Communication as the Basis for Learning in Multi-Agent Systems

M. Kaiser, R. Dillmann, O. Rogalla
University of Karlsruhe
Institute for Real-Time Computer Systems & Robotics
D-76128 Karlsruhe, Germany
Phone: +49 721 6084051 – Fax: +49 721 606740 – E-Mail: kaiser@ira.uka.de

Abstract

This paper discusses the significance of communication between individual agents that are embedded into learning Multi-Agent Systems. For several ***learning tasks*** *occurring within a Multi-Agent System, communication activities are investigated and the need for a mutual understanding of agents participating in the learning process is made explicit. Thus, the need for a* ***common ontology*** *to exchange learning-related information is shown. Building this ontology is an additional learning task that is not only extremely important, but also extremely difficult. We propose a solution that is motivated by the human ability to understand each other even in the absence of a common language by using alternative communication channels, such as gestures.*

1 Introduction

Learning in Multi-Agent Systems has become a major research field within Distributed Artificial Intelligence and Machine Learning [20, 17]. It is motivated by the insight that it is impossible to determine **a-priori** the complete knowledge that must exist within each component of a distributed, heterogeneous system in order to allow satisfactory performance of that system. Especially if we want to exploit the potential of modularity, such that it is possible for individual agents to join and leave the Multi-Agent System, there's a constant need for the acquisition of new and the adaption of already existing knowledge, i.e., for learning.

Within this setting, different kinds of learning tasks must be investigated, such as "traditional" single agent learning tasks, learning in teams, learning to act within a team, and learning to coordinate other agents. To solve any of these tasks, the existence of appropriate **information** that can be **communicated** to the learning agents is of primary importance. Since any learning agent may represent its individual knowledge by means of a formalism that will – in the general case – not to be known to other agents, a common, agent-independent language must be used for communication during task negotiation, cooperative task execution, and, especially, learning. More important, the communicating agents must share a common **ontology**, i.e., any agent receiving a message must understand it exactly in the way it was meant. Especially during communication between pupil and teacher, this is a crucial point.

However, when designing such a language we experience the same dilemma that initially motivated the use of learning: If we have a known, agent-independent "area" that we want to describe by a language, both syntax and semantics of that language can be defined a-priori. KQML [4] is a good example of such a language that has been designed specifically to facilitate **communication** in a content-independent manner. If the area or function we want to describe is not a-priori given or sufficiently complex, a general-purpose language such as KIF [5] is a reasonable choice. Nevertheless, the problem of "meaning" still exists: Imagine an agent a joins a team and wants to inform the members of the team that it has a capability c. If the team members know c, and the new agent and the team share the same symbol for c, that's no problem. In the general case, however, c may be completely new for the team, the symbol used by a to describe c may mean nothing (or something different) to the team, and the symbols used by the team may mean nothing to a, such that it cannot translate its internal representation appropriately. All these things can happen despite a common communication language and a common content language. This problem, for which currently solutions are being sought via tools for creating, accessing, and maintaining ontologies [3], results therefore in another learning task, the task **to learn the meaning of symbols.**

Throughout this paper, we will analyze the learning tasks existing within a Multi-Agent System with respect to their requirements regarding the communication between the agents involved. We'll show that understanding each other is of primary importance during learning, and what understanding really means with respect to a specific learning task and scenario. Finally, we will present an approach to learn the meaning of symbols in a co-operative way, which exploits "non-verbalizable" knowledge within several agents and the language-related competence of a knowledgeable agent, should the latter exist. To provide the formal basis for these investigations, we'll employ state space models, following previous work by Beer [1] and ourselves [9].

2 Modeling Multi-Agent Systems

2.1 Single agent model

Our model of an agent a is that of a **skilled subsystem.** A single agent is able to perform **competent** actions that are related to its locally (possibly internally) defined goal and facilitate goal-oriented state transitions. More specifically, an important component of an agent a is the agent's strategy C_a^g with

$$C_a^g(\boldsymbol{x}) = \boldsymbol{u}. \tag{1}$$

$C_a^g(\boldsymbol{x})$ determines the goal-oriented action $\boldsymbol{u}$ that the agent executes if it is in state $\boldsymbol{x}$. To actually perform such a competent action requires **perceptual** capabilities (to determine $\boldsymbol{x}$), **cognitive** capabilities (to calculate $\boldsymbol{u}$), and **effectual** capabilities (to execute or apply $\boldsymbol{u}$, respectively). The action itself may be a physical one, such as the motion of a robot, or some "intellectual" effort, such as a request for data or a planning step. Similarly, the state $\boldsymbol{x}$ may include physically measurable information about the current situation of the agent with respect to its environment as well as internal, "mental" state variables such as estimations of the validity of internal models etc.

