Fostering Cooperation among Intelligent Machines

State of the Art

Rui Rocha

ISR – Universidade de Coimbra

Pólo II, Pinhal de Marrocos

3030-290 Coimbra, PORTUGAL

E-mail: rprocha@isr.uc.pt

Technical Report, September 2002

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the key driving forces in the development of mobile robotic systems is their potential for reducing the need for human presence in dangerous applications, in which human casualties are possible or even likely. Examples of such applications are: the cleanup of toxic waste, nuclear power plant decommissioning, planetary exploration, fire fighting, search and rescue missions, security, surveillance, etc. In these applications, it is desirable to reduce the risk to humans through the use of autonomous technology. In manufacturing applications, although being less risky, the use of autonomous technology may increase the work efficiency, due to the highly repetitive and monotonous nature of the inherent tasks.

One possible solution to create such autonomous systems is to try to build single agent solutions ([1]). This agent would have all the capabilities necessary to accomplish the specified mission on its own. This solution may be feasible for small-scale applications, however it is impossible or disadvantageous for the real world applications referred above. Usually, solutions to those applications employ the use of multiple human workers cooperating and complementing each other [Par94]. Some tasks, which are typified by the high potential for damage to individual collective elements, seem to be ideally suited to multi-robot systems, and thus it is the expendability of collective elements that is identified as the major reason for proposing robot collectives for the task [DJM02]. For some specific robotic tasks, such as exploring an unknown planet [AB98b], pushing objects [Par94, MNS95, RDJ95], or cleaning up toxic waste [Par98], it has been suggested that, rather than sending one very complex robot to perform the task, it would more effective to send a number of smaller and simpler robots. Such a collection of autonomous

[1] In such applications, agents are usually robots or robotic agents. We have chosen the term *agent* in order to keep the text as much general as possible, because these issues are not restricted to multi-robot systems.

agents is sometimes described as a *swarm* [JLB94], a *colony* [DMC96], or as a *collective* [KZ94], or the robots may be said to exhibit *cooperative behavior* [Par93].

There are some tasks that are more effectively performed by a single agent solution, namely those that neither are spatially distributed nor require some sort of synchronization [DJM02]. However, this is not the most usual case and there are several reasons why a multi-agent solution performs better than a single agent solution [AB98a]:

- *Space Distribution* – many agents can be in many places at the same time;
- *Parallelism or Time Distribution* – many agents can do many, perhaps different, subtasks at the same time;
- *Divide and Conquer* – certain problems are well suited for decomposition and allocation among many agents;
- *Cost, Reliability and Robustness* – often, each agent in a team can be simpler than a more comprehensive single agent solution.

The latter reason means that, although some tasks do not require a multi-agent solution, it is very difficult to realize a single agent system simultaneously complex and robust [Jun98], because there is usually a tradeoff between performance and reliability (Figure 1.1) ([2]). Multi-agent solutions give greater flexibility in managing complexity by distribution of risk. For example, instead of building a monolithic robot designed to have all the sensing, perceptual and reasoning capabilities required for a particular task, a multi-robot system is a more reliable solution. If there is an overlap in the individual robot's capabilities, then the system has a greater robustness, because a failure of any particular robot will not necessarily mean the failure of the whole system. Tasks that are traditionally multi-agent are typically parallelized and require small amounts of coordinating communication. Between this extreme and the tasks that are more suited to single agent solutions, there are tasks that could be performed faster or more reliably with a collective rather than with a single agent (e.g. finding a particular object in a finite region). Collectives of simple robots may be simpler in terms of individual physical design than a larger and more complex robot. Thus, the resulting system may be more economical, more scalable and less susceptible to overall failure [DJM02]. Multi-agent systems composed of situated, *flexible*, *autonomous* and *mobile agents*, usually denoted as *multi-robot systems*

[2]Figure reproduced from [Jun98].

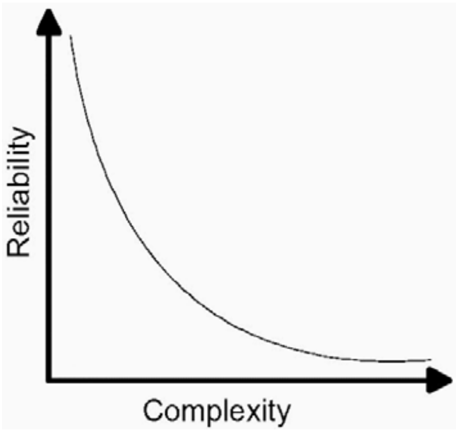

Figure 1.1: Tradeoff between reliability and complexity.

or cooperative robotics, are especially important for assisting or substituting human teams. A multi-robot system cannot be simply regarded as a generalization of the single robot case because the proposed solution need to be more precisely characterized in terms of assumptions about the environment and the internal system organization [INS01]. *Situated agents* are those that sense and act in a dynamic or uncertain environment. *Flexible agents* are both reactive and deliberative: they are reactive when responding to changes in the environment and deliberative when planning and acting ahead of time. *Autonomous agents* can automatically perform useful tasks without human intervention, even for long periods of time. *Mobile agents* are able to transport themselves within the environment. There are some other agents' properties that may be present, such as being *social* and having *learning* abilities.

Although such multi-agent solutions present many advantages over single agent solutions, they also present some challenges when operating in dynamic, uncertain and complex domains. The main research topics are:

- *Teamwork* – ensuring that the agents community acts in a coherent manner even when unexpected events occur in uncertain environments, such as: agents' failures; new information becoming available making current processing obsolete; unexpected requests being received that motivate the abandon of previous requests without maintaining the teamwork consistency; violation of inter-agent synchronization points [Jen95].

- *Multi-Agent Coordination* – coordinating in order to achieve overall coherence and performance with distributed control, given that each agent has only a partial view of the world and cannot make the best decision alone; reactivity vs. deliberation is a central issue on coordination strategies.

- *Cooperative perception* – understanding the world by fusing noisy observations from multiple robots to build a world model shared by the team, which should be purposeful to the team overall task and as much accurate and comprehensive as possible.

- *Cooperative planning* – decomposing complex tasks in a partial ordered set of subtasks, assigning subtasks to individual agents, conflict resolution and re-planning in the presence of contingencies (e.g. arrival of new tasks, failures or deadlocks).

- *Cooperative learning* – cooperating in order to learn coordinated behavior and to adapt to uncertain and dynamic environments.

- Other orthogonal issues, such as: assessing utility of shared information, i.e. what is task-relevant, what to share of individual world model, when to share, how to resolve conflicting information, etc.; and quantitatively evaluating cooperation.

In [CFK97], Cao et al. tried to summarize the research question regarding attaining cooperation: Given some complex task, associated to a system goal shared by all the agents, how to design cooperation mechanisms for the multi-agent system, so as the individual agents exhibit coherent individual behaviors that contribute for the efficient execution of the overall task in a manner that increases the total utility of the system? By other words, how to foster *cooperation* among the agents? This complex question can be divided into several subsidiary questions: (1) How to identify the types of mechanisms that need to be present within the agents to deliver desirable cooperation? (2) What forms of cooperation are appropriate and in what circumstances? (3) How should the agents act in a given social context to benefit most from the potential of cooperative problem solving? (4) How to predict what types of failures can occur in cooperation and in what conditions will they happen?

This report refers a significant part of research work on cooperative multi-agent systems and, particularly, multi-robot systems. It is structured in three parts and it tries to be a self-contained text. The first part is concerned to the essence of the cooperation concept and it is mainly devoted to biological manifestations of cooperation in nature, namely in human societies and animal species. This knowledge is relevant, because it has been an important source of inspiration on the design and implementation of human-made cooperative systems. The second part presents the most referenced taxonomies of cooperative systems. These taxonomies try to map out the space of possible designs for cooperative systems along different classification axes, which are a valuable help to organize the background that is already available about those systems. Moreover, taxonomies serve as design and implementation guidelines for new cooperative systems. The third part makes use of the taxonomies presented in the second part in order to cover a significant part of cooperative systems architectures and frameworks already known in the literature. The report ends with a short discussion about the state of the art and future research work on cooperative systems.

Chapter 2

The essence of cooperation

For some problems, especially those that are intrinsically distributed and complex, fostering *cooperation* among intelligent machines is driven by the assumption that multi-agent solutions have advantages over single agent solutions. Research on this issue has been mainly conducted by roboticists in the context of multi-robot systems. However, besides knowledge about building single robot systems, this research area is multi-disciplinary and integrates a huge number of distinct fields, outside the Engineering Sciences, where it bears inspiration to obtain a cooperative collective behavior upon engineering the behavior of individuals. Thus, it integrates Engineering Sciences, Artificial Intelligence, Social Sciences (Organization Theory, Economics, Cognitive Psychology) and Life Sciences (Theoretical Biology, Animal Ethology).

Distributed Artificial Intelligence, a sub-field of Artificial Intelligence (AI), is particularly relevant for studying cooperation among intelligent machines, as it studies the problems related with constructing large, complex and knowledge-rich systems. It advocates that such systems should be decomposed into a number of autonomous *agents* that communicate and cooperate with one another within a decentralized control regime. These systems are denoted as *multi-agent systems* (MAS) [SV00]. For the last two decades, MAS scientists have developed extensive work, providing both principles for constructing such complex systems, involving multiple agents, and mechanisms of coordination of independent agents' behaviors. Much of the research on non-robotic MAS is relevant to robotic MAS, which are usually denoted as *multi-robot systems*. The concept of *agent* plays a central role on these systems. Although there is no generally accepted definition of agent in AI, it may be defined as an autonomous and intelligent entity (e.g. a robot) with goals, actions and domain knowledge, situated in an environment. The way it acts is often called its *behavior*.

Before the word *cooperation* can be applied to rule the behavior between intelligent agents and between them and humans, some thought about its meaning must be given. Although its literal reading — simultaneous cooperation — is quite general, the word has historically been used primarily to refer to the joint behavior of humans, and sometimes animals. The specific mechanisms of cooperation we can find in the animal and human sphere depend on behavioral tendencies that effect the willingness to cooperate [Jun98].

Robotics researchers often distinguish between two types of cooperation: *collective robotics* and *cooperative robotics* [CFK97]. The former is often denoted as *swarm cooperation* and the latter as *explicit cooperation.* They are two different approaches to the same problem: how to obtain a desired *collective behavior* upon engineering the behavior of individuals. The term *collective behavior* denotes any behavior in a system having more than one agent (e.g. a multi-agent system). *Cooperative behavior* is a subclass of collective behavior that is characterized by cooperation.

Explicit definitions of cooperation in the robotics literature include [CFK97]:

- Joint collaborative behavior that is directed toward some goal in which there is a common interest or reward;
- A form of interaction usually based on communication;
- Joining together for doing something that creates a beneficial result, such as increasing the overall system performance.

These definitions emphasize three important dimensions of the cooperative behavior, namely *task*, *mechanism* and *performance.* The *task* is directed toward a goal shared by all the agents of the community, in which there is a common reward or interest beneficial to all the agents. The *mechanism* of cooperation, perhaps supported on some distributed control architecture and some explicit communication, rules the interactions among the agents, so that the actions of the individual agents are coherent with the system goal and beneficial to the system as a whole ([1]). *The performance* of the system, as a whole, is enhanced through the existence of cooperation, creating a beneficial result that is a reward for all the agents (e.g. reducing time to complete a task, increasing resources utilization, reducing energy waste, etc.). This means that cooperation renders a globally rewarding utility which is greater than the sum of the individual utilities.

[1]This statement has an implicit distinction between the goals of an individual agent and the system goal. Each agent may have its local goals and a system goal, common to all the agents. In this context, a given agent may have to choose some actions that, not representing a direct reward to its individual (local) goals, are required in order to benefit the system as a whole and to achieve the system goal.

A possible definition which encompasses all the three dimensions may be [CFK97]:

> *"Given some task specified by a designer, a system exhibits cooperative behavior if, due to some underlying mechanism (mechanism of cooperation), there is an increase in the total utility of the system."*

A more precise definition of cooperation from the Artificial Intelligence area is [DFJN97]:

> *"To cooperate is to act with another or others for a common purpose and for common benefit."*

There are two primary ways to give an agent a purpose: the agent is provided with a set of behaviors that are designed in such a way that the agent pursues some *implicit purpose* (*goal-oriented* or purely *behavior-based control*); alternatively, the agent is motivated by *explicit goals* and employs decision-making processes (planning, negotiating, etc.) to direct its action towards the achievement of those goals (*goal-directed control*). Sharing the same purpose (implicit or explicit) is not a sufficient condition for agents to achieve explicit or intended cooperation; they must intend also to act together or to have a commitment to joint activity. This is only possible with some goal-directed control because it is not possible without internal state and purely goal-oriented control.

A common definition of Distributed Problem Solving (one of the areas of Artificial Intelligence) is the cooperative solution of problems by a decentralized and loosely coupled collection of knowledge sources, located in a number of distinct processor nodes [Smi80]. The knowledge sources cooperate in the sense that no one of them has sufficient information to solve the entire problem; mutual sharing of information is necessary to allow the group, as a whole, to produce an answer; here decentralized means that both control and data are logically and often geographically distributed and there is neither global control nor global data; loosely coupled means that individual knowledge sources spend most of their time in computation rather than communication.

2.1 The emergence of cooperation

The concept of cooperation is a human, and possibly animal, symbolic concept whose meaning is intimately related to the behavioral and cultural references to which it is grounded [Jun98]. For this reason, the specific mechanisms we find employed for cooperation in the animal and human sphere depend on behavioral tendencies that effect the willingness to cooperate. The conditions under which organisms cooperate are complex and closely tied to the ecology of individual genes. Although the design of cooperative human-made systems are in a different context, knowing why organisms cooperate can help to identify particular conditions under which those systems may benefit from cooperation.

The most obvious question that arises when trying to understand why biological organisms cooperate is why do they cooperate, knowing that Darwin's natural selection theory implies that they behave in a completely selfish manner to increase their own fitness. Although individuals are genetically selfish, the main mistake of Darwin's theory is the belief that they are necessarily selfish. There are three main reasons that explain why cooperation emerges in biological societies: *pair bonding*, *kin selection* and *reciprocal altruism* [Jun98]. The first two reasons have a genetic basis, whereas the latter explains cooperation between unrelated individuals.

Sexual reproduction is an evolutionary advantage because it allows for faster adaptation to changing environmental conditions. Behavioral mechanisms are necessary to ensure that males and females mate and cooperate in child rearing. Long-term *pair bonding* provides a willingness of males and females to cooperate to achieve a variety of tasks related to secure reproductive opportunities and child rearing. Based on the application of Darwin's theory, the theory of *kin selection* is a selfish-gene approach that postulates that individuals cooperate to varying degrees with kin because they have genes in common. *Reciprocal altruism* is the process by which altruistic relationships arise between unrelated individuals. A given altruistic relationship is an *evolutionary stable strategy* if: (1) the cost of an altruistic act is low in relation to the received benefit; (2) individuals are able to recognize each other as individuals and to keep track of their history of previous dealings; (3) the group is stable, giving the individuals the chance to encounter each other repeatedly in situations that present opportunities for altruistic acts. Two individuals can profit by forming a relationship based on reciprocal altruism because it provides the opportunity to barter resources and information for mutual benefit. Human societies encourage a basic level of altruism through cultural controls over behavior, such as legal systems and social conventions. If an individual fails to show this basic level of altruism, he will lose social or legal status and hence resources. Moreover, this basic level of reciprocal altruism dictates how people expect other people to behave and is often inherent in what is meant when they talk of cooperation. Obviously, it is unlikely that human-made cooperative systems can cooperate by pair bonding or kin selection, but they can benefit from displaying reciprocal altruism toward each other, and toward humans.

2.1.1 The prisoner's dilemma

Political sciences' researchers have already studied the emergence of cooperation using game theory models. The *prisoner's dilemma* is a classic of game theory, which has been used to study interactions based on reciprocal altruism in different areas, such as political and social sciences, economy and biology. The situation inherent to the prisoner's dilemma occurs when selfish individuals, pursuing their own interests, lead to a poor outcome for the collective. In the prisoner's dilemma game, there are two players that have two different choices in each interaction: cooperate or defect. Each must make the choice without knowing what the other will do. No matter what the other does, defection yields a higher payoff than cooperation. The dilemma is that if both defect, both do worse than if both had cooperated.

This game was used by Robert Axelrod, a political sciences researcher, who tried to identify under what conditions cooperative behavior emerges in a group of selfish individuals without a central authority, where pursuing self interests does not imply the group welfare [Axe80a, Axe80b, Axe84]. Table 2.1 [Axe84] depicts the prisoner's dilemma game he used to pursue his work. One player chooses a row, either cooperating or defecting. The other player simultaneously chooses a column, either cooperating or defecting. Together, these two choices result in one of the four possible outcomes shown in the matrix. If both players cooperate, both get the reward for mutual cooperation ($R = 3$ points). If one player cooperates but the other defects, the defecting player gets the temptation to defect, while the cooperating player gets the sucker's payoff ($T = 5$ points and $S = 0$ points, respectively). If both defect, both get the punishment for mutual defection ($P = 1$ point). The four parameters were chosen so as $T > R > P > S$ and $R > (S + T)/2$. These conditions ensure that mutual cooperation gets a higher cumulative payoff in consecutive interactions than alternating between exploiting and being exploited (exploiting each other). An iterated prisoner's dilemma game (IPD) is a sequence of interactions, whose length is not known by the players. Each player knows the complete history of previous interactions with the other player, but it does not know the decision that will be chosen by the other player in the current interaction.

Given that individuals have a sufficiently large chance to meet again, so that they have a stake in their future interaction, Axelrod used IPD to explore the following general conditions for the evolution of cooperation: firstly, cooperative strategies must have success on a given environment, so as they can be adopted by the agents; secondly, these strategies must have success in dynamic environments with learning capabilities (learning agents), so as they can thrive and propagate in a population; thirdly, once cooperation is estab-

Table 2.1: Payoff matrix of the *prisoner's dilemma* game used by Axelrod (1984): C=Cooperate; D=Defect; T=Temptation to defect; R=Reward for mutual cooperation; P=Punishment for mutual defection; S=Sucker's payoff.

		Column Player	
		C	**D**
Row Player	**C**	$R = 3$, $R = 3$	$S = 0$, $T = 5$
	D	$T = 5$, $S = 0$	$P = 1$, $P = 1$

lished in a population on the basis of the reciprocity, it must protect itself from invasion by less cooperative strategies. In [Axe84], the following experience is described. Fourteen game theory specialists from different areas, such as mathematics, economy, psychology and sociology, were invited to participate in an IPD tournament. Each player created a computer program to participate in the tournament, which implemented a given strategy. All the players knew that one of the participant strategies decided randomly in each interaction. The 14 programs were confronted in a round-robin tournament, including confronts of each program with itself. The duration of the interaction was 200 iterations, but the participants did not know it at start. Along each of the 200 iterations, each program summed a score accordingly with Table 2.1. After all rounds, it was summed the overall score accumulated by each program to determine the winner. It was submitted strategies of very different nature, ranging from very simple to mathematically very complex. The objective of all participants was to sum the maximum score, perhaps cooperating most of the time and trying also to exploit the opponents with occasional defections. Surprisingly, the winner was the simplest submitted program: its name is tit for tat (TFT) and it was submitted by a Canadian psychologist (Anatol Rapoport). TFT is a very simple strategy that always cooperates in the first iteration. In the following iterations, it simply does whatever the other player did on the previous iteration: if the opponent defected in the previous iteration, TFT retaliates (defects); if, however, the opponent cooperated in the previous iteration, showing good will or regret, TFT cooperates as a way to establish a reciprocal cooperative relationship, beneficial to both players. The main conclusions of the experience were: the best strategies are nice, because they never are the first to defect; it would be possible to create variants of some submitted strategies that would have won against TFT. This last conclusion means that, probably, there is no a strategy that is always the best in a IPD, because the success of a given strategy always depends on the nature of the other interacting strategies, relying on a tradeoff between exploiting the good will of the opponents and cooperating with them.

Because the results of this tournament were not completely conclusive, a second tournament was conducted [Axe84]. The participants of this second tournament were aware of the detailed analysis of the first tournament. They were aware of the pitfalls of some strategies and also of some variants that would have allowed to get better results or even to win TFT. Although the number of participants grew a lot (62 participants against 14 in the first tournament), TFT won again and it was again the simplest submitted program. This surprising result leads to some conclusions about a successful cooperative strategy: *niceness*, *retaliation*, *forgiveness* and *clarity*. *Niceness* means that the strategy is never the first to defect. *Retaliation* means that it retaliates immediately after its opponent has defected, showing that it is willing to cooperate but not to be exploited. *Forgiveness* means that, after retaliating, punishment is ended as soon as the opponent cooperates. *Clarity* means a strategy that is easily identifiable and coherent, favoring the establishment of a cooperation relationship based on reciprocal confidence.

