Chapter 1

Introduction

He turned once more to the robot. “Get up!”
The robot towered upward slowly and Donovan’s head craned and his puckered lips whistled.
Powell said: “Can you go out upon the surface? In the light?”
There was consideration while the robot’s slow brain worked. Then, “Yes, Master.”
“Good. Do you know what a mile is?”
Another consideration, and another slow answer. “Yes, Master.”
“We will take you up to the surface then, and indicate a direction. You will go about seventeen miles, and somewhere in that general region you will meet another robot, smaller than yourself. You understand so far?”
“Yes, Master.”
“You will find this robot and order him to return. If he does not wish to, you are to bring him back by force.”

Extracted from the short story Runaround,
in Asimov’s I, Robot *[7]*.

“Now on four occasions recently,” Powell said, “your boss deviated from brain-scheme. Do you remember those occasions?”[...]
Powell turned back to the robot, “What were you doing each time... I mean the whole group.”[...]
He said [the robot], “The first time we were at work on a difficult outcropping in Tunnel 17, Level B. The second time we were buttressing the roof against a possible cave-in. The third time we were preparing accurate blasts in order to tunnel farther without breaking into a subterranean fissure. The fourth time was just after a minor cave-in.”
“What happened at these times?”
“It is difficult to describe. An order would be issued, but before we could receive and interpret it, a new order came to march in queer formation.”

Extracted from the short story Catch that Rabbit,
in Asimov’s I, Robot *[7]*.

Perhaps the excerpts shown previously describe simple and irrelevant scenes that usually would not attract much of our attention. Moreover, probably the only curious event that would even draw a slight smile in the reader's face is the fact that a human communicates with a robot through natural language, i.e. talking, while the robot not only understands the conversation, but also replies in the same way. However, this is not our focus of interest, although it is in indeed a very challenging task that researchers in artificial intelligence (AI) are still working on.

The short story from where the first text is extracted (Runaround) takes place in Mercury and is about a robot, SPD-13 ("Speedy"), that is sent to bring selenium from the nearest selenium pool, 17 miles away from the base station. However, after five hours of having departed, Speedy has not returned yet. Therefore, Donovan and Powell decide to send another robot to get him and to analyze what happened. At this point, the conversation shown in the text takes place. Powell orders the robot what seems a simple task. Although the story actually continues without the robot having to execute the task, we are interested in analyzing the consequences of this "simple" task. Let us first assume that somehow the robot understands what the task commanded by Powell is about. Some of the abilities the robot must have to perform this "simple" task are: ability to **perceive** its environment in order to create its internal world model; ability to build a map (if the robot does not have it a priori) of the environment and **localize** itself and the place it has to go within the map; ability to **plan** a route to the goal location and then to come back; ability to **navigate** through the environment, probably avoiding obstacles, conflicting paths, etc.; ability to **recognize** another robot; ability to **decide** how to perform the task, i.e. which actions to execute; ability to **react** and **recover** upon possible conflicts it could encounter during the execution of the task. Hence, what seemed a simple task turned out to be a more complex one, requiring a set of abilities where each of them leads to a broad range of challenging problems that different fields in AI have addressed since their origins in the 1950's.

Besides the above mentioned abilities we expect a robot to be programmed with, we can find a last interesting component within the second text. In this occasion (Catch that Rabbit), Powell and Donovan have to discover why the robot DV-5 (Dave) fails executing the task it is commanded to perform. The robot's peculiarity is that it is a robot with six subsidiary robots which are controlled through positronic fields. In other words, Dave can be seen as a coordinator robot with six "worker" robots under its responsability which perform the tasks commanded by the coordinator. Back to the story, Powell and Donovan decide to spy while the robots work to discover why the task is not being correctly fulfilled. They discover that in general the robots are working the right way until something unexpected happens and they start marching and dancing leaving the task aside. To understand why they behave that way, Powell asks one of the subsidiary robots what is going on (extracted text). At this point the robot relates the tasks they were assigned to do. Two ingredients in this story draw our attention. First, the story is related to a **team of robots**, and second, the tasks to perform cannot be accomplished individually, but through **teamwork**. In order to fulfill the tasks, a coordinator is in charge of the team, sending the commands to the team and supervising the task execution. When dealing with **cooperative tasks** where a group of robots have to achieve a joint task, some of the challenges, besides the ones mentioned before of course, consist in answering the following questions: who decides what to do?, i.e. a single robot decides which actions to perform (centralized system) or all robots discuss the selected actions (distributed system); who does what?, i.e. one robot is selected to perform the complete task (single task execution) or each robot may perform part of the task or subtasks (distributed task execution); who monitors the task execution?, i.e. one robot receives all the information from the rest of the robots and decides by itself (single monitoring) or each robot has its own beliefs of the world and reacts accordingly (distributed monitoring); and finally, which **coordination mechanism** to employ in order to synchronize the robots' actions?

