,” *Neural Networks*, vol. 18, pp. 445–455, 2005.

[128] S. C. Gadanho and J. Hallam, “Exploring the Role of Emotions Autonomous Robot Learning,” in *IN PROCEEDINGS OF THE AAAI FALL SYMPOSIUM ON EMOTIONAL INTELLIGENCE*, no. Wilson 91, pp. 84—-89, AAAI Press, 1998.

[129] S. Gadanho and J. Hallam, “Emotion- triggered learning in autonomous robot control,” *Cybernetics and Systems*, vol. 32(5), pp. 531–59, 2001.

[130] S. C. Gadanho and L. Custódio, “Asynchronous learning by emotions and cognition,” in *In Proceedings of the Seventh International Conference on the Simulation of Adaptive Behavior (SAB2002)*, 2002.

[131] S. Gadanho and L. Custodio, “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*, 2002.

[132] A. R. Damasio, *Descartes’ Error: Emotion, Reason, and the Human Brain*. Harper Perennial, 1995.

[133] C. Breazeal and L. Aryananda, “Recognition of affective communicative intent in robot- directed speech,” *Autonomous Robots*, vol. 12, pp. 83–104, 2002.

[134] C. Breazeal and R. Brooks, *Who Needs Emotions: The Brain Meets the Robot*, ch. Robot Emotion: A Functional Perspective. MIT Press, 2004.

[135] C. Breazeal, D. Buchsbaum, J. Gray, D. Gatenby, and B. Blumberg, “Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots,” *Artificial Life*, vol. 11, pp. 1–32, 2005.

[136] B. M. Blumberg, P. M. Todd, and P. Maes, “No Bad Dogs: Ethological Lessons for Learning in Hamsterdam,” in *Collection*, vol. 01463, pp. 295–304, MIT Press, 1996.

[137] R. R. Murphy, R. R. Murphy, C. Lisetti, R. Tardif, L. Irish, and A. Gage, “Emotion-Based Control of Cooperating Heterogeneous Mobile Robots,” *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, vol. 18, pp. 744 – 757, 2002.

[138] G. A. Hollinger, Y. Georgiev, A. Manfredi, B. A. Maxwell, Z. A. Pezzementi, and B. Mitchell, “Design of a Social Mobile Robot Using Emotion-Based Decision Mechanisms,” 2006.

[139] C. L. Lisetti and A. Marpaung, *KI 2006: Advances in Artificial Intelligence*, ch. Affective Cognitive Modeling for Autonomous Agents Based on Scherer’s Emotion Theory, pp. 19–32. 2007.

[140] W. P. Lee, J. W. Kuo, and P. C. Lai, “Building Adaptive Emotion-Based Pet Robots,” in *Proceedings of the World Congress on Engineering* (S. I. Ao, L. Gelman, D. W. Hukins, A. Hunter, and A. M. Korsunsky, eds.), vol. I, pp. 85–90, Newswood Limited, 2008.

[141] S. B. Nair, W. W. Godfrey, and D. H. Kim, “On Realizing a Multi-Agent Emotion Engine,” *International Journal of Synthetic Emotions*, vol. 2, pp. 1–27, June 2011.

[142] R. C. Arkin, P. Ulam, and A. R. Wagner, “Moral Decision-making in Autonomous Systems: Enforcement, Moral Emotions, Dignity, Trust and Deception,”

[143] J. Haidt, “The Moral Emotions,” in *Handbook of Affective Sciences* (R. Davidson, ed.), Oxford University Press, 2003.

[144] G. Hollinger, Y. Georgiev, A. Manfredi, B. Maxwell, Z. Pezzementi, and B. Mitchell, “Design of a social mobile robot using emotion-based decision mechanisms,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 3093–3098, 2007.

[145] R. S. Lazarus, *Appraisal processes in emotion: Theory, methods, research*, ch. Relational meaning and discrete emotions, pp. 37–67. New York: Oxford University Press, 2001.

[146] J. S. de Freitas and J. Queiroz, *Advances in Artificial Life*, ch. Artificial Emotions: Are We Ready for Them?, pp. 223–232. 2007.

[147] D. Cañamero, “Emotions and adaptation in autonomous agents: a design perspective,” *Cybernetics and systems: International Journal*, vol. 32, pp. 507–529, 2001.

[148] D. R. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid*. Penguin Philosophy, Basic Books, 1979.

[149] L. Cañamero and P. Gaussier, “Emotion understanding: robots as tools and models,” *Emotional Development*, pp. 235—-258, 2005.

[150] M. Scheutz, “Useful roles of emotions in artificial agents: a case study from artificial life.,” in *AAAI 2004*, pp. 42–48, AAAI press, Menlo Park, 2004.