In [Axe84], it is also described that, after this TFT success, an evolutionary study was conducted to study if it may thrive, propagate and resist to invasions of less cooperative strategies. It was simulated several generations of a tournament, so that more successful strategies were more likely to be submitted in the next generation, and the less successful entries were less likely to be submitted again. The number of copies of a given entry in the next generation was proportional to its score in the current generation. Again, the results provided a victory for TFT: by the one thousand generation it was the most successful rule and it was still growing at a faster rate than any other rule. It was also stated some propositions about stable strategies and it was proven that TFT is an evolutionary stable strategy, which can thrive and protect itself with a cluster of individuals who rely on reciprocity.

2.2 Examples of cooperation in the animal kingdom

Cooperation between simple organisms on earth is almost as old as life on earth itself. Biologists have long understood that bacteria live in colonies, but only recently it has become evident that most bacteria communicate using a number of sophisticated chemical signals and engage in altruistic behavior

[Jun98]. They emit and react to chemicals in a genetically determined way that associates chemical and elicited behavior. This can be considered an *interaction via the environment*, as the chemical environment, immediately surrounding each bacterium, acts as a communication channel for information implicit in the emitted chemicals that must be sensed and reacted to. These chemical signals only have meaning when interpreted in a behavioral context and they are an explicit signaling and a consequence of the evolutionary history of bacteria [JZ00]. The interaction distance is moderate compared with the size of a bacterium, and the simultaneity (period between the signal emission and reception) is determined by the speed of chemical propagation. As the emitter generates a signal without interpreting it, the communication does not preserve the signal meaning and the receiver interprets it iconically ([2]). The resulting cooperative behavior emerges as a consequence of the behavior policy genetically encoded in each individual. This mechanism of cooperation is simple as there is no recognition of other individuals, neither explicit communication.

Social insect societies have been thoroughly studied by biologists, especially ants, termites, bees and wasps [BG00]. For example, termites collectively build huge nests and ant colonies plan shortest paths between their nest and a food source, using a powerful signaling mechanism, which is also a kind of interaction via the environment: the exuding of a pheromone — a chemical substance — attracts other ants. When ants forage food sources, they lay and follow trails of pheromone. The first ants returning to the nest from the food source are those that have taken the shorter path in both directions. Because this route is the first to be doubly marked with pheromone, the other ants are attracted to it and tend to follow the optimized route. Path planning is an emergent characteristic of the ant colony not present at the level of the individual. In this communication scheme, the interaction is local because the receiver senses the pheromone at the location it was emitted. As the signal persists in the environment for long periods, there may be significant delay between emission and reception. This signaling mechanism is likely to be explicit, and the interpretation is relatively simple. Since both emitter and receiver can interpret the signal in the same way, the communication preserves the meaning for the signals, being the crucial element both agents sharing the same grounding for the signal, probably genetically determined. This constitutes also iconic representation, which is grounded directly in the environment for all individuals identically [JZ00]. Ants also

[2]Iconic representation is by physical similarity to what it represents. A chemical receptor on the bacteria's surface triggers a chain of chemical events in a stereotypical way that is an icon for the presence of the external chemical signal.

extensively use cooperative mechanisms that involve explicit (intentional) sensing of other individuals. Ants achieve the identification of castes of other individuals using chemicals sensed with their antennae (they cannot identify specific individuals). Although this *interaction via sensing* is a more sophisticated interaction than broadcast style of interaction via environment, the former is built upon the same mechanisms of the latter (chemical signals). The interaction via sensing is built and layered upon interaction via environment. As ants, animals in general, which use more sophisticated schemes, have also more basic schemes upon which the more sophisticated ones are built. Scientists who study the behavior of social insects have found that although the individual activities appear seamlessly integrated, without any supervision, the group cooperation at colony level is largely self-organized. The coordination simply arises from interactions among individuals. Although these interactions might be simple (e.g. following the trail left by another), together they can solve difficult problems (e.g. finding the shortest route among countless possible paths to a food source) and emerge a beneficial collective behavior, denoted as *swarm intelligence* [BG00].

The wolves, social mammals of the canine family, are carnivores that usually hunt in packs, formed upon strict social hierarchies and mating systems. They organize themselves, demarcating territories. Territory marking is done through repeated urination on objects on the periphery and within territories. Wolves also communicate with pheromones excreted via glands near the anus and the dorsal surface of the tail. As the chemical trails of antes, these are also examples of schemes based on interaction via environment. Wolves also interact via sensing when they hunt in packs: they cooperate by closely observing the actions of each other and, in particular, the dominant male who directs the hunt. Each wolf knows all the pack members and can identify them individually, visually and by smell. Wolves can also interact *via explicit communication*, as they communicate explicitly with a particular individual using a combination of specific postures and vocalizations [Jun98]. In this case, the interaction distance is the visual or auditory range and the emission and reception is effectively simultaneous. The signals may be implicit, in the case of observing locomotive behavior, or more explicit in the case of posturing, vocalizing and scent markings [JZ00]. These communicated signals have the same meaning to both emitter and receiver because both have a shared grounding that is learnt during development in a social environment similar to both. As with ants, wolves' communication exhibits meaning preservation for the signals, but with a significant difference: the shared grounding that enables the uniform interpretation of some signals (e.g. postures and vocalizations) is not wholly genetically determined. Instead, the grounding is partially learnt during development in a social environment similar to both individuals that ensures a shared meaning.

Primates also use each of the three mechanisms referred above — *interaction via environment*, *interaction via sensing* and *interaction via explicit communication*. The main difference between primates and other animals is their sophistication in learning and representing the internal goals, plans and actions of others, and their ability to construct cooperative plans jointly and flexibly adapt and repair them in real time [Jun98]. A joint plan can be defined as a sequence of actions, each enacted by a particular member of the group. Each individual assesses the goals, actions and plans of others, and adjusts its own goals, actions and plans to achieve a more coordinated interaction where joint goals are satisfied. Non-human primates use extensively the passive observation of others (interaction via sensing), via visual and auditory cues interpreted as actions and intentions. The interaction is simultaneous and occurs within visual or auditory range. The signaling is implicit (side effect of the behavior) but the sophistication of its interpretation is considerable [JZ00]. As with the wolves, the communication also exhibits meaning preservation through a shared grounding. However, the groundings are more complex, as is the development process required to attain them.

Humans own the heritage of our primate ancestors, using many types of signaling for communication [JZ00]. Like primates, we make extensive use of implicit communication, such as posturing and explicit gesturing (e.g. pointing), but we also make extensive use of explicit communication, both written and spoken, that is explicitly evolved or learnt. Posturing, gesturing and speaking all involve simultaneous interaction. Human have developed symbolical communication, which enables long-term interactions (e.g. written language). It requires considerable sophistication in interpretation, but we also use signals that are more easily interpreted, like laughing. Being the shared groundings for human symbolic communication more complex, our cultural language learning can be seen as an extension of the process present in our non-human primate ancestors. Based on these communication mechanisms, humans display a basic level of altruism and cooperate in many and varied ways toward humans and sometimes animals, fostering symbolic contracts with mates, kin, friends, organizations and societies, whereby we exchange resources for mutual benefit. In some cases, we cooperate and provide resources with no immediate reward, except the promise that the other party will honor the contract and will provide resources when we need them.

As a conclusion of the previous biologic examples, we may say that the sophistication of cooperation increases as we go from bacteria cooperation to primate and human cooperation, and this seems to have a high correlation with the increase of the sophistication of the communication schemes. It is likely that sophistication of cooperation scales with that of communication.

In [Par94], it is referred a broad classification of animal societies, which has particular interest for research in cooperative robotics, as it parallels two possible approaches to cooperative systems development. Animal societies can be grouped into two broad categories: those that *differentiate* and those that *integrate*. Insect societies are an example of societies that *differentiate* because they arise due to an innate differentiation of blood relatives that creates a strict division of work and a system of social interactions among members. Members are formed within the group according to the needs of the society. The individual exists for the good of the society and is totally dependent upon the society existence. A group can make accomplishments that are not possible to achieve individually. On the other hand, societies that *integrate* depend upon the formation of groups of individuals that are independent animals to each other. Such groups do not consist of blood relatives that stay together, but instead consist of individuals of the same species that come together by integrating ways of behavior. These individuals are driven by a selfish motivation that leads them to seek the group life, because it is in their own best interests. Wolves that hunt in packs are an example of this kind of cooperative societies. Another example is breeding colonies of many species of birds, in which birds do not come together due to any blood relationship, but instead they thrive the support provided by the group. Rather than the individual existing for the good of the society, these societies exist for the good of the individual.

Chapter 3

Taxonomies of cooperative systems

A key difficulty in the design of cooperative systems, relying on a multi-agent structure, is the size and complexity of the space of possible designs. Thus, an understanding of the many possible system configurations is essential to take principled design decisions [DJMW93, DJMW96, DJM02]. It provides for the succinct description of systems and results in the literature and it maps out the space of possible designs for a collective, giving the researcher guidance and perspective. This suggests that is useful to find a descriptive taxonomy for describing and classifying cooperative systems along different axes, like group architecture, communication and computational capabilities, resource conflicts, origins of cooperation, learning, geometric problems, etc.

There have been a number of efforts to develop descriptive categories for describing robot collectives. For example, in [YP92], Yuta and Premvuti subdivided collectives based on the interactions of collective elements, i.e. whether individual elements work towards a common objective or they work independently towards their own objectives. In [ABN93], Arkin et al. examined different collectives dedicated to foraging activities (retrieval tasks) along several dimensions, such as reactive vs. hierarchical planning and no communication vs. state communication. This work was continued on [BA94], where there was investigated the tradeoffs between no communication (implicit communication) and explicit communication, and in the latter case, between state communication and goal communication, through simulation experiments of three types of tasks: forage (look for things), consume (look for things and then do work there removing it) and graze (consume everything).

In [Bal02], Balch presented two highly focused taxonomies of multi-robot systems, illustrated with practical examples of multi-robot tasks and reinforcement learning configurations. The first is a features' taxonomy of the multi-robot task to be accomplished:

- *Duration time* – fixed, minimum, unlimited, or synchronized;
- *Optimization Criteria* – over finite period, average over all future, or discount future performance geometrically;
- *Subject of Action* – movement/placement of objects or movement/placement of robots;
- *Resource Limits* – limited external resources, minimum energy, competition for resources between team members, or competition with external agencies;
- *Group Movement* – robots spread apart (coverage), robots converge, movement to a position (configuring a formation), or movement while maintaining a formation;
- *Platform Dependencies* – a single agent can perform task, multiple agents are required, agents must be dispersed, can sense all relevant features, world is partially observable and communication is required.

The second is a taxonomy of rewards, assuming a reinforcement learning framework [Bal02]:

- *Source of Reward* – internal reward based on sensor values, reward generated by external agent or combined internal and external reward;
- *Relation to Performance* – reward tied directly to performance or based on intuition of state value;
- *Time* – immediate or delayed rewards;
- *Continuity* – reward takes on discrete values or continuous values drawn from an interval;
- *Locality* – individual agents receive unique rewards, all agents receive identical reward signal, or combination of local and global.

In [Par00], Parker presented a survey of research areas in distributed mobile robot systems, namely biological inspirations, communication, architectures, task planning and control, localization, mapping and exploration, object transport and manipulation, motion coordination, reconfigurable robots and learning. It was also identified open research issues and claimed that biological influences, as emulations of known biological systems on simple tasks involving cooperation and/or competition, are well understood, whereas biological approaches to complex tasks, such as robot soccer and especially their learning aspects, are still wide open.

In [CFK97], Cao et al. made an extensive survey or research in cooperative robotics, presenting a comprehensive set of references. It was proposed a taxonomy focused on problems and solutions, along five research axes:

- *Group Architecture* – it defines whether the control is centralized or decentralized, whether robots are differentiated or not, communication and interaction structures and modeling of other agents;

- *Resource Conflicts* – how members deal with resource conflicts when there is a need to share space, physical resources (e.g. tools) or communications media;

- *Origin of Cooperation* – how cooperation is motivated and achieved;

- *Learning* – how the collectives adapt to the task, using evolutionary techniques, such as reinforcement learning, neural networks and genetic algorithms;

- *Geometric Problems* – how robots interact with each other physically, addressing issues such as path planning and keeping a formation.

As these research axes are highly interdependent and very broad, it is difficult to identify isolated sample points within the taxonomy. Instead, Dudek et al., in [DJMW93, DJMW96, DJM02], did not expand the axes of comparison to include learning and the geometric structure of the problem, and concentrated on defining a taxonomy within which different robot collectives could be compared and contrasted. The taxonomy classifies robotic collectives along seven axes, which address characteristics of the collective as a whole, rather than the architectural characteristics of individual robots. Dudek et al., in [DJM02], illustrate its application on classifying some multi-robot systems case studies, accordingly with the descriptive tags presented on Table 3.1 ([1]). The seven classification axes are:

- *Size of the Collective* – alone (one single robot), a pair of robots (the simplest group), n multiple robots with n small relative to the size of the task or environment, or an infinite (very large) number of robots;

[1] Table reproduced from [DJM02].

- *Communication Range* – robots cannot communicate directly, robots can only communicate with other robots which are sufficiently nearby, or robots can communicate with any other robot;

- *Communication Topology* – each robot communicate with all of the other robots (broadcast), every robot communicate with any arbitrary robot by name or address, robots are linked in a tree and may only communicate through the hierarchy, or robots are linked in a general graph different from a tree (eventually with redundant connections);

- *Communication Bandwidth* – sufficiently high that communication cost and overhead can be ignored (infinite), communication costs of the same order of magnitude of the cost of moving the robot between locations, communications much more costly than moving from one location to another, or no possible communication (zero bandwidth);

- *Collective Reconfigurability* – fixed topology (static), coordinated rearrangement (e.g. formation control), or dynamic (members can arbitrarily change their roles);

- *Processing Ability of each collective unit* – non-linear summation unit (basic element of a neural network), finite state automaton, push-down automaton, or turing machine equivalent;

- *Collective Composition* – identical (homogeneous in both hardware and behavior), homogeneous (same physical characteristics), or heterogeneous (units not physically uniform).

In [INS01], Iocchi, Nardi and Salerno presented a multi-robot system taxonomy, using a top-down approach to refine the level of the system's structure representation. The taxonomy includes four different levels (Figure 3.1) ([2]): *cooperation level*, *knowledge level*, *coordination level* and *organization level*.

The *cooperation level* is concerned with the ability of the system to cooperate in order to accomplish a given global task. A multi-robot system is considered a cooperative system if team members operate together in the same environment and have a common goal to achieve. The *knowledge level* characterizes how much knowledge (awareness) each robot has about the presence of other robots in the environment. If robots are completely unaware, they perform their tasks as if they were the only robots within the environment. Cooperation among unaware robotic agents is the weakest

[2]Figure reproduced from [INS01].

Table 3.1: Taxonomy of multi-robot systems by Dudek et al. (2002).

Description	**Meaning**
Size of the Collective	
SIZE-ALONE	1 robot.
SIZE-PAIR	2 robots (the minimal collective).
SIZE-LIM	Multiple robots.
SIZE-INF	Infinite number of robots.
Communication Range	
COM-NONE	No communication.
COM-NEAR	With robots nearby.
COM-INF	With any robot.
Communication Topology	
BAND-INF	Infinite or not restrictive.
BAND-MOTION	Communication costs as much as moving the robot between locations.
BAND-LOW	Very high cost.
BAND-ZERO	No communication.
Collective Reconfigurability	
ARR-STATIC	Static arrangement.
ARR-COMM	Coordinated rearrangement.
ARR-DYN	Dynamic arrangement.
Processing Ability	
PROC-SUM	Non-linear summation unit.
PROC-FSA	Finite state automaton.
PROC-PDA	Push-down automaton.
PROC-TME	Turing machine equivalent.
Collective Composition	
CMP-IDENT	Identical.
CMP-HOM	Homogeneous.
CMP-HET	Heterogeneous.

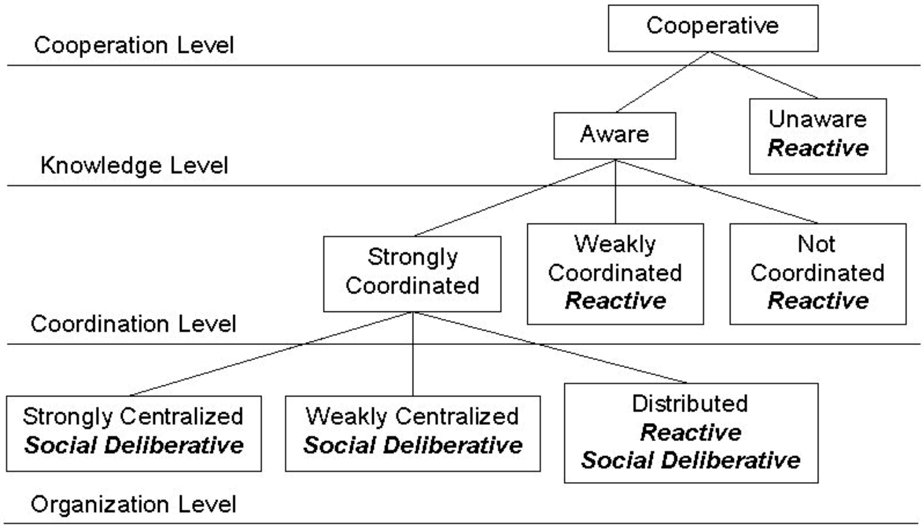

Figure 3.1: Taxonomy of multi-robot systems by Iocchi et al. (2001).

form of cooperation. The *coordination level* classifies the coordination mechanisms used by robotic agents to take into account the actions executed by the other teammates, in such a way that the group operates in a coherent and efficient manner. Coordination requires some awareness about others and enables explicit cooperation and systems with no awareness are necessarily not coordinated. However, coordination is not a sufficient condition to achieve cooperation, e.g., robotic agents may coordinate to avoid interference although they have different and independent goals. On the other hand, for some simple space distributed and highly repetitive tasks, cooperation may emerge in the absence of coordination mechanisms (e.g. foraging task). These weakly coordinated multi-robot systems tend to be more robust to failures (e.g. communication failures), but they lack of many organizational capabilities offered by coordination protocols, which can minimize waste of resources and interference. In general, the more the tasks or mission is complex the more a strongly coordinated system is required to effectively achieve the goal. The *organization level* characterizes the way the decision system is organized, i.e. if it is centralized (strongly or weakly) or decentralized.

In [INS01], Iocchi et al. mention two more taxonomic dimensions, which are orthogonal to the previous ones, namely *communication* and *system composition*. The former has a strong influence on the coordination and organization levels (e.g. strongly coordinated systems necessarily require extensive communication). The latter is concerned with the differentiation among the

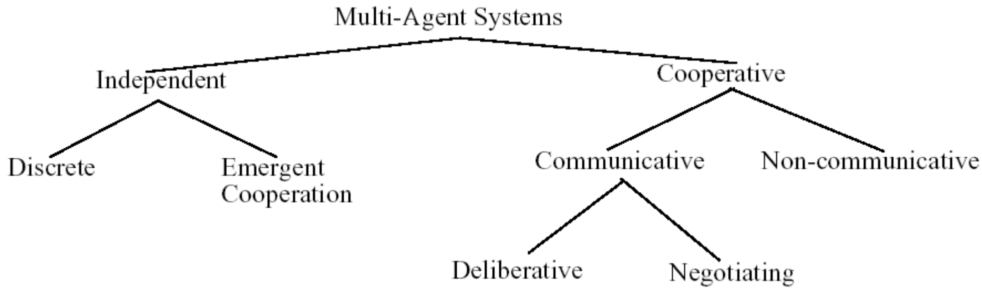

Figure 3.2: Cooperation typology from Artificial Intelligence.

robots, ranging from homogeneous to heterogeneous systems.