2.2 Multi-Agent System model

The Multi-Agent System is also a **skilled** system. Its actions – being compositions of the actions of all agents in the system – should be competent with respect to a system-wide goal. Therefore, a Multi-Agent System is modeled by a strategy function

$$C^G(\boldsymbol{X}) = \boldsymbol{U}, \tag{2}$$

where G denotes the current global goal of activity that is being pursued, $\boldsymbol{X}$ is the system's state and $\boldsymbol{U}$ is the calculated action.

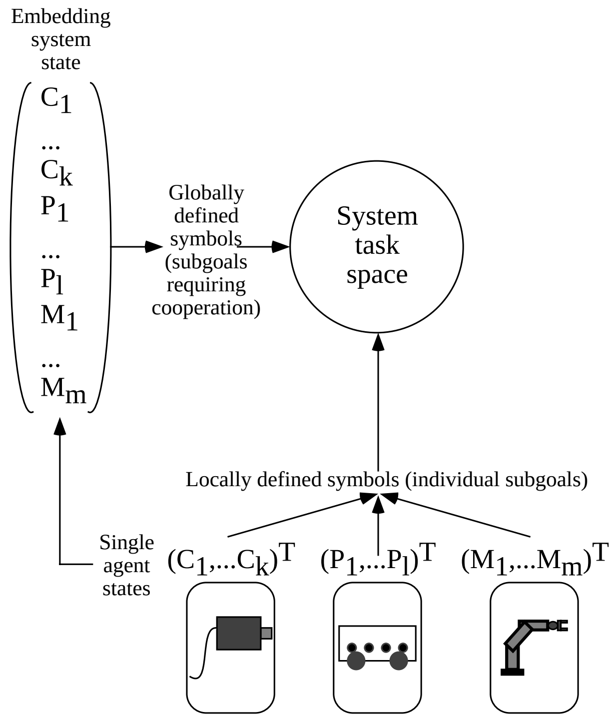

Figure 1: Example for the definition of the system's state vector, action vector, and task space on the lowest level of abstraction, i.e., by means of local, single agent related subgoals. Cooperation related subgoals require abstraction.

On the lowest level of abstraction, the state vector $\boldsymbol{X}$ of a Multi-Agent System is simply the collection of all state vectors $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}$ of all embedded agents $a_1, \ldots, a_n$. Also, its action vector $\boldsymbol{U}$ is built from the individual action vectors $\boldsymbol{u_1}, \ldots, \boldsymbol{u_n}$, such that

$$\boldsymbol{X} = (\boldsymbol{x_1}^T, \ldots, \boldsymbol{x_n}^T)^T$$

and

$$\boldsymbol{U} = (\boldsymbol{u_1}^T, \ldots, \boldsymbol{u_n}^T)^T,$$

respectively. Fig. 1 illustrates this situation for a system (a service robot) consisting of three agents, an active camera system C, a mobile platform P, and a manipulator M. However, in the general case a process of **abstraction** is necessary to maintain modularity and extendibility and to allow for coordinated actions. Abstraction, however, means to associate signals (state vectors, action vectors), clusters of signals or sequences of signals to **symbols** that have a common meaning for all agents within the system. In other words, abstraction is a process that is complementary to explanation required for learning the meaning of symbols, i.e., to explain the meaning of a symbol, we must reverse the process of abstraction. We'll exploit this duality later.

3 Learning and Communication in Multi-Agent Systems

The purpose of learning in a Multi-Agent System is to enhance the system by extending its general capabilities or by improving its performance with respect to a particular criterion. In terms of the Multi-Agent System model (represented via equation (2)), learning comprises to change an existing function $C^{G}(\boldsymbol{X})$ as well as to newly define functions $C^{G_n}(\boldsymbol{X})$ that enable the system to pursue a new goal G_n. Both tasks include several subtasks, such as learning in a single agent (isolated learning), learning to act as a team[1], building symbols from signals, and learning the meaning of those symbols.

3.1 Isolated learning

Learning within a single agent a means to alter the action $\boldsymbol{u}$ calculated via $C_a^g(\boldsymbol{x}) = \boldsymbol{u}$ from the state vector $\boldsymbol{x}$ or to build a new function $C_a^h(\boldsymbol{x}) = \boldsymbol{u}$ that enables the agent to contribute to a subgoal h.