[151] A. Ortony, D. A. Norman, and W. Revelle, *J.M. Fellous and M.A. Arbib, Who needs emotions: The brain meets the machine*, ch. Affect and proto-affect in effective functioning. 2005.

[152] J. Fellows, “From human emotions to robot emotions,” tech. rep., AAAI 2004 Spring Symposium on Architectures for Modelling Emotion: Cross- Disciplinary Foundations.SS-04-02. AAAI Press., 2004.

[153] A. Kelley, *Who Needs Emotions? The Brain Meets the Robot*, ch. Neurochemical networks encoding emotion and motivation: an evolutionary perspective. Oxford University Press, 2005.

[154] R. W. Picard, *Emotions in Humans and Artifacts*, ch. What does it mean for a computer to have emotions? MIT Press, 2003.

[155] M. Minsky, *The Society of Mind*. Simon and Schuster, 1986.

[156] N. Alvarado, S. Adams, and S. Burbeck, “The role of emotion in an architecture of mind,” *IBM Research*, 2002.

[157] M. Malfaz and M. Salichs, “Learning behaviour-selection algorithms for autonomous social agents living in a role-playing game,” in *Proccedings of the AISB’06: Adaptation in Artificial and Biological Systems. University of Bristol, Bristol, England*, April 2006.

[158] M. Malfaz and M. Salichs, “Learning to deal with objects,” in *Proccedings of the 8th International Conference on Development and Learning (ICDL 2009)*, 2009.

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

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

[161] O. Ávila García and L. Cañamero, “Using hormonal feedback to modulate action selection in a competitive scenario,” in *Procceding of the 8th International Conference on Simulation of Adaptive Behavior (SAB’04)*, 2004.

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

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

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

[165] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[166] C. Isbell, C. R. Shelton, M. Kearns, S. Singh, and P. Stone, “A social reinforcement learning agent,” in *the fifth international conference on Autonomous agents, Montreal, Quebec, Canada*, 2001.

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

[168] B. Bakker, V. Zhumatiy, G. Gruener, and J. Schmidhuber, “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*, 2003.

[169] C. H. C. Ribeiro, R. Pegoraro, and A. H. RealiCosta, “Experience generalization for concurrent reinforcement learners: the minimax-qs algorithm,” in *AAMAS 2002*, 2002.

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

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

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

[173] C. Touzet, *The Handbook of Brain Theory and Neural Networks*, ch. Q-learning for robots, pp. 934–937. MIT Press, 2003.

[174] S. Mahadevan and J. Connell, “Automatic programming of behavior-based robots using reinforcement learning,” *Artificial intelligence*, 1992.

[175] E. Martinson, A. Stoytchev, and R. C. Arkin, “Robot behavioral selection using q-learning,” 2001.

[176] A. Castro-González, F. Amirabdollahian, D. Polani, M. Malfaz, and M. A. Salichs, “Robot self-preservation and adaptation to user preferences in game play, a preliminary study,” in *2011 IEEE International Conference on Robotics and Biomimetics*, pp. 2491–2498, IEEE, Dec. 2011.

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

[178] J. A. Starzyk, “Motivated Learning for Computational Intelligence,” *Computational Modeling and Simulation of Intellect: Current State and Future Perspectives*, 2010.

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

[180] A. Olteanu, I. Simion, A. Purcăreanu, and N. Bîzdoacă, “Robotic architecture for experiments on emotional behavior,” *The Annals of Craiova University. Series: Automation, Computers, Electronics, Mechatronics*, vol. 8, no. 2, pp. 36–43, 2011.

[181] M. A. Salichs, R.Barber, A. M.Khamis, M.Malfaz, J. F.Gorostiza, R.Pacheco, R.Rivas, A.Corrales, and E.Delgado, “Maggie: A robotic platform for human-robot social interaction,” in *IEEE International Conference on Robotics, Automation and Mechatronics (RAM 2006). Bangkok. Thailand*, 2006.

[182] R. Barber and M. Salichs, “A new human based architecture for intelligent autonomous robots,” in *Proceedings of The 4th IFAC Symposium on Intelligent Autonomous Vehicles*, pp. 85–90, Elsevier, 2002.

[183] R. Barber, *Desarrollo de una Arquitectura para Robots Moviles Autonomos. Aplicacion a un Sistema de Navegacion Topologica*. PhD thesis, Universidad Carlos III de Madrid, 2000.

[184] R. Barber and M. A. Salichs, “Mobile robot navigation based on event maps,” in *International Conference on Field and Service Robotics*, pp. 61–66, June 2001.

[185] R. Rivas, A. Corrales, R. Barber, and M. A. Salichs, “Robot skill abstraction for ad architecture,” in *6th IFAC Symposium on Intelligent Autonomous Vehicles*, 2007.