3.1 Multi-agent systems taxonomies from Artificial Intelligence

One of the cooperation typologies that Artificial Intelligence researchers usually consider is depicted in Figure 3.2 [DFJN97] ([3]). A multi-agent system (MAS) is *independent* if each agent pursues its actions independently of the others. A MAS is *discrete* if it is independent and if the actions of each agent have no relation to other agents' actions. *Discrete systems* do not exhibit cooperation. However, agents can cooperate with no intention of doing so. In this case, the agents appear to be working together although they simply carry out their own individual behavior, rendering an emergent cooperative behavior denoted as *emergent cooperation.* The complement of independent systems is systems in which the agents include in their actions the explicit intention for cooperating with other agents (*cooperative systems*). Such cooperation can either be *communicative* or *non-communicative.* In the former case, the agents communicate with each other, sending and receiving intentionally signals, in order to cooperate. In the latter case, agents coordinate their cooperative activity by each observing and reacting to the behavior of the other. Communication in communicative cooperative systems can take two forms: *deliberative* or *negotiating.* In deliberative systems, agents jointly plan their actions so as to cooperate with each other through coordination. Negotiating systems are like deliberative systems, except that they also include some competition.

Another alternative typology of cooperation in Artificial Intelligence can

[3] Figure reproduced from [DFJN97].

be considered if we look at cooperation as a property of the actions of the agents involved [DFJN97]. Given a multi-agent system in which individuals may be assigned one or more goals, cooperation occurs when the actions of each agent satisfy either or both the following conditions:

- The agents have a goal in common, which no agent could achieve on its own, and their actions tend to achieve that goal;
- The agents perform actions that enable or achieve not only their own individual goals, but also the goals of the other agents.

These two conditions may be achieved either with emergent cooperation or intentional cooperation. While in the former case the goals are not explicit within the agents, in the latter case the goals are explicit and the agents deliberate and/or negotiate their plans.

Cooperation may also be classified by *patterns of cooperative actions*, arising from different intra-agent processes [DFJN97]. With *reactive agents*, they simply act without reflection upon possible actions and prediction or predictive planning and, therefore, with no intention (emergent cooperation). Unlike reactive agents, *deliberative agents* reflect upon the combinations of actions they and others might perform. They choose between different combinations of action, possibly after some negotiation process, leading to a convergence of their behavior. *Concept based agents*, endowed with a *beliefs-desires-intentions* knowledge structure, maintain an explicit concept of cooperation that they use to select their actions. Thus, concept based agents may decide to cooperate prior to any particular set of actions being considered. With agents either deliberative or concept based, cooperation can also be classified by the *degree of altruism* implicit in the cooperative actions. If the agent deliberates and selects only those cooperative actions that it believes will further its own individual goals (selfish agent), cooperation is purely self-interested. If, otherwise, the agent deliberates and selects cooperative actions that it believes to further a groups' interests (a group of which it is a member) irrespective of its own individual goals, cooperation is partly or wholly altruistic.

The cooperative actions may be originated from distinct sources and this is an orthogonal question [DFJN97]. If they come from *explicit design*, the creator of the agents deliberately designs the agent's behaviors or rules so that various instances of cooperation occur. It they come from *adaptation*, the agents are endowed with learning capabilities that enable them to develop or augment their tendency to cooperate along their lifetime. If they come from *evolution*, the population of agents selects them through an evolutionary process.

In [SV00], Stone and Veloso survey multi-agent systems, with a focus on learning issues that arise because of the presence of multiple interacting agents, and proposes a taxonomy along what are believed to be the most important aspects of agents, namely *degree of heterogeneity* and *degree of communication*. Using a classic multi-agent example — the predator/prey domain ([4]) — it was described the main characteristics, research issues and techniques of four different scenarios — *homogeneous non-communicating agents*, *homogeneous communicating agents*, *heterogeneous non-communicating agents* and *heterogeneous communicating agents* — highlighting learning opportunities of each scenario. In this context, there are referred learning techniques such as reinforcement learning or genetic programming.

In *homogeneous non-communicating systems*, all of the agents have the same internal structure, including goals, domain knowledge and possible actions; they only differ on their sensory inputs and the actual actions they take, i.e. they are situated differently in the world [SV00]. The agents can be either *reactive* or *deliberative*. While in the former case the agents simply retrieve pre-set behaviors similar to reflexes without maintaining any internal state, in the later case they maintain an internal state, search through a space of behaviors and predict the effects of their actions. Some issues touched upon include whether agents have global or local view and whether an agent alters the environment so as to either affect the sensory input or the effects of another agent's actions. Giving the agents a global perspective is not always more effective than to limit them to local views. Sometimes, knowing just enough to coordinate well is preferable than maintaining global knowledge. When no communication is possible, agents cannot interact with each other directly but, coexisting in the same environment, they can affect each other indirectly. If they can be sensed by other agents, they may be able to change the state of another agent. Agents can also affect each other by one of two types of *stigmergy*: by *active stigmergy*, they can alter the environment so as to affect the sensory input of another agent; by *passive stigmergy*, they can alter the environment so that the effects of another agent's actions change.

In *heterogeneous non-communicative systems*, besides agents being dif-

[4]The predator/prey domain, or pursuit domain, is a foraging game that involves agents moving around in a virtual world, usually a prey and several predators [SV00]. The goal of the predators is to capture the prey, or surround it so that it cannot move anywhere. The virtual world is typically a grid-like world with square spaces and may be finite (e.g. a small finite board with edges) or infinite (e.g. a toroidal world where an agent can move off on end of the board and come back on the other end). The agents can move around the world, being allowed to move diagonally, horizontally and vertically, or simply horizontally and vertically.

ferently situated in the environment, they may be different in any number of ways, from having different goals to having different domain models and actions [SV00]. They may be either *benevolent* or *competitive*. In the former case, they are willing to help each other to achieve their respective goals, or they are organized as a team and it must be provided some method for assigning different *roles* to different agents [Tam97]. In the latter case, the agents may be involved in a zero-sum situation so that they must actively oppose other agents' goals in order to achieve their own. However, whether agents exhibit altruism or not, if they are not truly competitive and are not involved in a zero-sum situation, it may be beneficial for them to cooperate each other and cooperation may even emerge among selfish agents [Axe84]. Agents may also differentiate and become heterogeneous if their behaviors are not fixed and they can learn from experience. This learning capability is crucial in dynamic environments. Whether agents are benevolent or competitive, if they have learning capabilities they are either *cooperative learning agents* or *competitive learning agents*. In the latter case, as it is not clear whether the improvement is due to an improvement in that agent's behavior or a negative change in the opponent's behavior, there is a credit assignment problem that may impede agents to stabilize at a good behavior. Being modeling of other agents an important requirement to coordinate the agents' actions, it is much more complex in the heterogeneous case than in the homogeneous case. Without communication, agents are forced to model each other strictly through observation, because goals, actions and domain knowledge of the other agents may also be unknown and thus need to be modeled. This kind of implicit communication may be used to reach *tacit social conventions* for agents to coordinate without explicit communication.

Homogeneous communicating systems inherit all the characteristics of the non-communicating case, but agents can communicate directly (e.g. broadcast, blackboard based or point-to-point). This additional feature raises some issues that are, however, usually addressed in literature about heterogeneous communicative agents [SV00]. There must be some set language or protocol so that agents can *understand each other*. Important aspects for the communication protocol, i.e. *how to communicate*, are information content, message format and coordinating conventions ([5]). When an agent transmits information to another agent, it has an effect just like any other action would have. Within a *planning communicative acts* framework, it can be defined preconditions and effects for communicative acts. When combined with a model of other agents, the effect of a communication act might be to alter an

[5]Rather than prescribing syntactic rules for communication, it is fundamental defining its semantics, i.e. *what to communicate*.

agent's belief about the state of another agent. The theory of communication as action is usually denoted as *speech acts* [CL91]. *Negotiation* allows agents to resolve conflicts that could interfere with cooperative behavior. It can be defined as the process of improving agreement on common viewpoints or plans through the structured exchange of relevant information, as means to reduce inconsistency and uncertainty [OJ96]. One of the most influential negotiation approaches is the *contract-net protocol* proposed in [Smi80], based on the law of supply and demand. In the contract nets framework, agents all have their own goals, are self-interested and have limited reasoning resources. When agents communicate, they may decide to cooperate on a given task or for a given amount of time, making *commitments* to each other, which involves agreeing to pursue a given goal. There are three types of commitment [SV00]: *internal commitment*, when an agent binds itself to do something; *social commitment*, when an agent commits to anther agent; and *collective commitment*, when an agent agrees to fill a certain role. Commitment states have been used as planning states within the *beliefs-desires-intentions* model, which is a popular technique for modeling other agents.

3.2 Swarm vs. explicit cooperation

Robotics researchers often distinguish between two types of cooperative research: *collective* or *swarm robotics* and *explicit cooperative robotics*. Although these two approaches look to cooperation from different angles, both approach the same problem: how to obtain a desired group behavior from a multi-agent system, by engineering the behavior of individuals.

Swarm cooperation assumes that cooperation is not explicitly designed into the system: cooperation is not predefined but emergent [BHD94, JLB94, KZ94, DMC96]. *Collective robotics*, which is also known as *swarm robotics*, may be defined as the distributed control of homogeneous robot teams, whose collective dynamic is obtained as an emergent property of the local interaction between the behaviors designed in the individual robots [CFK97]. These behaviors are often reactive or behavior-based, using mainly local sensorial information. This approach relies on the anti-classical Artificial Intelligence idea that a group of robots may be able to perform tasks without centralized control or the provision of a global model (explicit representation of the environment and of the other robots) and that predictive planning may be replaced by reactivity. These reactive systems does not present any decision-making process based on decomposition, allocation or accomplishment of tasks. The outputs of each agent are a direct function of the observations sensed on the environment [Bot00].

Swarm intelligence is defined as a "property of systems of non-intelligent robots exhibiting collectively intelligent behavior" [HB91, JLB94]. Another definition [BDG99] states that *swarm intelligence* "is the property of a system whereby the collective behaviors of (unsophisticated) agents interacting locally with their environment cause coherent functional global patterns to emerge". This concept has inspired some researchers on the development of swarm solutions for complex problems from different areas: the cooperative interaction of ants working to transport a large food item may lead to more effective algorithms for robots; foraging of ants has led to a novel method of routing of traffic in a busy telecom network and new solutions for the classical traveling salesman problem; the way in which insects cluster their colony's dead and sort their larvae can aid in analyzing banking data; the division of labor among honeybees could help make more efficient assembly lines in factories. Swarm cooperation is found in nature and it is denoted as eusocial behavior. *Eusocial behavior* is found in many insect species (e.g. colonies of ants or bees) as a result of genetically determined individual behavior [CFK97]. Although individual agents are not very capable, intelligent behavior arises out of their interactions and is vital for the survival of the individuals in the colonies. This kind of biologic knowledge about simple local control rules of various biological societies — particularly ants, bees and birds — has been often applied to the development of similar behaviors in cooperative robot systems.

Swarm cooperation will be probably a promising approach in many potential applications based on new robot technologies characterized by miniaturization, like small, micro, nano and modular robots [YZD02], which will have severely limited sensing and computation. These very small robots operating in large groups or swarms would be capable of performing complex tasks, like moving objects bigger than individual robots, in non-structured terrains and with high robustness, versatility and adapting quickly to rapidly fluctuating mechanisms. Giving that these collectives do not use explicit communication, they may be suited to execute reliably tasks typically parallelized, requiring small amounts of coordinating communication. Swarm intelligence approaches have been effective at performing a number of optimization and control tasks, but the systems developed have been inherently reactive and lack the necessary overview to solve problems that require in-depth reasoning techniques and some cognition [BG00]. Although this approach may maximize reliability, it fails to maximize performance, as members of the collective cannot be directed to uncompleted work that they cannot sense directly [DJM02].

Explicit cooperation means that cooperation is explicitly designed into the system, through adequate mechanisms. Unlike eusocial behavior, explicit cooperative mechanisms are not motivated by innate behavior, but by an intentional desire to cooperate in order to maximize individual utility. It deals with achieving intentional and more purposeful cooperation among a limited number of typically heterogeneous agents, performing several distinct tasks, but it may be used also with homogeneous teams [Par94]. In the former case, individual agents have specialized capabilities complementing each other. In contrast with the swarm approach, the agents often have to deal with some sort of efficiency constraint that requires a more directed type of cooperation, because the mission usually requires that several distinct tasks be performed concurrently. Although individual agents are typically able to perform some useful task on their own, groups of such agents are often able to accomplish missions that no individual robot can accomplish on its own. In these systems, the cooperative dynamic is commonly achieved by planning and is based on multi-agent systems' explicit models of teamwork coordination and negotiation mechanisms, using explicit communication. The negotiation mechanisms rule the allocation of several distinct roles or subtasks to the individual agents and the resolution of conflicts. This approach usually employs either central control or a mix of central and distributed control, supported on a global model [Jun98]. Besides action recognition, through perception mechanisms based on sensorial information, explicit communication using the exchange of communicated messages provides the required awareness for modeling and reasoning about other agents' goals, actions and states. Thus, the reliability and fault tolerance is highly dependent upon the presence of a reliable communications medium with a sufficient bandwidth [Par95].

We can found two types of decision-making processes in architectures that implement the concept of explicit cooperation: *deliberative systems* and *conceptual systems* [Bot00]. *Deliberative systems* have a decision-making process that anticipates the agents' actions, providing detection and resolution of conflicts and establishment of cooperation among the agents. Deliberative systems may be *planned* or *reactive*. In the former case, the agents perform autonomous planning actions which try to minimize inter-agent interference and generate a sequence of actions that change the world state, so as a desired final state is reached. In the latter case, the systems are also based on plans, but they are able to adapt a plan during its execution, in reaction to environmental changes, so as there is always a feasible plan. *Conceptual systems* are multi-agent systems based on *beliefs-desires-intentions* model [OJ96] in which the agents are able to reason about the mental states of its teammates, which are classified on beliefs, desires and intentions. These three elements represent, respectively, information, motivations and deliberative states of the agent itself and its teammates. The objective is guiding the behavior of the system so as an adequate or optimal performance is obtained, through the reasoning capacities resident on each individual agent.

In [INS01], the taxonomy proposed by Iocchi et al. (Figure 3.1, page 22) is used to relate the four classification levels with reactivity and social deliberation. *Reactivity*, which is typical in *swarm cooperation*, is a system behavior in which each team element copes with environmental changes, by providing a specific solution to reorganize its own task and fulfill the accomplishment of its originally assigned goal. On the other hand, *social deliberation*, which is typical in *explicit cooperation*, is a system behavior that allows the team to cope with those environmental changes, by providing a strategy that, when adopted by all the team members, makes use of all the resources available to the system to effectively achieve the global goal. Figure 3.1 (page 22) shows how the characteristics of the multi-robot system impacts on the implementation of reactivity and social deliberation.

3.2.1 Artificial Intelligence teamwork models

A common dictionary definition of *teamwork* is: "cooperative effort by the members of a team to achieve a common goal". Many Artificial Intelligence researchers are today striving to build teams for complex, dynamic multi-agent domains where uncertainties obstruct coherent behavior. It is claimed that teamwork in such domains is more than the union of simultaneous coordinated activity. Cohen and Levesque, in [CL91], illustrated this point using a convoy example and focusing the distinction between ordinary traffic and driving in a convoy. Ordinary traffic is simultaneous and coordinated by traffic signs, but it cannot be considered teamwork or a joint activity. Conversely, driving in a convoy does involve not only coordination and having the right intentions and the right beliefs about each other (a joint goal to maintain the convoy), but also having a mutual belief of what agents agreed to (joint commitment), which underlie the formation of joint intentions and its accomplishment with joint activities and cooperation among team members.

Several teamwork theories have been proposed in the artificial intelligence literature [CL91, Jen94, Jen95, Tam97]. Being based on the *beliefs-desires-intentions* framework [OJ96], they focus on the development of general models of teamwork to enable team to act coherently, overcoming uncertainties of complex, dynamic environments, where team members often encounter differing, incomplete and possibly inconsistent views of the world and mental state of other agents. Those theories are not intended to be directly implemented, but to be used as a specification for agent design, prescribing general, rather than domain-specific, reasoning processes or heuristics for teamwork. The main requirements of such teamwork models are: (1) to create a framework that enables flexible communication between team members, so that miscoordination is avoided and they act coherently; and (2) to be capable of monitoring performance and flexibly reorganizing and reallocating resources to meet any contingencies caused by unexpected team members faults in fulfilling responsibilities, or discovered unexpected opportunities.

In order to obtain a deeper understanding of the fundamental concepts that underpin agent interactions, Nick Jennings proposed in [Jen93] the concepts of *commitments*, *conventions* and *social conventions*. *Commitments* are pledges made by agents about both actions and beliefs, which can either be about the future or the past. Once an agent commits itself to perform a particular action then, provided that its circumstances do not change, it will endeavor to honor that pledge. To operate successfully and intelligently, agents need general policies for governing the reconsideration of their commitments. *Conventions* describe circumstances under which an agent should reconsider its commitments. They also indicate the appropriate course of action to retain, rectify or abandon these commitments. Although conventions play an important role in multi-agent systems, they are essentially asocial constructs, describing how an agent should monitor its present and future commitments. They do not specify how an agent should behave towards its fellow community members if it alters or modifies its commitments. This is not important for goals that are unrelated to other activities. However, for goals that are inter-dependent, it is important that the relevant acquaintances are informed of any substantial change that affects their processing, if the community is to act in a coherent manner. *Social conventions* specify how agents should behave with respect to the other community members when their commitments alter.

In [CL91], Cohen and Levesque proposed the first prominent teamwork model: the *joint intentions theory*. A rough definition of joint intention is a property that holds a group together in a shared activity. The theory is described in terms of beliefs, goals and mutual beliefs. *Belief* is taken to be what an agent is sure of or a proposition that it takes to be true, after competing opinions and wishful thinking are eliminated. *Goal* is the most desired condition that an agent has chosen as being the most desired among those that it finds accessible. A particular case of a goal is to believe that something is false now and to have a goal that it be true later; this is called an *achievement goal*. *Mutual belief* among members of a group is an infinite conjunction of beliefs about other agent' beliefs about other agent' beliefs (and so on to any depth) about some proposition.

In a seminal work of the same authors of [CL91], it was defined the notions of individual commitment and individual intention. *Individual commitment* is seen as a *persistent goal*:

> *"An agent has a persistent goal relative to q to achieve p iff: (1) it believes that p is currently false, it wants p to be true eventually; (3) it is true that (2) will continue to hold until it comes to believe either that p is true, or that it will never be true, or that q is false."*

Condition q is an irrelevance or escape clause against which the agent has made relative its persistent goal. Frequently, this clause encodes the network of reasons why the agent has adopted the commitment. Individual commitment commits the agent to a given mental state. *Individual intention* constrains the agent's adoption of other mental states, committing it to act in a certain mental state, so as a persistent goal may persist over time:

> *"An agent intends relative to some condition to do an action just in case it has a persistent goal (relative to that condition) of having done the action and, moreover, having done it, believing throughout that it is doing it."*

In [CL91], Cohen and Levesque made a generalization of the persistent goal and intention concepts to the case where a group is supposed to jointly act. *Joint commitment* cannot be simply a version of individual commitment where a team is taken to be the agent, for the reason that the team member may diverge in their beliefs. If an agent finds out a goal is impossible, then it must give up the goal. But when a member of a team finds out a goal is impossible, the team as a whole must again give up the goal, but the team does not necessarily know enough to do so. So, any team member who discovers privately that a goal is impossible (has been achieved, or is irrelevant) should be left with a goal to make this fact known to the team as a whole. Before the joint commitment can be discharged, the agents must in fact arrive at the mutual belief that a termination condition holds. In order to support the achievement of this mutual belief, the notion of joint persistent goal is constructed upon the notion of *weak persistent goal*:

> *"An agent has a weak persistent goal relative to q and with respect to a team to bring about p if either of the following conditions holds: the agent has a normal achievement goal to bring about p, that is, the agent does not yet believe that p is true and has p eventually being true as a goal; the agent believes that p is true, will never be true, or is irrelevant (that is, q is false), but has a goal that the status of p be mutually believed by all the team members."*

The definition of *joint persistent goal* is:

> *"A team of agents have a joint persistent goal relative to q to achieve p just in case: (1) they mutually believe that p is currently false; (2) they*

> *mutually know they all want p to eventually be true; (3) it is true (and mutual knowledge) that until they come to mutually believe either that p is true, that p will never be true, or that q is false, they will continue to mutually believe that they each have p as a weak achievement goal relative to q and with respect to the team."*

This definition states that although members of a team jointly committed to achieve p mutually believed initially that they each have p as an achievement goal, as time passes, the team members cannot conclude about each other that they still have p as an achievement goal, but only that they have it as a weak achievement goal. A corollary of the definition is that if a team has a joint persistent goal to achieve p, then each member has p as an individual persistent goal. Thus, the definition of *joint commitment* is:

> *"If a team is jointly committed to some goal, then under certain conditions, until the team as a whole is finished, if one of the members comes to believe that the goal is finished but that is not yet mutually known, it will be left with an individual persistent goal to make the status of the goal mutually known."*

This definition states that if there is a joint commitment, agents can count on the commitment of the other members, first to the goal in question and then, if necessary, to the mutual belief of the status of the goal. The definition of *joint intention* is:

> *"A team of agents jointly intends, relative to some escape condition, to do an action iff the members have a joint persistent goal relative to that condition of their having done the action and, moreover, having done it mutually believing throughout that they were doing it."*

In [Jen94, Jen95], Nick Jennings implemented multi-agent collaboration based on joint intentions in the domain of electricity transportation management, within the *ARCHON* project. This was likely one of the first implementations in a complex domain based on a general model of teamwork. It was presented a framework denoted as joint responsibility based on a joint commitment to the team's joint goal and a joint recipe commitment to a common recipe. The main modification proposed to the joint intentions framework is to claim necessary two types of joint commitments, because they necessitate different courses of action when joint they are dropped. It's claimed that existence of a joint commitment to a joint goal is not sufficient to guarantee that cooperative problem solving will ensue. It is also necessary a joint commitment to a common recipe, which provides a context for the performance of actions in much the same way as the joint goal guides the objectives of the individuals. When commitment to the joint goal is dropped the joint action is over. However, if the group becomes uncommitted to the common recipe there may still be useful processing to be carried out. For instance, if the recipe is deemed invalid, the agents may try a different sequence of actions that produce the same result.