Thus, from what seemed to be two independent excerpts of not much interest, we have remarked a set of problems that probably, from our human point of view, have obvious solutions (we face them in our daily life without even noticing their difficulty). However, from an AI researcher perspective, when designing the robot behavior, these problems are not trivial at all, and in fact, result in big challenges for AI nowadays.

1.1 Motivation and Overview

The dissertation presented in this work is addressed to two of the presented challenges. First, the decision-making for the action selection problem and second, the incorporation of a coordination mechanism to achieve cooperative tasks within a multi-robot system. We next overview the main problems which we have to deal with and how we propose to solve them.

An important aspect to consider when designing the reasoning engine is the type of environment where the robot performs its task. The difficulties that arise within deterministic environments (controlled environments) are far much easier to deal with that when dealing with stochastic environments (uncontrolled). Clearly the latter is much more interesting, and is the one this dissertation addresses. In such environments, where the world continuously changes out of our control, the reasoning engine must include mechanisms to overcome imprecision and dynamism of the environment. More precisely, it has to be able to react and recover from unexpected situations that may occur during the performance of the task where a real-time response is fundamental.

The behavior of a robot results from the execution of actions for different states, if we define acting as the execution of a policy $\pi : s \rightarrow a$ (where s is the current state and a, the action to execute in the given state) [44]. Defining each possible state and the actions to perform at each state, i.e. defining the policy, is challenging and tedious to be done completely manually. This policy is one of the fundamental parts of the robot's reasoning engine. Therefore, it is crucial to find a way for automatically and efficiently acquiring it. As we review further on, several machine learning techniques have been proposed during the past years.

Besides the difficulties emerged due to the nature of the environment, we must also take into account the limitations of the robot performing the task. Thus, the uncertainty of the robot's internal beliefs of the world depends on the accuracy of the robot's sensors. The reasoning engine must be able to handle uncertainty so the behavior of the robot does not result degraded. A last important aspect to consider are the robot's computational resources. The processor determines the type of algorithms (in terms of complexity) that the reasoning engine may use.

From a multi-agent perspective, the problem we address in this work is related to cooperation or collaboration[1] among agents. Collaboration is desired in several domains where a group of robots (also seen as agents) work together to achieve a common goal. It is not only important to have the agents collaborate, but also to do it in a coordinated manner so the task can be organized to obtain effective results. Providing the agents with capacities to collaborate and to coordinate is complex, as it is not just a matter of dividing the tasks and assigning roles to each agent. Instead, it is also a matter of beliefs and commitments of all robots to fulfill a common task. Drogoul and Collinot [14] distinguish three levels of behaviors when designing a multi-agent system:

- elementary behaviors, actions or functions that the agents individually perform (what to do);
- relational behaviors, how agents interact with other agents and the influences of their elementary actions on the other agents (what to do in presence of other agents); and,
- organizational behaviors, how the agents can manage their interactions to stay organized (what to do with these agents).

Similarly, Grosz and Kraus [19] argue that collaborating agents must

- establish mutual beliefs on what actions they will perform to complete the task (relational level);
- agree on who does what (organizational level); and,
- establish mutual beliefs of their individual intentions to act (relational level).

Communication among agents is essential to achieve these requirements.