[186] M. Malfaz, A. Castro-González, R. Barber, and M. A. Salichs, “A Biologically Inspired Architecture for an Autonomous and Social Robot,” *Autonomous Mental Development, IEEE Transactions on*, vol. 3, no. 3, pp. 232–246, 2011.

[187] R. M. Shiffrin, “Atention,” *Stevens’ Handbook of Experimental Psychology. Second Edition.*, vol. 2, 1988.

[188] R. M. Shiffrin and W. Schneider, “Controlled and automatic human information processing: In perceptual learning, automatic attending and a general theory,” *Psicological Review*, pp. 127–190, 1997.

[189] R. Rivas, A. Corrales, R. Barber, and M. A. Salichs, “Robot skill abstraction for ad architecture,” in *6th IFAC Symposium on Intelligent Autonomous Vehicles*, 2007.

[190] M. A. Salichs and R. Barber, “A new human based architecture for intelligent autonomous robots.,” in *4th IFAC Symposium on Intelligent Autonomous Vehicles*, pp. 85–90, 2001.

[191] M. J. L. Boada, R. Barber, and M. A. Salichs, “Visual approach skill for a mobile robot using learning and fusion of simple skills,” *Robotics and Autonomous Systems*, vol. 38, pp. 157–70, March 2002.

[192] E. N. Zalta, *Stanford Encyclopedia of Philosophy*. http://plato.stanford.edu/entries/memory/, First published Tue Mar 11, 2003; substantive revision Wed Feb 3, 2010.

[193] R. C. Atkinson and R. M. Shiffrin, *The Psychology of Learning and Motivation*, vol. 2, ch. Human Memory: A Proposed System and Its Control Processes, pp. 89–195. K. W. Spence and J. T. Spence. New York: Academic Press, 1968.

[194] J. J. Bryson and E. Tanguy, “Simplifying the design of human-like behaviour: Emotions as durative dynamic state for action selection,” *International Journal of Synthetic Emotions*, pp. 355–377, 2009.

[195] J. Salichs, A. Castro-Gonzalez, and M. A. Salichs, “Infrared remote control with a social robot,” in *FIRA RoboWorld Congress 2009* (Springer, ed.), (Incheon, Korea.), Springer, August 2009.

[196] A. d. V. Corrales Paredes, *Sistema de Navegación para Robots Sociales Basado en Señales*. PhD thesis, Universidad Carlos III de Madrid, 2012.

[197] S. C. Gadanho and J. Hallam, “Robot learning driven by emotions,” *Adaptive Behavior*, 2001.

[198] S. C. Gadanho and J. Hallam, “Emotion-triggered learning in autonomous robot control,” *Cybernetics & Systems*, vol. 32, pp. 531—-559, 2001.

[199] C. Vigorito and A. Barto, “Intrinsically motivated hierarchical skill learning in structured environment,” *IEEE Transaction on Autonomous Mental Developmen. Special Issue on Active Learning and Intrinsically Motivated Exploration in Robots*, vol. 2(2), pp. 132–143, 2010.

[200] J. Boyan and A. Moore, “Generalization in reinforcement learning: Safely approximating the value function,” in *Advances in Neural Information Processing Systems 7*, pp. 369–376, MIT Press, 1995.

[201] N. Sprague and D. Ballard, “Multiple-goal reinforcement learning with modular sarsa(0),” in *the 18th International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico*, 2003.

[202] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman, “Efficient solution algorithms for factored mdps,” *Journal of Artificial Intelligence research (JAIR)*, vol. 19, pp. 399–468, 2003.

[203] L. Li, T. Walsh, and M. Littman, “Towards a unified theory of state abstraction for mdp,” in *Ninth International Symposium on Artificial Intelligence and Mathematics*, pp. 531–539, 2006.

[204] C. Boutilier, R. Dearden, and M. Goldszmidt, “Stochastic dynamic programming with factored representation,” . *Artificial Intelligence*, vol. 121 (1-2), pp. 49–107, 2000.

[205] R. Givan, T. Dean, and M. Greig, “Equivalence notions and model minimization in markov decision processes,” *Artificial Intelligence*, vol. 147(1-2), pp. 163–223, 2003.

[206] A. Callum, *Reinforcement Learning with Selective Perception and Hidden State*. PhD thesis, University of Rochester, Rochester, NY, 1995.

[207] M. Malfaz and M. Salichs, “Using muds as an experimental platform for testing a decision making system for self-motivated autonomous agents,” *Artificial Intelligence and Simulation of Behaviour Journal*, vol. 2(1), no. 1, pp. 21–44, 2010.

[208] M. Malfaz and M. A. Salichs, “Learning To Avoid Risky Actions,” *Cybernetics and Systems*, vol. 42, pp. 636–658, Nov. 2011.