In [Tam97], Milind Tambe presented an implemented general model of teamwork, called *STEAM* (*Shell for TEAMwork*). This model borrows from previous teamwork frameworks, especially joint intentions. These are the basic building blocks of teamwork and team members build up a complex hierarchical structure of joint intentions, individual indentions and beliefs about others' intentions. *STEAM* differs from previous frameworks, via its focus on teamwork capabilities that arise in domains of more than two-three agents, with more complex team organizational theories, and with practical emphasis on communication costs. The investigation focused on three domains: two domains (*attack* and *transport*) based on a real-world distributed, interactive simulator commercially developed for military training; and, as a third domain, *RoboCup simulator*. *STEAM* defines appropriate speech acts (e.g. *request* and *confirm* speech acts to establish joint commitments or persistent joint goals) to guarantee four fundamental issues: coherence in teamwork; appropriate tradeoff in the amount of information team members must maintain about teammates' activities; not reconciled task allocations; and generalization of communication capabilities to establish and terminate team operators (reactive team plans). The *STEAM's* main innovation is the integration of decision-theoretic communication selectivity, which attempts to follow the most cost-effective method of attaining mutual belief in joint intentions, via a measure of *likelihood that some relevant information may be already mutually believed*. (e.g. likelihood of lack of joint commitments and probability that a fact is not common knowledge). As the estimation of parameters that measure the expected utility may fail, there are some error recovery routines. One of the methods for estimating those parameters is to infer a team's mental state from observations its actions. Some issues that remain open for future work are: investigating interactions with learning, which would enable agents to render automatic routine teamwork activities, rather than always reasoning about them; failure detection and recovery, particularly in environments with unreliable communication; and enriching communication capabilities to form a basis of multi-agent negotiation protocols. For instance, when a joint persistent goal cannot be established if an agent refuses it, negotiations among team agents would ensue.

In [SV99], Stone and Veloso from CMU proposed a flexible team agent structure and a method for inter-agent communication in dynamic, real-time domains with unreliable, single-channel, low-bandwidth communication, where previous teamwork models take too much time or are infeasible due to communication restrictions. The structure of homogeneous agents allows them to capture and reason about team agreements. Collaboration between agents is achieved by the introduction of *formations*. A *formation* decomposes the task space defining a set of *roles*. Agents can flexibly switch roles within formations, and agents can dynamically change formations, in response to changing environments. The proposed team structure was focused in time-critical environments in which agents alternate between periods of limited and unlimited communication. For this purpose, it was introduced *periodic team synchronization* (*PTS*) domains, where agents can periodically synchronize in a safe, full-communication setting, forming off-line agreements for future use in periods in which agents act autonomously with limited or no communication possible. The communication method is designed for use during low-communication periods in *PTS* domains, through *locker-room agreements*. These agreements are formed during the periodic synchronization opportunities and are remembered identically by all agents during those periods with limited or no communication possible. In the proposed team structure (Figure 3.3) ([6]), each agent keeps track of three different types of state: *world state*, *locker-room agreement* and *internal state*. The *world state* reflects the agent's perception of the real world via its sensors and the predicted effects of its actions. The *locker-room agreement* is set by the team when it is able to privately synchronize. The *internal state* stores the agent internal variables (e.g. the agent's role within a team), which reflect previous and current world states, possibly as specified by the locker-room agreement. There are two classes of behaviors: *internal behaviors* update the agent's internal state based on its current internal state, the world state and the team's locker-room agreement; *external behaviors* reference the world and internal states and select the actions to send to the actuators. The architecture was fully implemented and successfully experimented in a robotic soccer simulator.

3.2.2 Negotiation

Negotiation is a key form of interaction that enables groups of agents to arrive at a mutual agreement regarding some belief, goal, or plan [BdL$^+$99]. The process of negotiating may be of many different forms, such as auctions, protocols like contract nets and argumentation. There are three broad topics for research on negotiation: first, *negotiation protocols* are the set of rules that govern the interaction; second, *negotiation objects* are the range

[6] Figure reproduced from [SV99].

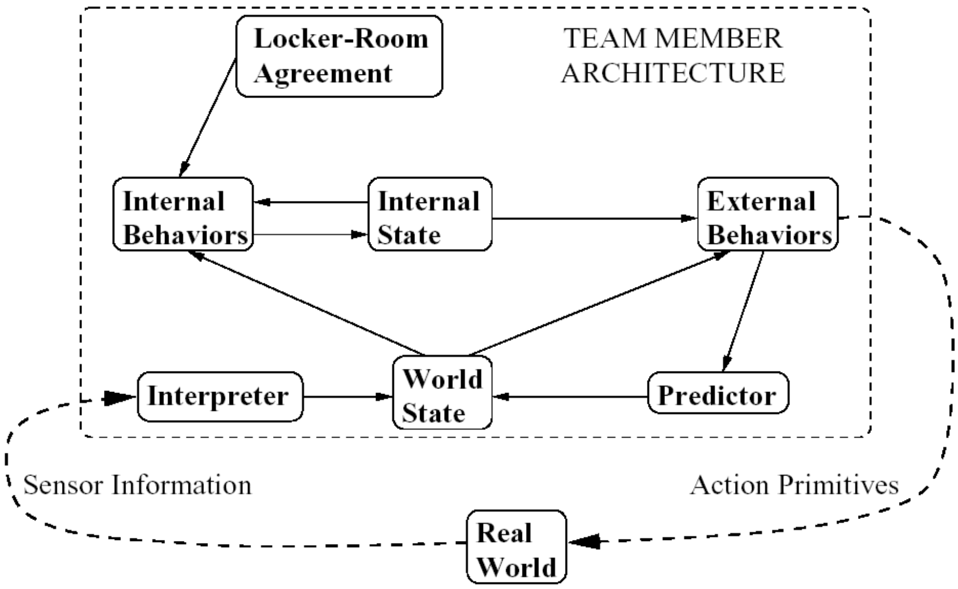

Figure 3.3: Functional model of a teamwork structure for *periodic team synchronization* domains.

of issues over which agreement must be reached (e.g. price, quality, timing, allowable operations, etc.); third, *reasoning models* provide the decision making apparatus by which participants attempt to achieve their objectives.

One of the most studied and influential protocols for negotiation is the *contract-net protocol* [Smi80]. It was inspired by contracting processes in human organizations and it offers a structure that assists the system designer in deciding what the interacting agents should say to each other, rather than how to say it. Agents coordinate their activities through contracts to accomplish specific goals. The execution of a task is dealt with as a contract between two agents. In a contract net (a network of distributed and loosely coupled agents), an agent can take on one of two roles: manager or contractor. A *manager* is responsible for monitoring the execution of a task and processing the results of its execution. A *contractor* is responsible for the actual execution of the task. Individual agents are not designated a priori as managers or contractors and they can take on either role dynamically during the cooperative problem solving. Typically, an agent will take on both roles simultaneously for different contracts. An agent, acting as a manager, decomposes its contract (the task or problem it was assigned with) into subcontracts to be accomplished by other potential contractor agents. For each subcontract, the manager *announces* a task to the network of agents with a deadline to receive proposals from potential contractors. Agents receive and evaluate the announcement. Agents that are eligible to be assigned to the announced task (that have appropriate resources to accomplish the task) reply to the manager with *bids*, which indicate their ability to achieve the tasks, based on a given performance criteria defined by the manager (e.g. proximity, number of resources, etc.). The manager evaluates the bids it has received and *awards* the task to the most suitable agent and nominates it as the contractor for that task. Finally, manager and contractor exchange information together during the accomplishment of the task.

3.3 Application domains of cooperative systems

The application domain of a cooperative system has strong influence on its design requirements. There are some tasks that can be accomplished with one agent alone and do not strictly require explicit cooperation among several agents, although better performance can be attained if they are executed in parallel [LH97]. Examples of these tasks are painting a wide wall, cleaning a room, searching for objects in a wide area (foraging), flocking, etc. [AMI89, Ark92, HB92, Mat92b, ABN93, BA94, KZ94, BDG99]. In these tasks, if all agents work independently and are unaware of the existence of other agents, the system is a simple extension of a single agent solution. This kind of weak cooperation, without an explicit coordination mechanism, is simply an emergent property.

In [BA94], Balch et al. proposed the *speedup measure*, which reveals how much more efficient several robots are than just one in completing a task. If $P(i)$ is the performance for i robots ([7]), speedup measure is given by Equation 3.1 ([8]). If speedup measure is equal to 1.0, i robots complete the task exactly i times faster than one robot. This is called *linear improvement* (performance proportional to the number of robots). Speedup values less than 1.0 reflect *sub-linear performance* and values greater than 1.0 reflect *super-linear performance.*

$$speedup = \frac{P(1)}{i \cdot P(i)} \tag{3.1}$$

[7] In this definition, it is assumed that, for a given performance criteria, higher values of the function P reflect better performance.

[8] This definition was reproduced from [BA94].

When agents execute similar tasks in parallel, working independently and without a coordination mechanism (emergent cooperation), the system performance depends on whether exist resources contention (traffic contention, shared tools, etc.) and goal conflicts (interference) or not. In the former case, the performance speedup will be sub-linear, whereas in the latter case the performance will be linearly speedup. In the former case, efficient resource contention mechanisms or coordination mechanisms to avoid interference should be used to attain better speedup performance.

However, there are many real world tasks that require explicit cooperation among agents and cannot be accomplished by any single agent alone, such as pushing a heavy or large box, multi-target observation and exploration. Since these tasks require tight coordination and strong cooperation among agents, the achievement of tasks will be accidental and very unlikely if agents work independently. The box-pushing task [Par94, MNS95, RDJ95, GM00] requires tight coordination among robotic agents and has analogies with other practical problems, e.g., storage and retrieval or truck loading and unloading. Multi-target observation consists in maximizing the time during which moving targets are observed by, at least, one of the robotic agents [WM00, Tou00, Par02]. Multi-target observation task is similar to the foraging task with the addition of dynamic targets that must be continuously tracked. It has many similarities with security, surveillance and recognition problems. Exploration groups different tasks regarding team members moving around in the environment. It includes flocking [Mat94], maintaining formations [BA98, BH00], bounding overwatch [Par94] and map building [AB98b]. In [LH97], Lin et al. defined the object-sorting task, which is a combination of foraging and cooperative transport. This task abstracts parallelism and cooperation in just one task and it was used to study cooperation protocols.

Robotics competitions, and especially *RoboCup*, have recently given a significant boost to research work on multi-robot systems [Tam97, LVAC99, SV99, SV00, INS01]. *RoboCup* [KANM98] is an initiative of the RoboCup Federation, which has been promoting intelligent robotics research, by providing robot soccer competitions as a common task for evaluation of various theories, algorithms and agent architectures. With this purposes, it has been organizing regular international robot soccer competitions since 1997, along different soccer leagues: small-size league (small and fast-moving robots with limited capabilities), middle-size league, humanoids and simulator league (a virtual soccer environment with a high degree of realism). Robotic soccer is a challenging testbed for research in multi-agent and multi-robot cooperation in a highly dynamic and uncertain environment. *RoboCup* also promotes other domains than soccer, such as search and rescue or exploration in dangerous terrains, e.g., the simulation of a rescue operation performed by robots within a catastrophic scenario.

3.4 Taxonomy of group architectures

The group architecture of a cooperative system provides the infrastructure upon which collective behaviors are implemented and a principled way of organizing a control system. Besides providing a structure, an architecture imposes constraints on the way the control problem can be solved [Mat92a, Mat99]. In [CFK97], Cao et al. propose a classification of the group architecture for cooperative systems through four characteristics: *centralization* vs. *decentralization*, *differentiation*, *communication structures* and *modeling of other agents*.

3.4.1 Centralization vs. decentralization

The most fundamental decision that is made when defining a group architecture is whether the system is *centralized* or *decentralized* and, if it is decentralized, whether the system is *hierarchical* or *distributed* [Bot00]. This characteristic of the group architecture defines the nature of the decision making process and how the system answers to questions like: What agent decides what task shall be executed, when there are several agents and missions in common? Does each agent decide autonomously about its own role in the system? Does the system have some agents that are more specialized for taking such decisions? Although many practical systems do not conform to a strict dichotomy between centralized and decentralized control, it is interesting to note the advantages and disadvantages of both approaches.

Centralized architectures are characterized by having a single control agent that is individually responsible for the decision-making process. It is presupposed that the central process has a global model of the world that enables it to produce, theoretically, optimal solutions for the problems. We may consider two different subtypes of centralized systems: *wholly centralized* and *partially centralized*. In a *wholly centralized* group of agents, each agent receives its future actions from a central processor (agent) and transmits to it local information [BB97]. Apart from the central agent, other agents have not any control autonomy. In a *partially centralized* system, there is also a central agent, but other agents can also take some autonomous local decisions. Apart the central agent, other agents act locally as central agents and generate local and partial plans, which reduce the planning state space complexity. Then, the central agent tries to conciliate the local plans so as to maximize their interaction towards global utility [ER94, ER95, AS97]. Centralized systems may lead to optimal, coherent and comprehensive solutions, but they have numerous shortcomings. Depending on the group dimension, it is very difficult, or even impracticable, to have and maintain a global model of the world on a single agent, based on local and potentially inconsistent views among the local agents. Furthermore, it tends to be a costly solution on time and resources, as it uses massive communication between the local agents and the central agent, which may become a severe communication bottleneck [Bot00]. It has also a high design complexity and low reliability, as all the intelligence is concentrated on a central agent. Moreover, for tasks with NP complexity, the centralized approach is impracticable due to the dimension of the search space [LH97].

Decentralized architectures are composed of a network of logically and physically independent agents. Each agent is able to reason about plans and to decide its own actions [OJ96] and views the system dynamics as being determined by the interactions with and among other agents. Decentralized systems may be classified, by the degree of autonomy conferred to each individual agent, as hierarchic or distributed systems [CFK97]. *Decentralized hierarchical* systems can be viewed as locally centralized systems, where there is a hierarchy of "central agents". Decisions are distributed across different hierarchical levels and there is a coordinator agent at each level, which can be viewed as a central agent that produces plans for the agents in lower hierarchical levels. Those plans are conciliated through a negotiation involving the coordinator agents [LH97]. *Decentralized distributed* systems endow all the agents with the same decision power and autonomy. Each agent produces autonomously its own plans as a function of its own goals and ensures coherent behavior through a coordination model [Ben88, Jen96]. In such decentralized systems, each agent is endowed with abilities that enable it perceiving the environment, reasoning about a complex task, taking decisions to accomplish that task and executing plans. Decentralized architectures have several recognized advantages over centralized architectures, including fault tolerance, reliability, robustness, natural exploitation of parallelism flexibility and scalability [CFK97]. As there is no a central controller, these systems with distributed control and data exhibit graceful degradation of performance and better robustness than centralized systems. As the role of each agent may change with context, these systems may be very flexible. However, distributing control and data means that knowledge of the system's overall state is dispersed throughout several entities and each individual has only a partial, incomplete and imprecise perspective. Thus, decentralized systems present an increased degree of uncertainty, making more difficult to attain coherent global behavior. Also, if there is no efficient coordination, the dynamics of such systems can become extremely complex, giving rise to nonlinear oscillations and chaos [Jen96].

In decentralized systems, *coordination* is the key for achieve coherent global behavior. It can be defined as the process by which an agent reasons about its local actions and the (anticipated) actions of others to try and ensure the community acts in a coherent manner [Jen96]. There are three main reasons why coordination is necessary: (1) because there are dependencies between agents' actions (local decisions have an impact on other agents and there is the possibility of harmful interactions); (2) because there is a need to meet global constraints; and (3) because no one individual has sufficient capacity, resources or information to solve the entire problem. The main objectives of the coordination process are to ensure: that all necessary portions of the overall problem are included in the activities of at least one agent; that agents interact in a manner which permits their activities to be developed and integrated into an overall solution; that team members act in a purposeful and consistent manner; and that all of these objectives are achievable within the available computational and resource limitations.

3.4.2 Differentiation

Another important characteristic of group architectures is *differentiation*. Differentiation is frequently dichotomized between *homogeneous* and *heterogeneous groups*. A group is *homogeneous* if the capabilities of the individual agents are identical, and *heterogeneous* otherwise. The degree of differentiation of a system has direct implications on the cooperation requirements of its group architecture.

In [Par94], Parker has introduced the *task coverage metric* to give a measure of the number of capabilities on a team that may allow some team member to achieve a given task. Consider a team of n robots $R = \{r_1, r_2, \cdots, r_n\}$ and a mission composed of m independent subtasks represented by the set $T = \{task_1, task_2, \cdots, task_m\}$. Consider also the set $A_i = \{a_{i1}, a_{i2}, \cdots\}$ of the high-level task-achieving functions possessed by the robot r_i and the set of n functions $\{h_1(a_{1k}), h_2(a_{2k}), \cdots, h_n(a_{nk})\}$, where $h_i(a_{ik})$ denotes the task in T that robot r_i is working on when it activates the behavior set a_{ik}. The task coverage is given by Equation 3.2 ([9]). Interestingly, if the team members are homogeneous, the task coverage is a positive multiple of the number of robots n. The task coverage index must be more or equal than one for all the tasks of its set T, in order to the system have some likelihood to be

[9]This definition was reproduced from [Par94].

able to carry out the mission through efficient cooperation among the team members. Greater values than one for that index indicate the existence of some redundant resources and overlap in capabilities, thus increasing the reliability and robustness of the team amidst individual failures. Task coverage is maximal in homogeneous teams and decreases as groups become more heterogeneous, towards a minimum limit where the index equals one for all the elements of set T. The degree of differentiation measured by task coverage may be interpreted as an index of demand for cooperation. When task differentiation is low, tasks can be accomplished without much cooperation, but otherwise cooperation is mandatory in order to accomplish a mission. In general, heterogeneity introduces control complexity, because greater differentiation requires a more effective cooperative task allocation and a greater need to model other individuals.

$$task\_coverage(task_k) = \sum_{i=1}^{n} \sum_{j} \left\{ \begin{array}{ll} 1 & \text{if } (h_i(a_{ij})) = task_k \\ 0 & \text{otherwise} \end{array} \right\} \tag{3.2}$$

However, it is nearly impossible in practice to build a truly homogeneous robot team, due to differences in sensor tuning, calibration, etc. Moreover, besides those physical deviations, several copies of the same model of robot can vary widely in its behavioral aspects if it is endowed with learning and adaptation capabilities, which is specially vital in less structured and outdoor environments. This means that heterogeneity is present in multi-robot teams whether we like it or not. In [Bal98], Balch investigated the impact of diversity, specially behavioral difference, on performance of multi-robot teams, and conversely, the impact of other task factors on diversity. Thus, the degree of heterogeneity is treated as a result rather than an initial condition. In order to address a quantitative comparison of heterogeneity, a quantitative metric of diversity was proposed, denoted as *hierarchical social entropy*, which provides a continuous scale of diversity [Bal98, Bal00]. This metric is based on the following observation: the measured diversity of a multi-agent society depends on the number of homogeneous subsets it contains and the proportion of agents in each subset. It adapts Shannon's measure of information uncertainty to a measure of societal diversity and uses some notions from biological literature related with numerical taxonomy, like maximum taxonomic distance and clustering methods, to provide a hierarchical metric of distribution of elements in a diversity classification space. Figure 3.4 ([10])

[10]Figure reproduced from [Bal98].

depicts an example of the computation of the metric for three different societies. When the metric is applied to a given system, a dendrogram ([11]) is created in a continuous scale of diversity (behavioral difference or taxonomic level), usually normalized between 0 and 1, to provide an orderly hierarchical view of the classification. Integrating the diversity across all taxonomic levels produces an overall measure of diversity for the system.