The robot soccer domain is a very challenging test-bed that incorporates most of the problems enumerated so far. Hence, we deal with a highly dynamic environment that requires real-time response. Robots' sensors are not very accurate and therefore, we must model uncertainty within the reasoning engine to act accordingly. Robots' actions performances are imprecise and recovery mechanisms should be considered. Finally, computational resources are very limited, and thus, simple processes have to be taken into account. In this dissertation we contribute with an approach for action selection and coordination for joint multi-robot tasks. More precisely, we apply Case-Based Reasoning (CBR) techniques to model the reasoning engine and its application in the robot soccer domain. Case-based reasoning is an approach to problem solving that emphasizes the role of prior experience during future problems solving [39]. It has been inspired by the way humans reason and use their memory of previous experiences to solve new problems. An example directly related with the work presented in this dissertation can be found in team sports. During training the coach studies with the players different game situations and the according movements (gameplays) that the players

[1] Through the dissertation we will refer to both concepts, cooperation or collaboration, as synonyms, although the latter can also be related to "working with the enemy", including a traitorous sense.

should perform. The *playbook* corresponds to the case base in the CBR system. During a game, when the players detect a similar configuration between the current situation in the field and the ones in the playbook (CBR's retrieval step), they automatically reproduce the gameplays reviewed during the training, performing certain adaptations if necessary (CBR's reuse step).

The approach models the state of the game at a given time as a problem description, which is compared to a set of predefined situations, called cases. A case describes the state of the environment (problem description) and the actions to perform in that state (solution description). The retrieval process consists in obtaining the most similar case. Next, the solution of the retrieved case is reused after some adaptation process, as required. We model the case solution as a set of sequences of actions, *gameplays*, which indicate what actions should each robot perform. Finally, we specify a multi-robot architecture and a coordination mechanism based on messages exchanged among the robots in order to achieve a cooperative behavior.

Why Case-Based Reasoning?

The first question that can arise when reading this work is *why CBR?* As we review in Chapter 2, different approaches to solve the action selection problem have been presented through the past years (Reinforcement Learning, Fuzzy Theory, Decision Trees, Neural Networks, etc.) obtaining successful results. However, we believe that Case-Based Reasoning integrates fundamental properties that not only help the designer in building a reasoning engine, but also result very intuitive for humans since it is tightly related to the way humans reason.

From the implementation point of view, we can classify the design of a robot behavior from a procedural implementation, where the behavior is composed of a sequence of subroutines and evaluating conditions (low level approach), to a model-based implementation, where the knowledge representation is done through state-action models and the task to learn is the mapping function between states and actions (high level approach). Although the high level approaches have several advantages over procedural implementations, low level approaches are still being widely used for designing robot behaviors, specially in very specific scenarios or in competition in the case of robot soccer (RoboCup). A common approach used within RoboCup to describe behaviors are hierarchical finite state machines (FSMs) [60, 67, 53] to provide a certain degree of abstraction level. The advantage of using this approach is probably due to its high reactivity. The robot is able to rapidly switch from one behavior to another when required. However, programming individual or complex behaviors is still tedious and slow. As argued in [46, 57] changes are complicated due to interdependencies and large amount of parameters to consider when programming the behaviors. A minor modification in the code of a behavior can have a big impact on other behaviors. Another important drawback within reactive approaches is that from a strategic point of view we can classify them as "short-sighted", meaning that their decision-making is usually driven by a partial state of the environment where the actions take place, without having a broader view of the world state. Thus, an action can be suitable for a given moment in time, but probably another action would have been a better choice if the whole world state or possible future states could have been considered or predicted.

Regarding model-based approaches, we can classify them according to the policy *readability*, i.e. how understandable for a human reader the learned policy is. Techniques as reinforcement learning (RL), neural networks (NN) or evolutionary algorithms (EA) have proved to be useful in many domains, including action selection in robotics. However, their main drawback is that the learned policy cannot be manually followed by an expert, and thus, analyzing why a certain action has been selected is not feasible. As other researchers have previously remarked [18, 46, 31, 10, 38, 15], we believe that this is an important property to consider when choosing among the available approaches, specially within complex domains, where some kind of justification is necessary for evaluating the appropriateness of the selected actions. On the other hand, the advantage of other approaches such as decision trees, expert systems, fuzzy rules (rule-based approaches in general) or case-base reasoning (instance-based approaches) is that their knowledge representation is readable from the expert perspective, not only facilitating the comprehension of the policy, but also providing easy access to modify the current knowledge of the reasoning engine.