In [Bal00], Balch described the application of the social entropy metric to experiment reinforcement learning schemes with different reward functions, including both individual rewards and group rewards upon delivery or progressive rewards as task gets accomplished. Reinforcement learning was used to associate actions with state. The team diversity was measured and correlated with performance for two tasks: soccer and multi-foraging. It was found that local rewards lead to greater homogeneity in both domains. In soccer, higher diversity is associated with higher performance, whereas in multi-foraging, higher diversity is associated with lower performance. The relationship between diversity and performance in soccer (positive correlation) is exactly opposite the relationship in multi-foraging (negative correlation). Although there are no obvious reasons for this difference, it is believed that they are due to differences in the task. Soccer is unavoidably a team activity, while foraging can be accomplished by an individual agent. It is also believed that when the task requires multiple agents, it is more likely that the team benefits from diversity. Greater homogeneity with local reinforcement is due to the fact that individuals are rewarded for their own actions, thus making reinforcement of the same state/action pair more likely in different agents than with global reinforcement.

3.4.3 Communication structures

The *communication structure* of a group architecture is another important characteristic of a group architecture, because it determines the possible modes of inter-agent interaction, as well as the ability of agents to model successfully other agents' mental states (denoted in the literature as *awareness*). In [CFK97], Cao et al. characterize three major types of interactions that can be supported: *interaction via environment*, *interaction via sensing* and *interaction via communications*. In the two first types communication

[11] A *dendrogram* is a taxonomic tree which is frequently used in biology to classify organisms and groups of organisms, at various levels. At the lowest level, organisms are more likely to be classified together (e.g. gorillas and humans are both primates but not canines), but at higher levels, more diverse organisms are grouped together through adequate clustering methods, due to some common characteristic or similarities (e.g. primates and canines are grouped in the class of mammals).

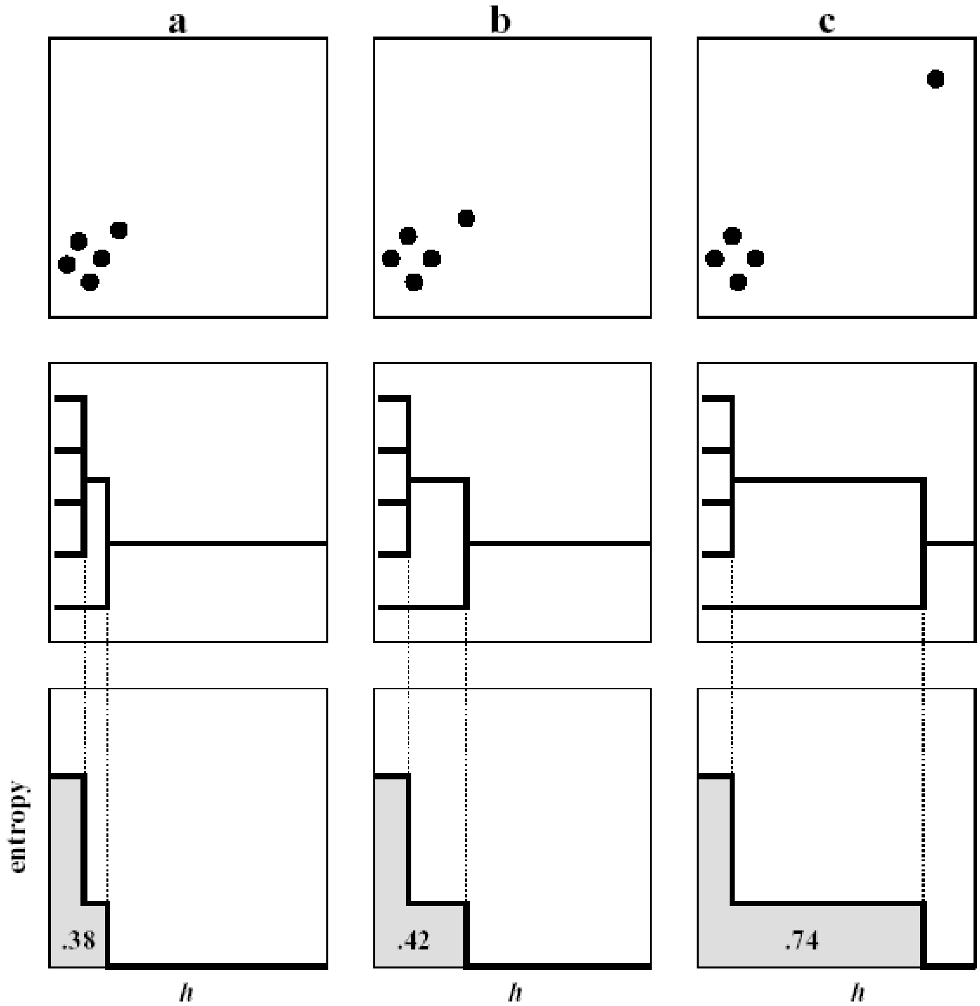

Figure 3.4: *Hierarchical social entropy* (bottom) is computed for three societies (top). As the element on the upper right is positioned further away from the group in the classification space, the overall measure of diversity increases from 0.38 to 0.74. Dendrograms for the groups are also displayed (middle row).

is implicit, whereas in the latter one it is explicit.

Interaction via environment is the simplest and the most limited type of interaction. In [Par95], Parker denotes it as *cooperation through the world.* It occurs when the environment itself is the communication medium (a kind of shared memory or broadcast implicit communication) and there is no explicit communication or interaction between agents. In this case, agents simply sense the effects of other teammates' actions on the world. Some authors denote this approach as *stigmergy. Stigmergy* stores state in the environment, so that specialized sensors can easily retrieve it (e.g. pheromones in nature and obstacle detection in multi-robot systems) [BHD94]. This is an appealing, reliable and robust approach because of its simplicity and its lack of dependence upon explicit communication channels and protocols, which may be fallible and have limited bandwidth. However, it is limited by the extent to which an agent's sensation (e.g. a robot's sensation) of the environment reflects the salient states of the mission the team must accomplish and the effects produced on the environment by the other teammates.

Interaction via sensing, also denoted as *passive action recognition* in [Par95], occurs when an agent knowingly uses its sensing capabilities to observe the actions of its teammates that are within its sensing range. The sensory information is then used to recognize its teammates' actions, goals and plans, through appropriate modeling of other agents and perception. As the agent can only observe the nearby agents, this is a kind of local communication mechanism. Like interaction via environment, this mechanism is appealing because of its lack of dependence upon explicit communication channels and protocols, which may be fallible and have limited bandwidth. However, it is limited by the degree to which an agent can successfully interpret its sensory information, as well as the difficulty of analyzing the actions of other agents and use that information to infer their mental state.

Interaction via communications (explicit communication) is appealing because of its directness and the ease with which agents can become aware of the actions and/or goals of the other agents, giving access to both local and global information. However, it has poorer fault tolerance and reliability than implicit communication mechanisms, in that it can be highly dependent upon the presence of a reliable communication channel for the successful accomplishment of a cooperative mission. Moreover, as it also depends on the communication channel bandwidth, it has worst scalability and extensibility than implicit interactions, because it is not possible to scale to more agents without additional communications overhead. In [JZ00], Jung and Zelinsky give some biologic examples of three possible types of representation in communicated signals: *iconic, indexical* and *symbolic. Iconic* representation is by similarity to what it represents, e.g. an orange disc painted on a cave wall may represent the sun. *Indexical* representation correlates or associate icons, e.g. some animals have learnt to correlate the icon of smoke with that of fire. *Symbolic* representation is a relationship between icons, indexes and other symbols, representing a higher-level pattern underlying sets of relationships. Some authors claim that language can be represented as a symbolic hierarchy. A *grounded symbol* requires following all the references, which may be icons, indexes or other symbols, that are necessary to interpret it. In general, symbolic communication between two agents require: some iconic references in common; either a shared grounding for some indexical representations, a common process that develops shared indexical groundings, or a combination of both (e.g. learning to correlate smoke with fire); a common process that develops shared symbolic groundings (e.g. language development); and a mechanism for learning new symbols by communicating known ones (e.g. learning through metaphor).

As mentioned previously, implicit mechanisms are common in some biologic systems, such as bacteria and insect societies, whereas explicit communication appears in more complex animals, especially primates and humans. Implicit interaction mechanisms are mainly suited to local rules based control, while explicit communication is more suited to global control approaches, eventually based on planning and negotiation. In [Par93], Parker discusses principles for determining the proper balance between local and global control, which determine the necessary communication mechanisms' requirements. In practice, there is a continuum between strictly global and strictly local control laws.

Global control laws utilize the global goals of the cooperative team and/or global knowledge about the team's current or upcoming actions to direct an individual agent's actions. The global goals of a team indicate the overall mission that the team is required to accomplish. They can be imposed by a central controller (e.g. a human), by one of the members of the team, or through planning and negotiation among the team members. Global knowledge refers to additional information that is normally not available to individual agents through their local sensors, but that may be necessary for the cooperative team to achieve the global goals. This information typically indicates what other agents in the team are doing or going to do (state information), or what they sense locally. The use of global goals and information enables the implementation of explicit models of cooperative teamwork, eventually more efficient, but it also has some shortcomings. Adequate global information may not be available to achieve the desired global goal. Even with comprehensive global knowledge, an agent may still not exhibit optimal global behavior unless it utilizes all of the available knowledge. Moreover, besides the cost of maintaining this global knowledge across the members of the team, processing it requires time and resources, which are typically scarce in real applications. If the global goals and knowledge change often enough, the agent may not be able to act upon the global knowledge before it becomes out-of-date, or simply may not be possible or viable to maintain real-time global information, due to the limited bandwidth of communication channel [Par93].

On the other hand, *local control* laws guide an agent's actions based on the nearby environment of that agent. Such information is usually obtained through implicit communication, using the agent's sensory capabilities, reflecting the state of the world near the agent. Local control laws allow agents to react to dynamic changes in their environment without relying on preconceived plans or expectations of the world. If local rules of individual agents are carefully designed, global functionality may emerge from their interaction. However, certain global goals cannot be attained through the use of local control laws alone, because those aspects of global goals that have no physical manifestation in the world cannot be acted upon by local control laws [Par93].

In [Par93], Parker described the simulation of several control strategies along the local versus global spectrum in a keep formation experiment. This task consists on a group of agents to stay in formation with one another, while the leader of the group follows a pre-specified route and while agents avoid obstacles as they appear. It was simulated four control strategies: local control alone; local control augmented by a global goal of the group; local control augmented by a global goal and partial global information; and local control augmented by a global goal and more complete global information. Following the conclusions of this simulation study, the following general principles and guidelines were proposed. If the global goals are known at design time, and all information required for an agent to act consistently with the global goals can be sensed locally by the agent at run-time, these goals can be designed into the agents. The more static, reliable, complete known, and easy to use the global knowledge is, the more practical its use in a global control law. Conversely, the more unknown the global information, the more dependence the team will have on local control, combined with behavioral and environmental analysis to approximate global knowledge. Behavioral analysis may provide a suitable approximation to global knowledge, being particularly useful when the agents possess a fixed set of discernible or communicable actions. In many applications, particularly those in which accomplishing the task is more important than how the agents accomplish the task, local control may provide a suitable approximation to the optimal group behavior, eliminating the need for the use of global knowledge. Global knowledge should be used to provide general guidance for the longer-term actions of an agent, whereas local knowledge indicates the more short-term, reactive actions the agent should perform within the scope of longer-term goals.

In [Ark92], Arkin demonstrated that sometimes cooperation between robotic agents was possible even in the absence of communication, however this is a weak form of cooperation and it may me very inefficient. In [ABN93], Arkin et al. described a research work that has as main goal to create a foundation theory to specify, for a given task, the most reliable, efficient and robust means of interaction between robots. It was described a simulation study of a multi-agent foraging task. It was found that inter-robot communication of state improves performance. In [BA94], Balch et al. presented the continuation of the previous work. It was described a number of simulation experiments of three tasks: forage task, in which an agent wanders about the environment looking for items and then attaches and returns them to a specified home position; consume task, in which an agent wanders about the environment to find items, attaches them and then performs work there; and graze task, in which the agents must completely cover or visit the environment. Simulations were constructed using different levels of communication, including no communication, state communication (information concerning the internal state of agents) and goal communication (specific goal-oriented information). These three levels corresponded to the three types of interaction referred previously (via environment, via sensing and via communication, respectively). In this particular experiments, goal communication was considered more complex than state communication, because the former type was implemented as an interaction via sensing, while the latter one was implemented via explicit communication. The general findings of this work were: (1) communication improves performance significantly in tasks with little implicit communication (e.g. forage task); (2) communication appears unnecessary in tasks for which implicit communication exists (e.g. consume and graze tasks); (3) more complex communication strategies offer little benefit over basic communications.

3.4.4 Awareness and modeling of other agents

Awareness capabilities of individual agents means their capacity to model other teammates' beliefs, goals, states and actions, by using different communication structures. The awareness level of the team members has strong influence on whether the group architecture is more reactive or deliberative (explicitly coordinated). One way to provide agents with sufficient awareness is developing sophisticated perception abilities, relying mostly on implicit communication and minimizing the need for explicit communications among team members. On the other hand, if an agent has limited perception abilities, it can be aware of other agents by constructing a world model mostly based on information explicitly communicated among team members, through a given communication channel.

There have been some studies about the correlation between performance and *awareness* in cooperative multi-robot tasks [Mat92b, Par95, Tou00]. The main challenge of multi-robot cooperation is to overcome interference and to achieve at least linear (break-even point), or preferably super-linear, improvement in efficiency [Mat92b] (see page 37 for a reference to the *speedup measure*).

In [Mat92b], Mataric addressed the problem of distributing a task over a collection of homogeneous mobile robots, in a homing task. It was applied a distributed control approach and different communication constraints. The main goal was to explore the interaction between computation and dynamics of the individual robots of a collective, taking the most advantage of the dynamics. The robots either acted in ignorance of one another (no awareness), informed coexistence, or intelligently cooperating with one another. If robots have no awareness, they behave as they were the only existing robots in the environment, i.e. all perceivable objects, not related with the task, are classified as obstacles (including other robots). In the informed coexistence case, the robots have the ability to sense each other, discriminating obstacles from other obstacles. In the latter case, each robot has a virtual sensor that provides a measure of the local population density and the population gradient. It was experimentally demonstrated that the ability to distinguish other robots from the rest of other objects in the worlds (increasing awareness) provides sufficient power to overcome interference, because trading off individual autonomy for collective behavior renders better efficiency than individual greedy strategies.

In [Par95], Parker investigated how the extent to which robot team members are aware of, or recognize the actions of their teammates, and the extent to which they use this information to effect their own actions, has impact on the cooperative team performance. With this purpose, it was performed some experiments with collectives whose members could and could not be aware of other collective members. Those experiments performed a puck-moving mission, varying the number of robots (redundancy) and the level of awareness the robots had of the actions of their teammates. Some conclusions that were extracted are: (1) according to an energy metric of performance, performance is improved with awareness, regardless of the team size, because replicated actions are prevented; and (2) redundancy may be more important than awareness, if a significant part of the mission consists of tasks whose effects can be sensed through the world.

In [Tou00], Touzet investigated awareness in the context of learning.

Learning involves the exploration of the search space to gather information about the task, and exploitation of the data, usually through generalization. The main restriction to the use of learning comes from the size of the search space, which increases almost exponentially in the number of team members, if it is used high awareness and/or explicit communication. Awareness of other robots implies the addition of several dimensions to the search space. It was investigated the impact on the search space size of cooperation mechanisms with various levels of awareness, through a cooperative multi-robot observation of multiple moving targets. Each level of awareness was evaluated by the number of inputs on each robot as a function of the number of robots, under the assumption that all robots have at least n sensor inputs (lower level awareness). For the higher-level awareness (complete awareness), each robot has $(n \cdot N)$ inputs when the team has N different robots (each robot is provided with the n inputs of each robot in the team). The preferable situation would be able to provide awareness independently of the number of robots, so as the learning team scalability would be guaranteed. In this case, the number of inputs provided to each robot would be $n + \delta$, with $\delta < N$, where δ represents the knowledge about all the other members of the group. Although it was not specified how to obtain such knowledge, it was proposed a generic method for estimating δ, as a function of the maximum number of neighbor robots, which depends on the workspace area, awareness range and robot policies.

Chapter 4

Cooperative multi-robot systems architectures

Robotics researchers are faced with the task of engineering machines that gather information about their world via sensors, reason and effect action via actuators [Jun98]. This definition led to the classical Artificial Intelligence approach to robotics, known as *sense-plan-act* paradigm (Figure 4.1) ([1]). In the 1980s, many robotics researchers began to realize that this approach failed to be scalable to real environments. A control system for a completely autonomous mobile robot must perform many complex information-processing tasks in real time. Moreover, if that robot is designed to act in a dynamic and complex environment, it must process sensory information change rapidly [Bro86]. The *sense-plan-act* paradigm decomposes such control problem into a series of functional units, illustrated by a series of vertical slices in top of Figure 4.2 ([2]). After analyzing the computational requirements for a mobile robot and recognizing that an iteration of the sense-plan-act cycle resulted in response times that were long for many robotic tasks, Brooks proposed in [Bro86] a different decomposition of the problem (bottom of Figure 4.2), which has been denoted as *behavior-based control*. Those requirements were:

- *Multiple Goals* – often a robot will have multiple goals, which may be conflicting;
- *Multiple Sensors* – a robot will likely to have multiple sensors that are not error-prune and some of them may lead to inconsistent readings;
- *Robustness* – when some sensors fail, the robot should be able to adapt and cope by relying on those still functional;

[1]Figure reproduced from [Jun98].
[2]Figure reproduced from [Bro86].

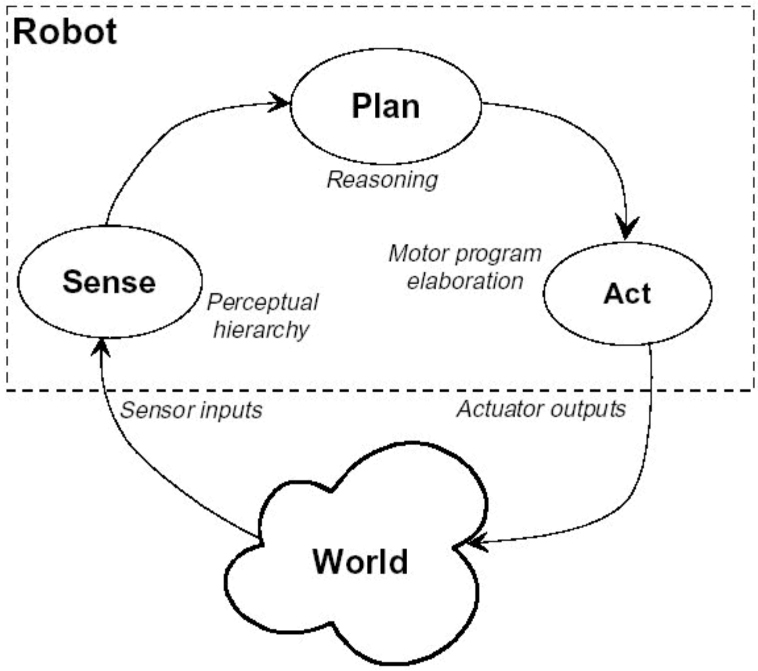

Figure 4.1: The *sense-plan-act* paradigm.

- *Extensibility* – it should be possible to add more sensors and capabilities to a robot, without need to completely rebuild its controller (e.g. more processing power to support processing information from new sensors).

Based on this vertical decomposition of the problem, Brooks proposed in [Bro86] the *subsumption architecture* (Figure 4.3) ([3]). This architecture presents some interesting properties, such as:

- *Multiple goals and sensors* – it is a functional decomposition that implements in each layer a level of competence, with higher level layers subsuming the roles of lower level layers when they wish to take control, or they execute concurrently;

- *Robustness* – failure of a higher level behavior (probably more complex) does not mean that robot fails to execute its task and robot continues to execute it at a lower level of competence;

- *Extensibility* – the functional decomposition allows to add new layers of control to an existing set, which can be debugged without disrupting the functioning of the lower ones that have been already well debugged.

[3]Figure reproduced from [Bro86].

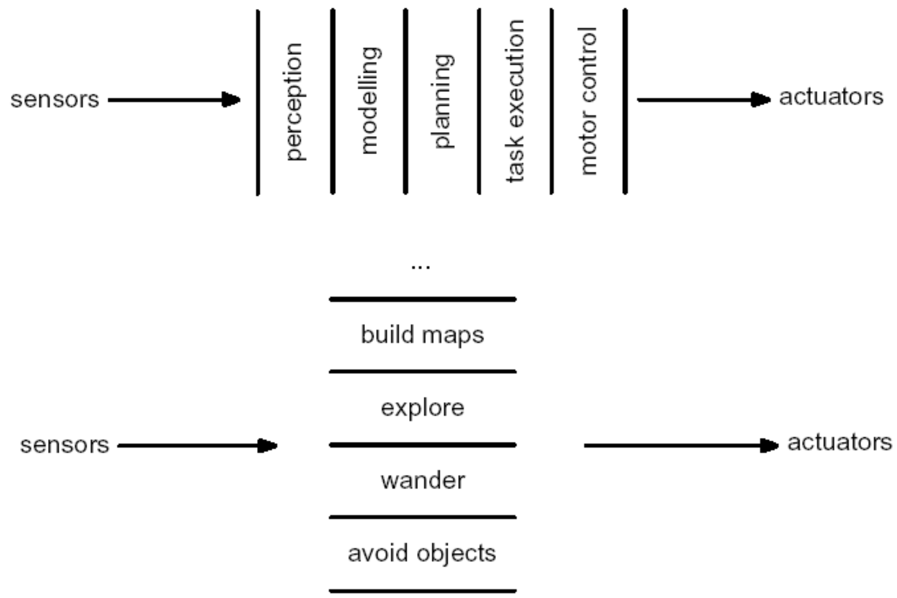

Figure 4.2: Classical Artificial Intelligence (top) vs. *behavior-based* (bottom) decomposition.

In [Mat92a, Mat99], Matarić defined the three basic types of control architectures: *purely reactive*, *behavior-based* and *planner-based*. *Purely reactive* systems, typically encountered in swarm cooperation, achieve rapid real-time responses by embedding the robot's controller in a collection of preprogrammed, concurrent condition-action rules with minimal internal state. Such reactive systems are limited by their lack of internal state, which makes them incapable of using internal representations and learning new behaviors. Although *behavior-based* systems, (following the *subsumption architecture*) are also developed bottom-up, they overcome the reactive systems limitations, because they can store state through its underlying unit of representation: behavior. *Planner-based* systems, usually following the *sense-plan-act* paradigm, are top-down and require the robot to perform a sequence of processing *sense-plan-act* steps, which sometimes compromises their application on real-time applications.

Any control architecture must answer the "what I do next?" question, which is known as the *action selection problem.* In *behavior-based systems*, this is known as the *behavior arbitration problem* or as *behavior coordination* [PM00]. This is a resource allocation problem because there is usually limited available time, energy, computation, sensors and actuators. There have been many different *action selection mechanisms* employed in robot control

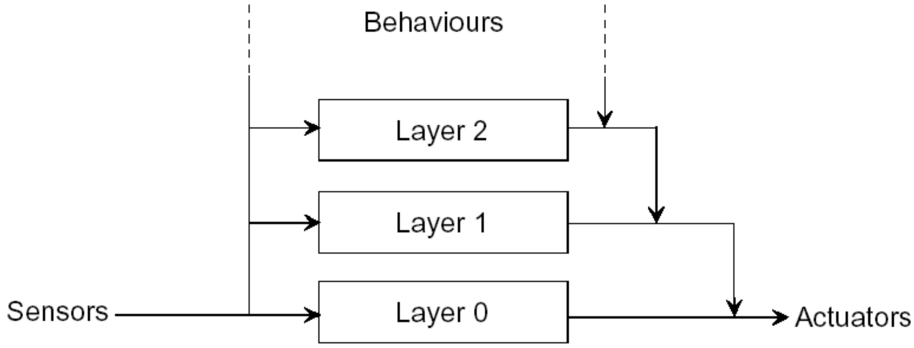

Figure 4.3: The *subsumption architecture*.

systems, which can be divided on *arbitration* and *command fusion* (Figure 4.4) ([4]). While in the former case a behavior is selected from a group of competing ones and give it ultimate control of the system until the next selection cycle, in the latter case, mechanisms combine recommendations from multiple behaviors to form a control action that represents their consensus (cooperative coordination). *Priority-based arbitration* ensures that only one behavior is active at any given time: the one with the highest priority. *State-based* mechanisms include the discrete event systems formalism and Bayesian decision theory. In *winner-take-all* mechanisms, action selection results from the interaction of a set of distributed behaviors until one wins and takes control of the system. *Voting* techniques interpret the output of each behavior as votes for or against possible actions and the action with the maximum weighted sum of votes is selected. *Superposition* techniques combine behavior recommendations using linear combinations (e.g. *potential-fields* [Kha86] and *motor-schemas* [Ark89]). *Fuzzy* mechanisms have several contact points with voting techniques, but notions of fuzzy logic (e.g. inference) are used to formalize the action selection processes and enable new mechanisms like *context-depending blending* [Saf97, SR01], which allow for weighted combination of behaviors. *Multiple objective* approaches provide a formal approach based on multiple objective decision theory [PM00].

Since fostering cooperation among robots necessarily requires adequate and effective control of each individual robot, the single-robot control frameworks referred above naturally influence the cooperative multi-robot architectures that have been proposed in the last decade. Moreover, those multi-robot architectures can be broadly categorized on swarm (reactive) cooperation and explicit cooperation.

[4]Figure reproduced from [PM00].

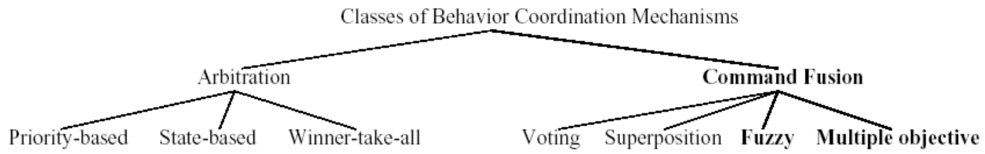

Figure 4.4: Classes of behavior coordination mechanisms.

4.1 Some representative swarm-based case studies

In [Ben88], Beni proposed the concept of *cellular robotic systems* (*CRS*). This is a system composed of a large (but finite) number of simple robotic units capable of accomplishing, collectively, relatively complex tasks through cooperation. Its main characteristics are reliability and the ability to self-organize and self-repair. The robotic units are autonomous because there is no central controller, nor shared memory, nor synchronous clock. A *swarm* is a distributed system inspired on *CRS*, with a large number of autonomous robots, usually with no differentiation among members. Self-organization in a *swarm* is the ability to adequately distribute itself for a given task, e.g., via geometric pattern formation or structural organization. Interaction takes place by each cell reacting to the state of its nearest neighbors. Mechanisms for self-organization have been studied in different contexts, such as large-scale displays and distributed sensing [HB91, HB92, JLB94].

CEBOT (*CEllular roBOTics System*) [UF93a, UF93b] is a decentralized, hierarchical architecture, based on the concept of *cellular robotic systems* [Ben88], and inspired by the cellular organization of biological entities. *CEBOT* is dynamically reconfigurable, because basic autonomous cells (e.g. robots) dynamically reconfigure their structure to an adequate configuration in response to changing environments. *CEBOT's* hierarchy has master cells that coordinate subtasks and communicate with other master cells. In [UF93b], Ueyama and Fukuda studied the formation of structured cellular modules from a population of initially separated cells. *CEBOT's* communications requirements have been extensively studied and various methods have been proposed that seek to reduce communication requirements, by increasing the awareness level of individual cells, i.e. enabling them to model the behavior of other cells (e.g. [FS94]). Under the influence of the *subsumption architecture* [Bro86], Cai et al. proposed in [CFA+95] a new behavior selection mechanism, based on two matrixes: the priority matrix and the interest relation matrix.

One of the popular domains for the study of cooperative behavior in distributed artificial intelligence is the pursuit game or the predator-prey game, in which the predators try to capture the prey or surround it so that it cannot move anywhere. In [Kor92], Korf investigated a simple solution to the pursuit game, based on an attraction force to the prey and a repulsive force from the other predators. The main conclusion of this work is that explicit cooperation is rarely necessary or useful in the pursuit game, and perhaps more broadly. This work supports the idea that much coordination and cooperation, in both natural and man-made systems, can be viewed as an emergent property of the interaction of greedy agents maximizing their particular utility functions in the presence of environmental constraints.

In [KZ94], Kube and Zhang investigated mechanisms used to invoke group behavior, allowing a system of robots to perform tasks without centralized control or explicit communication. Some experiments with a system of five mobile robots, capable of achieving simple collective tasks, like pushing boxes, have shown that decentralized control without communication can be used in performing cooperative tasks requiring collective behavior. In [KB00], it is described a solution to the problem of cooperative transport of boxes by a group of robots, which is based on how ants cooperate in collective prey transport. The experimental setup consisted of a robot environment, in which various boxes were placed along with two spotlights used to indicate final goal positions, and a set of identical mobile robots. In total, over 100 box-pushing trials were run using from one to 11 robots, four different box types and three different venues. The robots did not make use of any form of explicit or direct communication. Given that the implementation of individual robot behavior was based on ant behavior, the dynamics of the swarm of robots was very reminiscent of the emergent cooperative dynamics of ants.

In [BHD94], Beckers et al. illustrated the concept of *stigmergy* (communication by means of modifying the environment) through the implementation of a robot experiment of collective pile formation. The robot team was a loosely coupled team, without explicit communication. Each robot was equipped with IR sensors, to detect obstacles, and a force sensor, triggered when more than two pucks were pushed; these sensors implemented three simpler behaviors: if some IR sensor had detected some obstacle (not a puck), the robot turned away from obstacle through a random angle; if force sensor had been activated, the robot pucks were dropped, reversing the motors for one second and turning away to a random angle; the default behavior was moving forward until some sensor was activated. After being positioned in the center of the work area and oriented randomly, robots started to move pucks; initially a few small piles were formed; gradually, the piles were aggregated, because when a robot detected it was pushing more than two pucks, it dropped them; by adding pucks to a pile, a robot made the pile larger and was voting for that pile to be the largest (*stigmergy*); at the end of the experiment, all pucks were in a single pile.

In [Mat92b, Mat93, Mat94, Mat95], having the *subsumption architecture* [Bro86] as framework to control a single robot, Matarić addressed the synthesis and analysis of group behavior and learning in complex environments, based on the belief that intelligent collective behavior in a decentralized system can result from local interactions based on simple rules. Based on the definition of a set of basic behaviors — safe-wandering, following, aggregation, dispersion and homing — it was developed a methodology that uses basic behaviors to generate various robust group behaviors, like flocking and foraging, through combination operators. The combination may be direct, by summation, or temporal, by switching. The generation of these more complex behaviors tries to maximize the synergy between agents, while minimizing inter-agent interference. It is also introduced a formulation of reinforcement learning, using behaviors as the unit of representation, that allows a group of agents to learn complex tasks by learning to select the basic behavior set. The new formulation of reinforcement learning consists of using conditions and behaviors for more robust control and minimized state-spaces, and a reinforcement shaping methodology that enables principled embedding of domain knowledge with two types of shaping functions: heterogeneous reward functions and progress estimators. The proposed swarm architecture was implemented and validated both in simulation and in groups of up to 20 real robots (Figure 4.5) ([5]).

In [BA98], Balch and Arkin presented a *behavior-based* approach to robot formation-keeping. The goal was to keep in formation teams of up to four unmanned ground vehicles intended to be fielded as a scout unit by the United States Army. Formations allow individual team members to concentrate their sensors across a portion of the environment, while their partners cover the rest. Besides military applications, the formation control is applicable in other domains, such as search and rescue, agricultural coverage tasks and security patrols.

The formation behaviors were implemented using *schema-based reactive control* [Ark89]. *Schema based systems* are a form of reactive control that make the fusion of different behavioral outputs through *vector summation*, in a manner analogous to the *potential fields method* [Kha86]. In these systems, each task component is coded as a separate behavior and outputs a vector indicating which direction the robot should travel. The vectors of all task components are further multiplied by a gain (a weight), summed and normal-

[5]Figure reproduced from [Mat94].

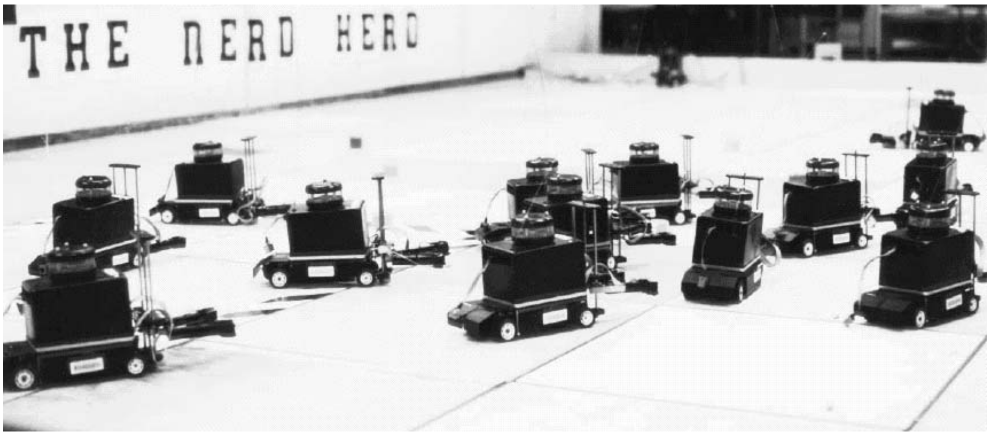

Figure 4.5: Some of the robots used to validate the group behavior methodology proposed by Matarić (1994).

ized to command the robot's movement. In [BA98], several *motor-schemas* — *move-to-goal*, *avoid-static-obstacle*, *avoid-robot* and *maintain-formation* — implement the overall behavior for a robot to move to a goal location while avoiding obstacles, collisions with other robots and remaining in formation. Endemic problems of *potential fields* techniques, such as local maxima, minima and cyclic behavior, were dealt with an additional *motor-schema* — *noise* — that generated movement in a pseudo-random direction. For instance, Figure 4.6 ([6]) depicts how the magnitude of the *maintain-formation* vector is computed. This vector is always in the direction of the desired formation position, but the magnitude depends on how far the robot is away from it. Several robot formations were considered, namely diamond, wedge, line and column. The approach was demonstrated on simulations (Figure 4.7) ([7]), on laboratory robots and on real scout units of the United States Army.

The work was continued in [BH00] to overcome several limitations of the seminal work, such as extending the technique to larger groups (more than four robots), and enabling that robots are not assigned to particular locations, but are instead attracted to the closest position in the formation. For these purposes, it was developed a novel perceptual technique to determine the proper formation position of a robot, using the concept of *attachment sites*. Each robot has several local *attachment sites* other robots may be attracted to. Using different *attachment sites configurations*, it is possible to

[6]Figure reproduced from [BA98].
[7]Figure reproduced from [BA98].

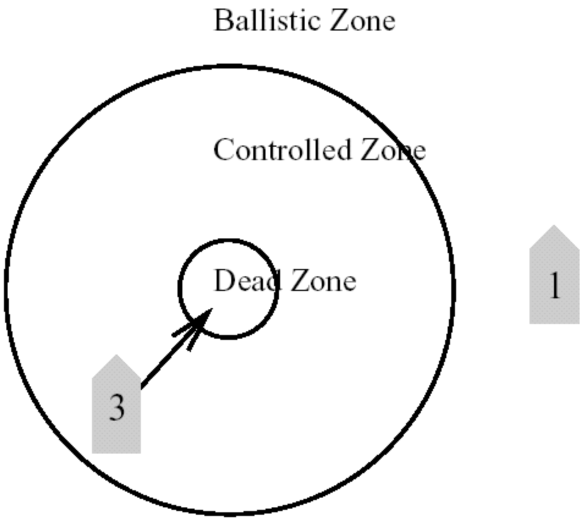

Figure 4.6: Zones for the computation of *maintain-formation motor-schema*. In the controlled zone, the magnitude varies linearly from a maximum at the farthest edge of the zone to zero at the inner edge. In the ballistic zone, the magnitude is set to a maximum. In the dead zone, the magnitude is zero. The dead zone provides tolerance to positional uncertainty, being its radius greater than or equal to the errors associated to this uncertainty.

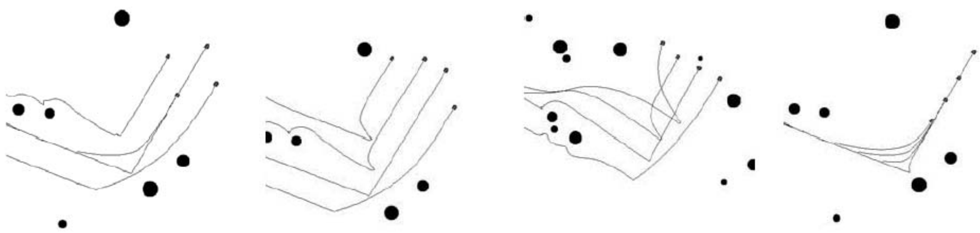

Figure 4.7: Four robots (small black dots) moving in formation, while avoiding obstacles (larger black dots). From left to right: diamond, wedge, line and column formations.

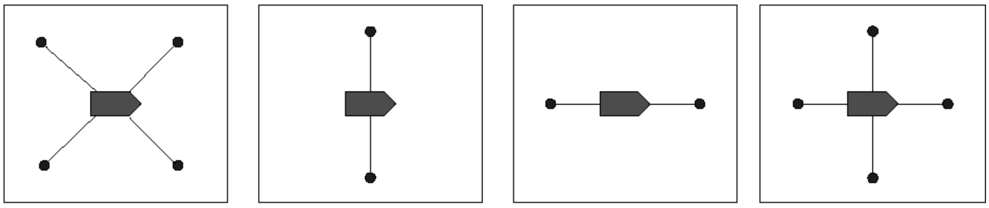

Figure 4.8: *Attachment site geometries* for different formations. From left to right: diamond, line, column and square.

configure different formations, with an arbitrary number of different robots (Figure 4.8) ([8]).

4.2 Some representative explicit cooperation case studies

There are mainly two bodies of research applicable to intentional or explicit cooperation: first, several researchers addressed the cooperative problem by using traditional Artificial Intelligence *planner-based approaches*, usually based on *sense-plan-act paradigm* and sometimes making reasonable assumptions about robot capabilities; second, *behavior-based approaches* that try to foster robustness and adaptability of the cooperative team, through situated, embodied, and sometimes learning physical robots.

4.2.1 Sense-plan-act based approaches

In [AMI89], Asama et al. presented the *ACTRESS* robot system, whose main objective was to develop the technology to synthesize multiple robotic elements. It was designed after analyzing the requirements of maintenance tasks in nuclear power plants. It was found that parallel action by multiple workers (e.g. robots) and their cooperation are essential in that context. *ACTRESS* (*ACTor-based Robot and Equipments Synthetic System*) is based on the classical *universal modular ACTOR formalism*, from Artificial Intelligence. This formalism provides a computational model in information processing, in which data structures and control structures are inseparably represented by a single kind of objects, called *actors*, and message passing between them. Robotic components were defined as *robotors* (robotic

[8]Figure reproduced from [BH00].

actors), as being autonomous components that have at least two basic functions: an ability to make decisions and an ability to communicate with any other components for parallel tasks, so that interference in components' motion is avoided or cooperative tasks are performed. These robotic actors are not necessarily robots and they can be information systems, such as intelligent sensors or knowledge base systems. *ACTRESS* is composed of a set of *robotors*, which usually have different structure and functions and embody a distributed system connected by a communication network. There are two possible communicating conditions: when a *robotor* acts independently and is only required to monitor the status of other *robotors* with occasional communication; and when a *robotor* executes a task cooperating with other *robotors* and is required to share the control signals with frequent communication. Communication in *ACTRESS* defines two independent protocols: a communication protocol to guarantee a reliable as possible data transmission; and a message protocol that defines a common syntax for communication. The latter one defines five different levels of communication, namely, control (e.g. control signals), physical (e.g. position, velocity, etc.), procedural (e.g. procedures to operate protocols), knowledge (e.g. knowledge about environment) and conceptual (e.g. intentions, objectives, etc.). *ACTRESS* also addresses task assignment and path planning among heterogeneous *robotors*, forming a negotiation framework that allows robots to recruit help when needed. In [AMI89], it was experimented through a moving obstacles task by micromouses. There were objects that could be moved by a single micromouse (light) and heavier objects that required cooperation of two micromouses to push it.