Another important component to consider is the time required for learning the policy and the amount of training data to achieve an acceptable accuracy level. From the above mentioned approaches, RL, NN, EA and decision trees either require a large amount of training data or time or both, which are usually not available within the robotics field.

Finally, and not less important, a on-line learning ability is desired for this kind of domains where the robot may encounter unexpected situations that where not considered during the design stage. With this last component, the adaptability of the robot's behavior is guaranteed, allowing it to acquire new knowledge as it performs the tasks. Rule-based approaches lack of this last component. The only way for introducing new knowledge is manually modifying the rule set. Other approaches, such as NN, EA or decision trees need to repeat the training process before using the new learned policy. Modifying the current knowledge of the system is time consuming and requires new training data.

After reviewing the desired properties of the approach used for the action selection problem, we conclude that Case-Based Reasoning (an instance-based approach) fulfills the requirements described. The case base contains the knowledge representation of the reasoning engine, which in fact, corresponds to a set of situations (cases) the robot encounters through the task execution. Each case may represent a complete or partial description of the state of the environment and the corresponding solution to that state, i.e. the actions to execute. Cases can either be generalized or specialized allowing the expert to gradually introduce knowledge as needed. The knowledge of the system is "transparent" and the expert can easily modify or insert new knowledge without spending time training the reasoning engine again.

Regarding cooperation and teamwork, several works have been presented so far, either using more formal methodologies as the *joint intention* theory introduced by Cohen and Levesque [11] in Tambe's *flexible team work* [62], or simpler mechanisms such as role assignment where cooperation usually results as an emerging property [57, 65, 74, 35, 15], or including explicit coordination mechanisms through communication to enforce commitment among the involved agents (request-acknowledge type) [66, 3, 18]. Interestingly, in our work the use of cases also allows us to easily model cooperative tasks. As mentioned before, in order to have agents performing joint tasks it is fundamental that: first, all agents agree on the task to perform; second, the implied agents commit to execute the task as planned; and third, these agents must be aware of the actions each of them performs to synchronize. In this work we specify a coordination mechanism that takes place during both the retrieval and the reuse steps based on messages exchanged among the robots about their internal states (beliefs and intentions). Hence, reviewing the requirements by Grosz and Kraus mentioned before, the combination of the case structure and the coordination mechanism we propose ensures that: the solution description indicates the actions the robots should perform (requirement *i*); the retrieval process allocates robots to actions (requirement *ii*); and finally, with the coordination mechanism, the robots share their individual intentions to act (requirement *iii*).

In conclusion, we believe that using CBR techniques is appropriate, not only due to the close relation with the way humans reason, but also because it provides a high level abstraction of the problem to solve through a modular methodology. This latter allows the expert to easily modify the robots' behavior as required, either introducing or replacing cases in the case base (knowledge of the system), defining new similarity functions, altering the retrieval process, etc. CBR is a very flexible and intuitive framework and thus, is suitable for the problem domain this dissertation is focused on.

1.2 Problem Domain

The problem domain where this thesis is applied to is robot soccer. RoboCup is a well known competition [69] whose ultimate goal is to develop a team of humanoid robots to play soccer against the human world champion team by 2050. Of course this is long term goal, but in fact, the main objective of designing this test-bed is to foster AI and robotics within a very complex domain and to motivate researchers of different fields to work together in order to achieve a common goal.

To this end RoboCup offers several leagues where, altough the goal is the same, the challenges differ. Currently we can find the following leagues: Simulation, Small size, Middle size, Standard Platform and Humanoid. The Standard Platform League is a new league that will start next year (2008) replacing the Four-Legged League[2]. Within this league all teams use the same robot, so the challenge is focused on developing the software for the robots, and not the physical robot contrarily to other leagues.

Within the The Four-Legged League teams consist of four Sony AIBO robots which operate fully autonomously, i.e. there is no external control, neither by humans nor by computers. Communication among robots of the same team is allowed through wireless or speakers and microphones (although the last ones are not usually used). There are two teams in a game: a red team and a blue team. The field dimensions are 6m long and 4m wide and represents a Cartesian plane as shown in Figure 1.1. There are two goals (cyan and yellow) and four colored markers the robots use to localize themselves on the field. A game consists of three parts, i. e. the first half, a half-time break, and the second half. Each half is 10 minutes. The teams change the goal defended and color

[2] The robots for this league were the AIBO robots from Sony. Since Sony stopped manufacturing the robots, the RoboCup organizers had to switch to another model, the humanoid Aldebaran Nao.

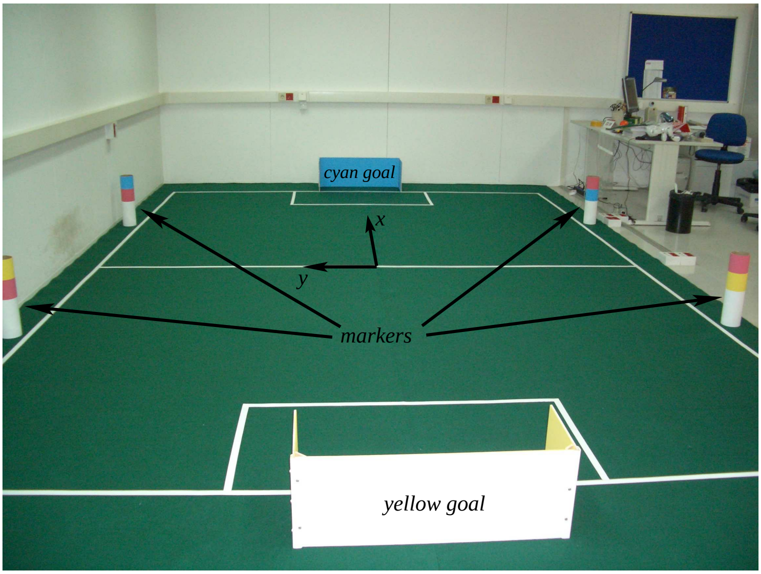

Figure 1.1: Snapshot of the Four-Legged League field (image extracted from the IIIA lab).

of the team markers during the half-time break. At any point of the game, if the score difference is greater than 10 points the game ends. There is also an external controller, the *GameController*, which sends messages to the robots in order to stop or resume the game after a goal, to notify penalized robots, to start or end the game, etc. For more details on the official rules of the game refer the RoboCup Four-Legged League Rule Book [12].

A Brief History of RoboCup

Extracted from the RoboCup Official website (Overview) [1].

In the history of artificial intelligence and robotics, the year 1997 will be remembered as a turning point. In May 1997, IBM Deep Blue defeated the human world champion in chess. Forty years of challenge in the AI community came to a successful conclusion. On July 4, 1997, NASA's pathfinder mission made a successful landing and the first autonomous robotics system, Sojourner, was deployed on the surface of Mars. Together with these accomplishments, RoboCup made its first steps toward the development of robotic soccer players which can beat a human World Cup champion team.

The idea of robots playing soccer was first mentioned by Professor Alan Mackworth (University of British Columbia, Canada) in a paper entitled "On Seeing Robots" presented at VI-92, 1992 and later published in a book *Computer Vision: System, Theory, and Applications*, pages 1-13, World Scientific Press, Singapore, 1993. A series of papers on the Dynamo robot soccer project was published by his group.