In [CCL$^+$90], Caloud et al. describe the *GOFER* project whose goal is to control the operation of many mobile robots in an indoor environment (e.g. office, shop-floor, airport, etc.), using traditional Artificial Intelligence techniques. It is described a *sense-plan-act architecture* which includes a *task planner*, a *task allocation*, a *motion planner* and an *execution monitor*. Each robot obtains goals to achieve either based on its own current situation, or via a request by another team member. The central planner communicates with all robots and has a global view of both the tasks to be performed and the availability of robots to perform the task. The central planner generates plan structures, comprising action hierarchies and sets of constraints, and informs available robots of the pending goals and plan structures. A plan structure can be viewed as a template for an instance of a plan. Further, task allocation uses a partially centralized method to allocate tasks to robots with respect to task characteristics and robot availability. Robots use a task allocation algorithm similar to *contract net protocol* [Smi80] to determine their roles and generate an instance of a plan, satisfying the constraints included in the plan structure. The motion planner is distributed and once a robot is allocated to a task, it is responsible for its own motion planning. Hierarchical Petri nets are used for interpretation of the plan decomposition and execution monitoring. The architecture was successfully used with three physical robots performing simple tasks, such as box-pushing and tracking walls in a corridor.

In [YP92], Yuta and Premvuti advised that the main challenge in designing an intelligent robot is to configure the proper balance between autonomy (in terms of decision making) and cooperation. Based on this assumption, it was proposed three levels for a multi-robot system, in which there is a common objective and each robot also has its own objectives: (1) a task specifying level, which specifies the common objectives of the system and assigns subtasks or roles to each robot by a centralized decision making process; (2) a robot objectives level, in which a robot works to achieve its objectives; (3) and a solving deadlock level, in which two robots are responsible for solving a deadlock problem (a common objective of the two robots) that is centrally solved by one of them. Thus, decision-making is continually switched between centralized and distributed modes. Most of the time, robots operate autonomously pursuing their own objectives and not disturbing each other. When the task must be specified or when a deadlock arises, the robots operate under centralized control, loosing some autonomy. A convention, denoted as *modest cooperation*, was proposed to deal with disturbances among robots, when there is a need to share the use of resources. For example, when a robot recognizes that a collision may occur if it tries to access the shared resource, it lets the other robot use it. In this case, if a deadlock arises (robots are waiting each other) it is centrally solved by one of the robots.

In [LH97], Lin et al. defined the *object-sorting task* (*OST*), which is a combination of foraging and cooperative transport. *OST* includes searching for objects in a wide area and transporting those objects to their destination (each found object has its own destination). As some objects (large objects) require more than one agent to be transported, explicit cooperation is needed to accomplish successfully the *OST*. This task abstracts parallelism and cooperation in just one task and it was used to study two cooperation protocols: *help-based cooperation protocol* (*HCP*) and *coordination-based cooperation protocol* (*CCP*). In both protocols, the working area is partitioned into disjoint sub-areas, and each sub-area is assigned to one agent, which exhaustively searches its sub-area. In *HCP*, when an agent finds a large object, it immediately requests help and then selects its partners from the agent willing to offer help. After there are enough agents arriving at the found object, the group of agents cooperatively transports the object to its destination. Since each agent autonomously selects its partners, simultaneous selection of partners may cause a deadlock. Several schemes were presented to handle deadlocks. In *CCP*, instead of requesting help once an object is found, an agent broadcasts the object's information to the group. As each agent has access to the information relative to all objects that have been found by the group, each agent makes its optimal decision and negotiates with the others for a global object transportation sequence. These negotiations take place in predefined coordination points, e.g., once some predefined number of objects have been found. Then, each agent moves objects according to its object sequence. Some social rules were employed in order to minimize the negotiation overhead. The performance of both protocols was compared through simulations and it was found that *CCP* is better than *HCP*. However, *CCP* uses more communication resources to maintain the global knowledge and to coordinate through negotiations.

In [LVAC99], Lima et al. introduced a *three-level architecture* for a team of fully autonomous mobile robots. Although the decomposition in three layers was inspired in a known multi-agent systems approach, innovative work was developed for inter-agent negotiation and role assignment. Complexity is reduced by the decomposition of team strategies (what should be done) into individual behaviors, which in turn are composed of primitive tasks. The set of behaviors assigned to each robot is designated as the tactics (how to do it) for a given strategy. The architecture is layered in *organizational*, *relational* and *individual* levels. The *organizational level* is modeled as a *state-machine* that establishes the strategy to be followed by the whole team, given the team and world states. The team state corresponds to the current set of behaviors under execution. The *relational level* establishes relationships among robots through negotiation. The robots are endowed with a individual and team goals and negotiate the adequate tactics (a recipe) to pursue the strategy defined by the organizational layer, using concepts inspired on the *joint intentions framework* [CL91]. *Individual behavior level* encompasses all the available robot behaviors and their relations. A *behavior* is decomposed in a set of purposive (with a goal) *primitive tasks* sequentially and/or concurrently executed. Each behavior is modeled as a *state-machine*, being each primitive task a *sense-think-act loop*, which is a generalization of a closed loop control system. The logical conditions that determine the execution of the sequence of primitive tasks are defined over a predicate set. The world model that provides information to the relational and organizational levels is implemented as a *distributed blackboard*, which can be viewed as a global shared memory and event-based communication. This architecture has been validated in a robot soccer team, which has participated in the RoboCup's middle-size league competitions.

Alami et al. from the LAAS/CNRS (France) have been involved in the

MARTHA project, which addresses the control and the management of fleets of autonomous mobile robots for transshipment tasks in harbors, airports and marshaling yards [AFH+98]. The project's most challenging problem is multi-robot cooperation with as little as possible centralized control, evolving in already existing open sites, not designed specifically to that application and that may be traversed by other vehicles. The proposed decentralized approach only requires local communication between the robots and a low bandwidth intermittent communication with the *central station*. It gives more autonomy to the vehicles to allow them to cope with unexpected events and obstacles and inaccurate environment models. This autonomy is accomplished providing robots with advanced sensory and perceptual capabilities (localization, obstacle detection and modeling), as well as planning and deliberation through local communication and coordination. Robots incrementally determine the resources they need, taking into account the execution context.

The *LAAS's architecture* [AFH+98] is composed of a *central station* (*CS*) and a *set of autonomous mobile robots* able to communicate with each other and with the *CS*. Both the *CS* and the robots make use of the same description of the environment when pursuing specification, robot navigation, or multiple conflict resolution. The environment model is a topological and geometrical representation of the environment, consisting of areas, routes and crossings. Although it is minimized the usage of a *CS*, it is still necessary to plan the transshipment operations and the routes the robot should use. However, the required communication bandwidth between the *CS* and the robots is low and the required processing power in the *CS* is not very significant than would be in a completely centralized system, because the *CS* neither intervenes in the robot plans coordination (e.g. in crossing areas), nor plans the precise trajectories that are executed by the robots. Each robot receives its mission from the *CS* and then performs it on its own. In doing that, it refines the mission and plans its routes and trajectories. The resulting plans are further coordinated with the plans of other robots. While the robot executes its mission, it also monitors critical situations (e.g. unknown obstacles) and reports unrecoverable action failures to the *CS* that need external assistance (e.g. by a human operator). Alami et al. developed the *plan-merging-paradigm* [AFH+97], which is used by the autonomous robots to *refine*, *plan* and *coordinate* route sections and crossings use, as well trajectories in open areas. It is a domain independent multi-robot cooperation scheme, which is applicable to systems that involve simultaneous operation of several autonomous agents, each one seeking to achieve its own task or goal. The basic idea is that whenever a robot produces a plan that makes use of some kind of resources (e.g. trajectories), it must validate it in the current multi-robot context. With this purpose, it advertises it and collects from other robots the plans that specify how they plan to use those resources, as well as the right to perform its plan coordination (tokens used for mutual exclusion). Further, the robot merges its own plan with the other robots' plans, produces eventually a coordinated plan (if it is possible) and informs the other robots of events occurrence of which it wants to be informed, so that it can synchronize itself with those events. Although it is not claimed that this merging approach may not always solve multi-robot planning problems, it is claimed that it is safe and guarantees a coherent collective behavior. The approach was validated with a large number of emulated robots under a Unix simulator and with three mobile robots in laboratory.

In [Bot00, BA00], Botelho and Alami extended the previous work with the LAAS's coordination architecture, in order to accomplish more complex and generic missions requiring autonomous and deliberative agents with the ability of planning their actions, perform their tasks in a coherent and non-conflict manner and cooperatively enhance their performance. Any *autonomous multi-robot system* should address the *decomposition of a mission* into tasks (*mission planning*), the *allocation* of the obtained tasks among the available robots and the *task achievement* in a multi-robot context. The work only addresses the latter issue, assuming that a set of autonomous robots have been given a set of partially ordered tasks (e.g. the output a central planner). However, the allocated tasks cannot be executed and require further refinement to cope with a multiplicity of uncertainties. Thus, after the robot has synthesized its own plan for achieving the allocated tasks, it is necessary to avoid and/or solve conflicts and to enhance the efficiency of the system. In order to address the latter issue, Botelho and Alami developed the *M+ cooperative task achievement* [BA99], which have the following features: *opportunistic action re-allocation* when some robot opportunistically detects that it will be beneficial for the global performance if it could perform an action that was originally planned by another robot; *suppression of redundancy*, when various robots have planned the execution of redundant actions; and *incremental/additive actions*, which allows the robots to detect an action originally planned by one robot can be incrementally achieved by several robots, being this beneficial to the global performance. The *M+* main ingredients are a *world description*, a set of *social rules* and their use in a *cooperative decisional process* based on *incremental planning*, as well as on a set of *mechanisms for plan adaptation*. The *world description* is described through two sets of *predicates*: *stable* and *evolutionary predicates*. While *stable predicates* represent constant environment features, *evolutionary predicates* represent features that can be changed and whose modification can be planned. In the latter set, there is the *exclusive predicates* subset, which represents features that can only be changed by the robot itself. After a robot has been allocated to a task, it produces its individual plans, which are further incrementally adapted to the multi-robot context, through a negotiation process with the other robots. The *planning* and *negotiation* activities are constrained by a set of *social rules*, which facilitate the production of merge-able plans. Each social rule is associated to an *obligation level*, which helps to distinguish between rules that must be systematically respected and rules that can be deferred (planned but not necessarily executed). The proposed cooperative multi-robot task achievement involves: *task planning*, a purely internal activity that produces merge-able plans; *plan negotiation*, which adapts the plan to the multi-robot context; and *effective plan execution*. The two latter activities are performed in a critical section, in order to ensure a coherent distributed mobile robot plan management and execution. The negotiation activity is based on auctions. This cooperative decision process was demonstrated through a simulating model of a hospital, where three robots execute servicing tasks.

4.2.2 Behavior-based approaches

In [WM00], Werger and Matarić presented the *broadcast of local eligibility* (*BLE*), a general tool for coordination between robots, which extended the *port-arbitrated behavior* paradigm across a network of robots. The goal was to demonstrate that *behavior-based* systems, restricted to well-defined *port-arbitrated interactions*, could scale to higher levels of competence than was generally assumed. *BLE* is comprised of three specific *ports*: *local*, *best* and *inhibit*. Each robot makes a local estimate of its own eligibility for a given task, which is derived from the robot's own sensors. This eligibility is written to the appropriate behavior's *local port*, which is connected so as to broadcast the estimate to the *best port* of each behavior of the same name, on every robot on the local network. The *best port* filters all the incoming messages for the maximum. A comparison is made between the locally determined eligibility (*local port*) and the best eligibility (*best port*). When a robot's local eligibility is best for some behavior, it writes to its *inhibit port*, which is connected so as to inhibit the peer behaviors on all other robots. As this is an active inhibition, if a robot fails to execute a task, the task is immediately freed for potential takeover by another robot. *BLE* was validated in a multi-target tracking task. It was shown that *BLE-based systems* are able to dynamically reconfigure themselves in order to allocate resources in response to task constraints, environmental conditions and system resources.

In [GM00], Gerker and Matarić described a framework for inter-robot communication (interaction via communications) that is used to dynamically allocate tasks in teams of cooperative mobile robots. Towards this end, it was proposed *Murdoch*, a principled, resource-centric, completely distributed, publish/subscribe communication model, which makes extensive use of explicit communication. It offers a distributed approximation to a global optimum of resource usage, which is equivalent to a greedy scheduler. The communication model is a *broadcast-oriented blackboard* model, in which messages are addressed by content (subject) rather than by destination. In order to allocate a given task, it is used a simple auction, somewhat similar to the *contract nets protocol* [Smi80], in which each capable agent evaluates its own fitness for the task. The auction's winner is committed to perform the task until success of failure. *Murdoch* was validated in two different task domains: a short-term tightly-coupled cooperative box-pushing task by a team of three robots; and a long-term loosely-coupled multiple target tracking task, with many robots executing a collection of independent single-robot tasks.

In [MS01], Matarić and Sukhatme addressed the problem of *dynamic task allocation* in a group of multiple robots satisfying multiple goals, focusing on three studies: a first *opportunistic approach* using *broadcast of local eligibility* (*BLE*) and mutual inhibition among robots [WM00]; a second *commitment approach* using *Murdoch*, a task allocation mechanism based on market-based auction [GM00]; and a third approach for studying the *trade-off between opportunistic-based and commitment-based* task allocation. The three approaches to multi-robot coordination can be viewed using a common framework: all use communication among the robots through a blackboard. Each robot sends its relevant state communication to the blackboard at a fixed frequency and all robots read the blackboard information at a lower frequency. In the third approach, an emergency-handling problem, inspired on planetary exploration, was used to compare the impact of *opportunistic vs. commitment*, which is the main difference between *BLE* and *Murdoch*. It was used a set of three homogeneous robots and four different alarms. Deciding which robot should go where and when, was viewed as a dynamic (robots did not know the alarms a priori), distributed (multi-robot solution) and robust (to failure of individuals) scheduling algorithm. The property of a robot might hear an alarm before it saw the emergency was realistically simulated through the development of sound-emitting alarms detectable by the robots' microphones, before the robots were within visual range of the sound source. In the context of this experience, *commitment* means that, once assigned, a robot stays dedicated to handling a particular alarm, until the alarm can no longer be detected. Instead, *opportunism* means that a robot can switch alarms, if for example it detects another alarm with greater intensity or priority. The study showed that the opportunistic strategy worked significantly better that the *commitment-based strategy*, because the time taken by a robot to reach the source of an alarm was significantly larger than the time it took a robot to fix an alarm, once a robot was there.

In [Jun98], Jung proposed the *ABBA* architecture (*Architecture for Behavior Based Agents*), which supports the distributed planning of cooperative behavior in *behavior-based* multi-robot systems. The basic idea was to extend the action planning between behavior elements within a single agent (an action selection problem) to the cooperative action planning between behavioral elements distributed by different agents, which are able to communicate. The proposed architecture took inspiration in interaction schemes observed in biological systems, which are usually layered. Layering aids to manage complexity and increases the robustness of the system, because the failure of a high-level behavior will not cause the failure of the system to complete its task, but only a temporary reduction in performance. Each *layer* builds the sophistication of cooperation and communication by relying on the layers below it and sophistication of the communication scales with that of cooperation. *Four layers* were implemented for the solution of a cleaning task by two cooperative robots. The first (lower) layer implemented *emergent cooperation*, where the behavior of each robot was designed so that collective interaction solves the problem. In this lower layer, there was no awareness in the robots, no communication and no map learning. The second layer implemented *cooperation by passive observation*, adding the capacity for one robot to visually identify and track the other. This awareness increasing, in combination with limited reasoning about the actions and intentions of the other robot, provided performance-enhancing information when in visual range. The third layer implemented a *cooperation scheme by explicit communication*, which enhances the knowledge of one robot about the actions of the other. The fourth (upper) layer implemented *cooperation by planning*, which considered speech acts as interactions that affect the other robots' actions. This upper layer employs communication involving symbolic language for which is given a shared grounding (e.g. location labeling), although the robots have different sensory systems. The shared grounding in the behavior-sensor space of each robot is used to jointly plan actions, learn a spatial topological map of the environment and purposively navigate by it. The architecture was validated in a cleaning task by two cooperative robots with different sensory capabilities, which vacuumed up scattered pieces spread over the floor, including close to the walls and around furniture. One of the robots was vision-based and could not clean close to the walls, while the other one was based on whiskers that enabled it to follow walls, clean near to the walls and detect frontal obstacles. Results showed that symbolic communication of location information significantly enhanced the team performance on the cleaning task.

One of the most referenced architectures for cooperative systems is *ALLIANCE*, which was proposed by Parker [Par94, Par98]. *ALLIANCE* is a *behavior-based* approach to the control of small-to-medium teams of multiple, loosely coupled and heterogeneous robots. It is supported in Brook's *subsumption architecture* [Bro86], with three levels of control (Figure 4.9) ([9]). Robots are assumed to be able to sense, with some probability, the effects of their own actions and the actions of other robots, through perception and explicit broadcast of information. For this purpose, the highest level, designated as *motivational behaviors*, models the *impatience* and the *acquiescence* of the robot. *Impatience* measures the attitude of a robot towards the other robots in the team, increasing when the performance of the other robots is such that the task assigned to the system is not being accomplished. *Acquiescence* measures the attitude of the robot towards itself, increasing when the robot is taking too much time to accomplish its own task and recognizes that it may fail. The intermediate level of the architecture contains groups of *behavior sets*, with mutually exclusive behaviors placed in different sets. At each time, the *motivational behaviors* level only activates a single behavior set. The lowest level is composed by *basic behaviors*, or competences, that must be always active (e.g. avoiding obstacles). The access to the actuators by the behaviors is made using the suppression or inhibition mechanisms of the *subsumption architecture*. When higher-level behaviors need to take control of the actuators, they subsume the roles of lower level behaviors. The most critical design issue of an *ALLIANCE-based system* is setting the parameters controlling behavior set activation (e.g. how fast a robot becomes impatient). For this reason, it was developed an extension, called *L-ALLIANCE*, providing mechanisms that allow robots to dynamically *learn* to update their parameter settings based upon previous experiences (reinforcement learning). For this purpose, each robot uses performance monitors to observe, evaluate and catalogue the performance of any robot team member. Both architectures have been successfully validated for some cooperative tasks, including box-pushing, puck-gathering, moving in formation, janitorial service and hazardous waste cleanup [Par94], and multi-target observation [Par02].

[9]Figure reproduced from [Par94].

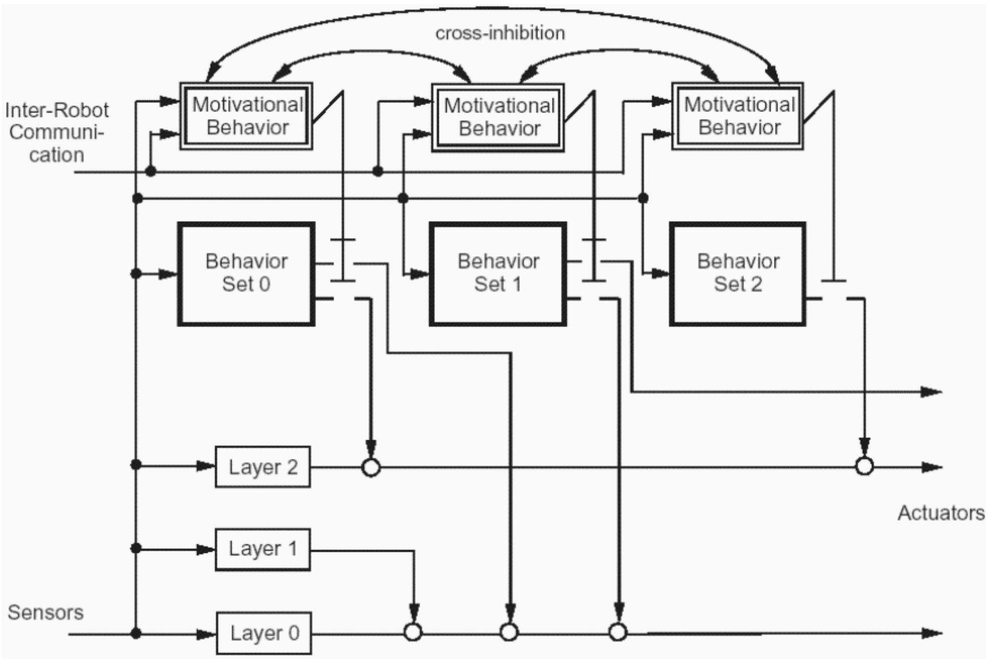

Figure 4.9: The *ALLIANCE* architecture.