Independently, a group of Japanese researchers organized a Workshop on Grand Challenges in Artificial Intelligence in October, 1992 in Tokyo, discussing possible grand challenge problems. This workshop led to a serious discussions of using the game of soccer for promoting science and technology. A series of investigations were carried out, including a technology feasibility study, a social impact assessment, and a financial feasibility study. In addition, rules were drafted, as well as prototype development of soccer robots and simulator systems. As a result of these studies, they concluded that the project is feasible and desirable. In June 1993, a group of researchers, including Minoru Asada, Yasuo Kuniyoshi, and Hiroaki Kitano, decided to launch a robotic competition, tentatively named the Robot J-League (J-League is the name of the newly established Japanese Professional soccer league). Within a month, however, they received overwhelming reactions from researchers outside of Japan, requesting that the initiative be extended as an international joint project. Accordingly, they renamed the project as the Robot World Cup Initiative, "RoboCup" for short.

Concurrent to this discussion, several researchers were already using the game of soccer as a domain for their research. For example, Itsuki Noda, at ElectroTechnical Laboratory (ETL), a government research center in Japan, was conducting multi-agent research using soccer, and started the development of a dedicated simulator for soccer games. This simulator later became the official soccer server of RoboCup. Independently, Professor Minoru Asada's Lab. at Osaka University, and Professor Manuela Veloso and her student Peter Stone at Carnegie Mellon University had been working on soccer playing robots. Without the participation of these early pioneers of the field, RoboCup could not have taken off.

In September 1993, the first public announcement of the initiative was made, and specific regulations were drafted. Accordingly, discussions on organizations and technical issues were held at numerous conferences and workshops, including AAAI-94, JSAI Symposium, and at various robotics society meetings.

Meanwhile, Noda's team at ETL announced the Soccer Server version 0 (LISP version), the first open system simulator for the soccer domain enabling multi-agent systems research, followed by version 1.0 of Soccer Server (C++ Version) which was distributed via the web. The first public demonstration of this simulator was made at IJCAI-95.

During the International Joint Conference on Artificial Intelligence (IJCAI-95) held at Montreal, Canada, August, 1995, the announcement was made to organize the First Robot World Cup Soccer Games and Conferences in conjunction with IJCAI-97 Nagoya. At the same time, the decision was made to organize Pre-RoboCup-96, in order to identify potential problems associated with organizing RoboCup at a large scale. The decision was made to provide two years of preparation and development time, so that initial groups of researchers could start robot and simulation team development, as well as giving lead time for their funding schedules.

Pre-RoboCup-96 was held during the International Conference on Intelligence Robotics and Systems (IROS-96), Osaka, from November 4–8, 1996, with eight teams competing in a simulation league and demonstration of real robot for the middle size league. While limited in scale, this competition was the first competition using soccer games for promotion of research and education.

The first official RoboCup games and conference was held in 1997 with great success. Over 40 teams participated (real and simulation combined), and over 5,000 spectators attended.

1.3 Contributions

This thesis contributes a novel case-based approach for action selection and coordination in joint multi-robot tasks. This approach is applied and evaluated in the representative domain of robot soccer.

The main characteristics of the approach can be summarized as follows:

- The case definition corresponds to a complete description of the environment, including the actions to perform by a team of robots and general domain knowledge to handle uncertainty in the incoming information from perception.

- Two types of features are introduced: controllable and non-controlable features. The former ones are related to those features whose values can be directly modified in order to increase the similarity between the evaluated case and the current problem; while the latter ones, correspond to those features that the system cannot modify.

- The retrieval step is composed of three measures: the aggregation of domain-dependent similarity measures; the cost of adapting the current problem to a case; and the applicability evaluation of a case combining domain knowledge rules and similarity measures. The retrieval step applies a filtering mechanism to reduce the search space as fast as possible due to the real-time response requirements.

- The internal robot architecture is defined as a three-layer hybrid architecture: the deliberative system, i.e. the case-based reasoning engine; the reactive system, i.e. a set of behaviors corresponding to skills the robot performs; and the low level, which includes the sensors and executors of the robot.

- The multi-robot architecture includes a set of robots called *retrievers* that incorporate the reasoning engine and therefore are in charge of deciding the cases to reuse, and the *executors*, who only perform the actions indicated by the retrievers (or default actions). However, any robot has the ability to abort the execution of a task when required.

- A coordination mechanism that enables the case reuse not through a single user, but through a team of users (in this case, the robots).

- A supervised learning process to acquire the scope of a case automatically.