Chapter 5

Conclusion and discussion

Fostering cooperation among intelligent machines is undoubtedly an important challenge in many different applications, where human teams are required to cooperatively execute complex missions. Those intelligent machines (e.g. autonomous mobile robots) can potentially reduce the need for human presence in hazardous environments, or may substitute humans in highly repetitive and monotonous tasks, perhaps with potential efficiency benefits. There are several reasons why cooperation is required to carry out the missions involved in such applications: an individual may be not able to accomplish the mission on its own; or the mission is intrinsically distributed in space, thus requiring distribution of resources; or the mission requires more efficient execution through parallelism and concurrent execution of many subtasks; or the mission is so complex that, in order to manage complexity, it must be decomposed on different subtasks that are concurrently executed. Even in some applications where a single agent solution could be considered, a multi-agent solution is generally preferable, in order to obtain more costly effective, reliable, robust, extensible and modular solutions.

In this report, a significant part of research work on cooperative multi-agent systems, particularly multi-robot systems, has been referred. After an introduction to cooperative systems, where it was mentioned what are their motivations, the report started by presenting some knowledge about the cooperation concept itself, borrowed from studies about manifestations of cooperation in nature, both in human societies and animal species. Then, a few taxonomies of cooperative systems already known in the literature were presented as a means to organize related work along different design axes. Those taxonomies were used to give some structure to the presentation of previous research on cooperative systems.

The main research topics of multi-agent cooperative systems are: teamwork, multi-agent coordination, cooperative perception, cooperative planning and cooperative learning. Extensive work covering these topics has been carried out for the last decade. A challenging and common design requirement to all these research directions is to develop new techniques that might work in complex, dynamic and uncertain environments. In this sense, there are a lot of open issues within those topics to cover in the future, in order to attain such challenging requirements. Cooperative systems that bear inspiration on biological systems are already well studied and explored and they are mainly suited to tasks that require the weakest forms of cooperation.

However, there are also some orthogonal issues that, although they have crucial importance in designing cooperative systems, they have not been covered yet, or at least they have been only superficially covered [ESS02]. Examples of these issues are theoretical approaches and performance metrics.

The main criticism about the current state of the art of research on cooperative systems is its informal and concept orientation. There is still a big lack of rigorous formalisms to clarify various assumptions about the systems being discussed, which might give a more precise language for discussion of elusive concepts such as cooperation. Most of the proposed architectures constitute a solution to a given instantiation of the general problem, which are perhaps demonstrated to work well within some predefined assumptions. However, little effort has been done to develop systematic and formal design guidelines, which might be used to assess what is the most suited cooperation protocol for a given global task and context.

The current theory of cooperative systems is mainly an instantiation of the concepts of agent and multi-agent systems (MAS), though there are some independent issues that are not found within MAS framework, especially those that concern with technological limitations (e.g. limited communication, limited and inaccurate perception, etc.). It might seem that MAS are a panacea to cope with all real problems and that cooperative systems are simply an example of their possible application domains. But this is a completely wrong idea, because there are no panaceas in research. MAS are commonly used to formalize cooperative systems simply because they are suited to their requirements, namely distribution of intelligence and resources, adaptability, dynamic organization and structure, etc. This is why MAS have had such an impact in research on cooperative systems. Although this influence is evident, it is not straightforward to apply in cooperative systems, especially multi-robot systems, the extensive amount of techniques and results that have been developed within the MAS framework by Artificial Intelligence researchers. This is a horizontal issue for which an effective answer has not been given yet. Theoretical approaches, such as MAS, adaptive and robust control of discrete event dynamic systems, Markov decision processes, etc., are welcome only if they are grounded on the real systems requirements and constraints. Theoretical and practical approaches must be carefully balanced.

Cooperative systems have a large potential to accomplish some tasks with better performance than single agent systems or monolithic solutions, but little effort has been devoted to ensure and quantitatively measure that performance. Formal metrics for cooperation and system performance (e.g. grades of cooperation) are noticeably missing form the literature. For instance, nobody has already answered the following question: Given a multi-robot mission and the environmental constraints, what is the right number of robots that are able to accomplish the mission within a given period of time ([1])?

Although previous work on communication structures for multi-robot systems has led to some useful conclusions and design guidelines, there is no a principled formalism that can be systematically used to assess information utility and support the efficient use of communication, whether implicit or explicit, in cooperative systems. Current architectures extensively use explicit communication (e.g. broadcast type communication), not taking care, giving low emphasis, or using no principled heuristics to avoid the communication of redundant information. As communication is always limited, either in resources applied to perceive the world or in bandwidth of a communication channel, using efficiently those resources is crucial to scale up cooperative architectures for teams of many robots, without limiting them to simple reactive and loosely-cooperative systems, with very limited or no awareness. Even with unlimited resources, some questions are not yet answered, such as what, when and how to communicate.

The previous comments mainly represent the scientific push that will foster future research on cooperative systems. Although science is largely domain-independent, future research will be also influenced by the application pull [ESS02]. Some examples of this application pull are applications on military domains, service robots, assistant robots, entertainment robots, tele-autonomy, etc. On military domains, the most obvious applications are reconnaissance, surveillance, mine clearing operation, decontamination, security, decoy and deception, transport, search and rescue, etc. In the future, usability is an essential factor for robotics if it is expected that robotics go beyond laboratories, industrial and military applications. Within this context, multi-robot systems should be able to cope with complex missions and tasks and to friendly interact with human (human-robot interaction) and other teams of robots, in order to assist and empower humans in several tasks. Tele-autonomy will enable humans to remotely and easily control a team of

[1] In this example, time was the chosen metric for evaluating the performance of the multi-robot system.

robots as a whole. Like assistant robots, entertainment robotics will develop as a consequence of the robotics to the masses trend (e.g. robot soccer).

Bibliography

[AB98a] R. Arkin and T. Balch. Cooperative multiagent robotic systems. In D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots*. MIT Press, 1998.

[AB98b] R. Aylett and D. Barnes. A multi-robot architecture for planetary rovers. In *Proc. of 5^{th} ESA Workshop on Space Robotics, ASTRA'98*, 1998.

[ABN93] R. Arkin, T. Balch, and E. Nitz. Communication of behavioral state in multi-agent retrieval tasks. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'93)*, volume 3, pages 588–594, 1993.

[AFH+97] R. Alami, S. Fleury, M. Herrb, F. Ingrand, and S. Qutub. Operating a large fleet of mobile robots using plan-merging paradigm. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'97)*, pages 2312–2317, 1997.

[AFH+98] R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert. Multirobot cooperation in the martha project. *IEEE Robotics and Automation Magazine*, 5(1):36–47, 1998.

[AMI89] H. Asama, A. Matsumoto, and Y. Ishida. Design of an autonomous and distributed robot system: ACTRESS. In *Proc. of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS'89), Tsukuba, Japan*, pages 283–290, 1989.

[Ark89] R. Arkin. Motor schema based mobile robot navigation. *International Journal of Robotics Research*, 8(4):92–112, 1989.

[Ark92] R. Arkin. Cooperation without communication: Multi-agent schema based robot navigation. *Journal of Robotic Systems*, 9(3):351–364, 1992.

[AS97] K. Azarm and G. Schmidt. Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'97)*, volume 4, pages 3526–3533, 1997.

[Axe80a] R. Axelrod. Effective choice in the prisoner's dilemma. *Journal of Conflict Resolution*, 24:3–25, 1980.

[Axe80b] R. Axelrod. More effective choice in the prisoner's dilemma. *Journal of Conflict Resolution*, pages 379–403, 1980.

[Axe84] Robert Axelrod. *The Evolution of Cooperation*. New York, Basic Books, 1984.

[BA94] T. Balch and R. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52, 1994.

[BA98] T. Balch and R. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

[BA99] S. Botelho and R. Alami. M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'99)*, pages 1234–1239, 1999.

[BA00] S. Botelho and R. Alami. Robots that cooperatively enhance their plans. In *Proc. of 5^{th} International Symposium on Distributed Autonomous Robotic Systems (DARS'00), Knoxville, Tennessee, USA*, 2000.

[Bal98] Tucker Balch. *Behavioral Diversity in Learning Robot Teams.* PhD thesis, Mobile Robot Laboratory, Georgia Institute of Technology, USA, 1998.

[Bal00] T. Balch. Hierarchical social entropy: An information measure of robot group diversity. *Autonomous Robots, Special Issue on Heterogeneous Multi-Robot Systems*, 8(3):209–237, Jun. 2000.

[Bal02] T. Balch. Taxonomies of multirobot task and reward. In T. Balch and L. Parker, editors, *Robot Teams: From Diversity to Polymorphism.* A. K. Peters Ltd., 2002.

[BB97] C. Boutilier and R. Brafman. Planning with concurrent interacting actions. In *Proc. of National Conference on Artificial Intelligence (AAAI'97)*, 1997.

[BDG99] E. Bonabeau, M. Dorigo, and Thraulaz G. *Swarm intelligence: from natural to artificial systems*. Oxford University Press, 1999.

[BdL+99] M. Beer, M. d'Inverno, M. Luck, N. Jennings, C. Preist, and M. Schroeder. Negotiation in multi-agent systems. *Knowledge Engineering Review*, 14(3):285–289, 1999.

[Ben88] G. Beni. The concept of cellular robotic system. In *Proc. of IEEE International Symposium on Intelligent Control*, pages 57–62, 1988.

[BG00] E. Bonabeau and Thraulaz G. Swarm smarts. *Scientific American*, pages 72–79, Mar. 2000.

[BH00] T. Balch and M. Hybinette. Social potentials for scalable multi-robot formations. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'00)*, pages 73–80, 2000.

[BHD94] R. Beckers, O. Holland, and J. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In *Proc. of Artificial Life IV*. MIT Press, 1994.

[Bot00] Silvia Botelho. *Une Architecture Décisionnelle pour la Coopération Multi-Robots.* PhD thesis, LAAS/CNRS, Toulouse, France, 2000.

[Bro86] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, RA-2:14–23, 1986.

[CCL+90] P. Caloud, W. Choi, J.C. Latombe, C. Le Pape, and M. Yim. Indoor automation with many mobile robots. In *Proc. of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS'90), Tsuchiura, Japan*, pages 67–72, 1990.

[CFA+95] A. Cai, T. Fukuda, F. Arai, T. Ueyama, and A. Sakai. Hierarchical control of cellular robotic system: Simulations and experiments. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'95)*, pages 1191–1196, 1995.

[CFK97] Y. Cao, A. Fukunaga, and A. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:1–23, 1997.

[CL91] P. Cohen and H. Levesque. Teamwork. *Nous, Special Issue on Cognitive Science and Artificial Intelligence*, 25(4):487–512, 1991.

[DFJN97] J. Doran, S. Franklin, N. Jennings, and T. J. Norman. On co-operation in multi-agent systems. *The Knowledge Engineering Review*, 12(3):309–314, 1997.

[DJM02] G. Dudek, M. Jenkin, and E. Milios. A taxonomy of multirobot systems. In T. Balch and L. Parker, editors, *Robot Teams: From Diversity to Polymorphism*. A. K. Peters Ltd., 2002.

[DJMW93] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for swarm robotics. In *Proc. of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS'93)*, pages 441–447, 1993.

[DJMW96] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3:375–397, 1996.

[DMC96] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernatics - Part B: Cybernatics*, 26(1):1–13, 1996.

[ER94] E. Ephrati and J. Rosenschein. Divide and conquer in multi-agent planning. In *Proc. of National Conference on Artificial Intelligence (AAAI'94)*, pages 375–380, 1994.

[ER95] E. Ephrati and J. Rosenschein. A tractable heuristic that maximizes global utility through local plan combination. In *Proc. of 1st International Conference on Multi-Agent Systems*, 1995.

[ESS02] *European Summer School on Cooperative Robotics*. Allessandro Saffiotti and Isabel Ribeiro, chairs. Organized by the Special Interest Group on Cooperative Robotics of EURON - European Robotics Research Network, Instituto Superior Técnico, Lisbon, Portugal, Sep. 2-7 2002.

[FS94] T. Fukuda and K. Sekiyama. Communication reduction with risk estimate for multiple robotic system. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'94)*, pages 2864–2869, 1994.

[GM00] B. Gerkey and M. Matarić. Principled communication for dynamic multi-robot task allocation. In *Proc. of International Symposium on Experimental Robotics 2000, Waikiki, Hawaii, USA*, pages 341–352, 2000.

[HB91] S. Hackwood and G. Beni. Self-organization sensors by deterministic annealing. In *Proc. of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS'91)*, pages 1177–1183, 1991.

[HB92] S. Hackwood and G. Beni. Self-organization of sensors for swarm intelligence. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'92)*, pages 819–829, 1992.

[INS01] L. Iocchi, D. Nardi, and M. Salerno. Reactivity and deliberation: A survey on multi-robot systems. In M. Hannebauer, J. Wendler, and E. Pagello, editors, *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*. Springer-Verlag, 2001.

[Jen93] N. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1993.

[Jen94] N. Jennings. *Cooperation in Industrial Multi-Agent Systems*. World Scientific Publishing, Singapore, 1994.

[Jen95] N. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 75:195–240, 1995.

[Jen96] N. Jennings. Coordination techniques for distributed artificial intelligence. In G. O'Hare and N. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. John Wiley & Sons, New York, 1996.

[JLB94] K. Jin, P. Liang, and G. Beni. Stability of synchronized control of discrete swarm structures. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'94)*, pages 1033–1038, 1994.

[Jun98] David Jung. *An Architecture for Cooperation among Autonomous Agents.* PhD thesis, Dep. of Computer Science, University of Wollongong, Australia, 1998.

[JZ00] D. Jung and A. Zelinsky. Grounded symbolic communication between heterogeneous cooperating robots. *Autonomous Robots, Special Issue on Heterogeneous Multi-Robot Systems*, 8(3):269–292, Jun. 2000.

[KANM98] H. Kitano, M. Asada, I. Noda, and H. Matsubara. RoboCup: Robot world cup. *IEEE Robotics and Automation Magazine*, 5(3):30–36, 1998.

[KB00] C.R. Kube and E. Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30(1-2):85–101, 2000.

[Kha86] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[Kor92] R. Korf. A simple solution to pursuit games. In *Proc. of 11^{th} International Workshop on Distributed Artificial Intelligence*, pages 183–194, 1992.

[KZ94] C.R. Kube and H. Zhang. Collective robotics: From social insects to robots. *Adaptive Behavior*, 2(2):189–219, 1994.

[LH97] F.C. Lin and J.Y. Hsu. Cooperation protocols in multi-agent robotic systems. *Autonomous Robots*, 4:175–198, 1997.

[LVAC99] P. Lima, R. Ventura, P. Aparício, and L. Custódio. A functional architecture for a team of fully autonomous cooperative robots. In M. Veloso and H. Kitano, editors, *RoboCup'99: Robot Soccer World Cup III.* Lecture Notes in Computer Science, Springer-Verlag, 1999.

[Mat92a] M. Matarić. Behavior-based systems: Main properties and implications. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'92), Workshop on Architectures for Intelligent Control Systems, Nice, France*, pages 46–54, 1992.

[Mat92b] M. Matarić. Minimizing complexity in controlling a mobile robot population. In *Proc. of IEEE International Conference on*

Robotics and Automation (ICRA'92), Nice, France, pages 830–835, 1992.

[Mat93] M. Matarić. Kin recognition, similarity and group behavior. In *Proc. of 15th Annual Cognitive Science Conference, Boulder, Colorado*, pages 705–710. Lawrence Erlbaum Associates, 1993.

[Mat94] Maja Matarić. *Interaction and Intelligent Behavior*. PhD thesis, Dep. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA, 1994.

[Mat95] M. Matarić. Designing and understanding adaptive group behavior. *Adaptive Behavior*, 4(1):51–80, 1995.

[Mat99] M. Matarić. Behavior-based robotics. In R. Wilson and F. Keil, editors, *MIT Encyclopedia of Cognitive Sciences*, pages 74–77. MIT Press, 1999.

[MNS95] M. Matarić, M. Nilsson, and K. Simsarian. Cooperative multi-robot box-pushing. In *Proc. of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS'95)*, volume 3, pages 556–561, 1995.

[MS01] M. Matarić and G. Sukhatme. Task-allocation and coordination of mltiple robots to planetary exploration. In *Proc. of 10th International Conference on Advanced Robotics (ICAR'01), Buda, Hungary*, 2001.

[OJ96] G. O'Hare and N. Jennings, editors. *Foundations of Distributed Artificial Intelligence*. Sixth Generation Computer Technology Series. John Wiley & Sons, New York, 1996.

[Par93] L. Parker. Designing control laws for cooperative agent teams. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'93)*, volume 3, pages 582–587, 1993.

[Par94] Lynne Parker. *Heterogeneous Multi-Robot Cooperation*. PhD thesis, Dep. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA, 1994.

[Par95] L. Parker. The effect of action recognition and robot awareness in cooperative robotic teams. In *Proc. of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS'95)*, pages 212–219, 1995.

[Par98] L. Parker. ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[Par00] L. Parker. Current state of the art in distributed autonomous mobile robotics. In G. Bekey and J. Barhen, editors, *Distributed Autonomous Robotic Systems 4*. Springer-Verlag, 2000.

[Par02] L. Parker. Distributed algorithms for multi-robot observation of multiple moving targets. To appear in Autonomous Robots, 2002.

[PM00] P. Pirjanian and M. Matarić. Multiple objective vs. fuzzy behavior coordination. In D. Drainkov and A. Saffiotti, editors, *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*, pages 235–253. Studies on Fuzziness and Soft Computing, Springer-Verlag, 2000.

[RDJ95] D. Rus, B. Donald, and J. Jennings. Moving furniture with teams of autonomous robots. In *Proc. of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS'95)*, volume 1, pages 235–242, 1995.

[Saf97] A. Saffiotti. Fuzzy logic in autonomous robotics: Behavior coordination. In *Proc. of 6th IEEE International Conference on Fuzzy Systems, Barcelona, Spain*, pages 573–578, 1997.

[Smi80] R. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, c-29(12):1104–1113, 1980.

[SR01] A. Saffiotti and E. Ruspini. Global team coordination by local computation. In *Proc. of European Control Conference (ECC'01), Porto, Portugal*, 2001.

[SV99] P. Stone and M. Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 110(2):241–273, 1999.

[SV00] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots, Special Issue on Heterogeneous Multi-Robot Systems*, 8(3):345–383, Jun. 2000.

[Tam97] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

[Tou00] C. Touzet. Robot awareness in cooperative mobile robot learning. *Autonomous Robots*, 2:1–13, 2000.

[UF93a] T. Ueyama and T. Fukuda. Knowledge acquisition and distributed decision making. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'93)*, volume 3, pages 167–172, 1993.

[UF93b] T. Ueyama and T. Fukuda. Self-organization of cellular robots using random walk with simple rules. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'93)*, volume 3, pages 595–600, 1993.

[WM00] B. Werger and M. Matarić. Broadcast of local eligibility for multiple-target observation. In *Proc. of 5^{th} International Symposium on Distributed Autonomous Robotic Systems (DARS'00), Knoxville, Tennessee, USA*, pages 347–356, 2000.

[YP92] S. Yuta and S. Premvuti. Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots. In *Proc. of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS'92)*, pages 1566–1574, 1992.

[YZD02] M. Yim, Y. Zhang, and D. Duff. Modular robots. *IEEE Spectrum*, pages 30–34, Feb. 2002.