Finally, in this dissertation we present empirical evaluation both in a simulated environment and in a real one with robots to prove the effectiveness of the proposed approach. Moreover, we argue that a collaborative behavior is advantageous to achieve the goal of the task, specially because of the adversarial component. It is well known that a good strategy to avoid an opponent during a game is to have passes between teammates. In contrast, using an individual strategy, where only one robot moves with the ball without taking into account its teammates, increases the chances for the opponent to block the attack, unless the robot is much faster than the opponent. Therefore, we have successfully included the pass action in our approach, which is not common, as far as we know, in this domain (Four-Legged League).

1.4 Publications

The following publications have been derived from this thesis:

- R. Ros, M. Veloso, R. López de Màntaras, C. Sierra and J.L. Arcos (2006), Retrieving and Reusing Game Plays for Robot Soccer. 8th European Conference on Case-Based Reasoning. *Advances in Case-Based Reasoning* of Lecture Notes in Computer Science, Volume 4106, pp. 47–61. Springer. ***Best paper award***.

- R. Ros, J.L. Arcos (2007). Acquiring a Robust Case Base for the Robot Soccer Domain. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 1029–1034. AAAI Press.

- R. Ros, M. Veloso (2007). Executing Multi-Robot Cases through a Single Coordinator. In *Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems*, E. H. Durfee, M. Yokoo eds., pp. 1264–1266.

- R. Ros, R. López de Màntaras, J.L. Arcos and M. Veloso (2007). Team Playing Behavior in Robot Soccer: A Case-Based Approach. In *Proceedings of the 7th International Conference on Case-Based Reasoning*. Case-Based Reasoning Research and Development of Lecture Notes in Computer Science, Volume 4626, pp. 46–60, Springer.

- R. Ros, M. Veloso, R. López de Màntaras, C. Sierra and J.L. Arcos (2007). Beyond Individualism: Modeling Team Playing Behavior in Robot Soccer through Case-Based Reasoning. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pp. 1671–1674. AAAI Press.

1.5 Outline of the Thesis

Next, we summarize the contents of Chapters 2 to 7. The core of the research work is described in Chapters 3 to 6.

Chapter 2: CBR Preliminaries and Related Work.

In this chapter we first review basic ideas of Case-Based Reasoning to familiarize the reader with the concepts used through the dissertation. Next, we present related work that describes the different techniques (including CBR) used by researchers in the past years within the robot soccer domain. A brief section is addressed to other robotic domains where CBR has been successfully applied.

Finally, the chapter concludes with a summary of the related work through a comparative table and where our work is located with respect to previous work.

Chapter 3: Modeling the CBR Approach: The Retrieval Step

This chapter corresponds to the first step of the CBR cycle, i.e. the Retrieval Step. Thus, we present the different components of the proposed CBR system, including: the case description, the case base structure, the similarity measures and the retrieval process itself. We also present experimental results in simulation to test the introduced process.

Chapter 4: Case Reuse through a Multi-Robot System

In this work the case reuse is fulfilled through a team of robots, instead of an individual robot. Hence, we not only have to define the internal robot architecture, but also the multi-robot architecture. In this chapter, we describe how the robots interact to perform the task, i.e. how to reuse the case in a coordinated way.

Chapter 5: Learning the Scopes of Cases

A first attempt towards the learning stage of the CBR cycle is presented in this chapter. More precisely, it is focused on automatically acquiring the scope of a case through a supervised learning algorithm. Different functions used in the algorithm are proposed to this end. The learning mechanism is evaluated both in simulation and with real robots.

Chapter 6: Experimentation

This chapter is devoted to the experimentation stage. To evaluate the overall system, we have performed experiments in simulation and with the real robots. The scenarios consist of two vs. two games, where two attackers play against a defender and a goalie. The CBR approach is compared with respect to a region-based approach. While the attackers use both approaches for evaluation, the opponents use a fixed behavior. Results are discussed and a trial example with real robots is described in detail.

Chapter 7: Conclusions and Future Work

In this last chapter, we summarize the conclusions addressed in each separate chapter. We also discuss future research lines and open challenges to improve the proposed approach.