In the first case, each learning agent obtains specific feedback that can be used to alter its action $\boldsymbol{u}$. In the most simple case, the optimal action $\boldsymbol{u}^*$ is communicated to the agent, such that **incremental supervised learning** may take place. [15] describes such a situation for an agent (a robot) that learns directly from user demonstrations. As in adaptive control, agents may also receive an indication of the direction $\boldsymbol{\Delta u}$ into which to alter their actions, instead of an optimal action [10]. Another prototypical setting is that of **reinforcement learning,** in which the agent receives a possibly delayed reward r as feedback (see, for example, [13]) and alters its actions in order to maximize the reward. This kind of learning requires **exploration,** i.e., the systematic alteration of the calculated action in order to estimate the optimal action.

In the second case, learning agents must discover that the instruction they get from the teacher is related to a new subgoal h. Following that, they can build a function C_a^h from these instructions. This setting is typical for transferring skills from one agent (e.g., a human) to another (e.g., a robot) [9].

3.1.1 Communication issues in isolated learning

In the isolated learning case, a single agent (the instructed agent or the "pupil") receives feedback from another agent. This agent may be an artificial one (a softbot), or a human supervisor. If learning takes place in a supervised manner, such that the given feedback consists of an optimal action or a quantitative indication of the error made by the agent, either

- the teacher must know the action space of the instructed agent and formulate the advice appropriately,

or

- the instructed agent must be able to map the teacher's advice onto its own action space.

[1]It should be noted that we use the term "learning to act as a team" instead of "team learning," in order to distinguish the "team learning" scenario (multiple agents trying to learn the same concept/language, as in [8]) from the situation considered here.

Both requirements are not trivial, especially if agents should learn from other agents that are not structurally identical.
For learning on the basis of a scalar reward, the situation is very similar. Either

- the teacher must know the range of possible rewards used by the instructed agent (i.e., what is the "good" and "bad" in terms of the pupil),

or

- the instructed agent must know the mapping between the teacher's reward and its own range of rewards (what does the teacher mean by "good" and "bad").

In all cases, teacher or pupil must initiate the learning process. An important requirement is also that the teacher knows the limits of the instructed agent, since it makes no sense to try to teach an agent to go beyond its maximum capabilities (e.g., to try to position a robot with a higher precision than it is capable of). To enable the teacher to take care of this aspect requires the teacher to query these limits from an agent and to correctly interpret the agent's answer. Similarly, the instructed agent must understand the teacher's request and relate its capabilities properly to the task specified by the teacher. Summarizing, the following communication-related activities are parts of an isolated learning task:

1. Initiation of the learning process (activation of the pupil's "learning engine"), including communication of the learning context (the agent-related subgoal g or a new subgoal h) to the pupil.
2. Request a description of the pupil's limitations with respect to the current learning task.
3. Communication of the pupil's limitations to the teacher.
4. Communication of a target value $(\boldsymbol{u}^*, \boldsymbol{\Delta u}, r)$ from the teacher to the pupil.
5. If the teacher requires a model of the pupil, communication of information about success/failure of the learning process from the pupil to the teacher.

3.2 Learning to act as a team

While learning to act as a team, the feedback available for learning is not related to the performance of a single agent but to that of a team of agents. I.e., learning aims at improving the performance of a team with respect to a common criterion. As in the isolated case, the setting that is easiest to handle is that of **incremental supervised learning.** Here, a teacher provides the optimal action that the team should take in each state. If team states and team actions can be directly mapped onto single agent states and actions, this scenario is equivalent to the one described in section 3.1, if also the function $C_a^g(\boldsymbol{x}, \boldsymbol{u})$ (the relevant subgoal) to be adapted is known for every single agent a. This is the case if we assume that task negotiation and agent coordination (the association of $g_{a_1}, \ldots, g_{a_n}$ to agents $a_1, \ldots, a_n$) have been done properly, i.e., based on symbols that correctly describe agent capabilities and constraints related to agent cooperation. As in the single agent case, agents a_i may also receive instructions to learn new functions $C_{a_i}^{h_i}(\boldsymbol{x})$ that are related to individual subgoals h_i or functions $C_{a_i}^{h}(\boldsymbol{x})$ related to team-specific subgoals h.
If only a scalar evaluation of the team's performance is available, the problem of **credit assignment** becomes evident. In contrast to isolated learning, which may incorporate the

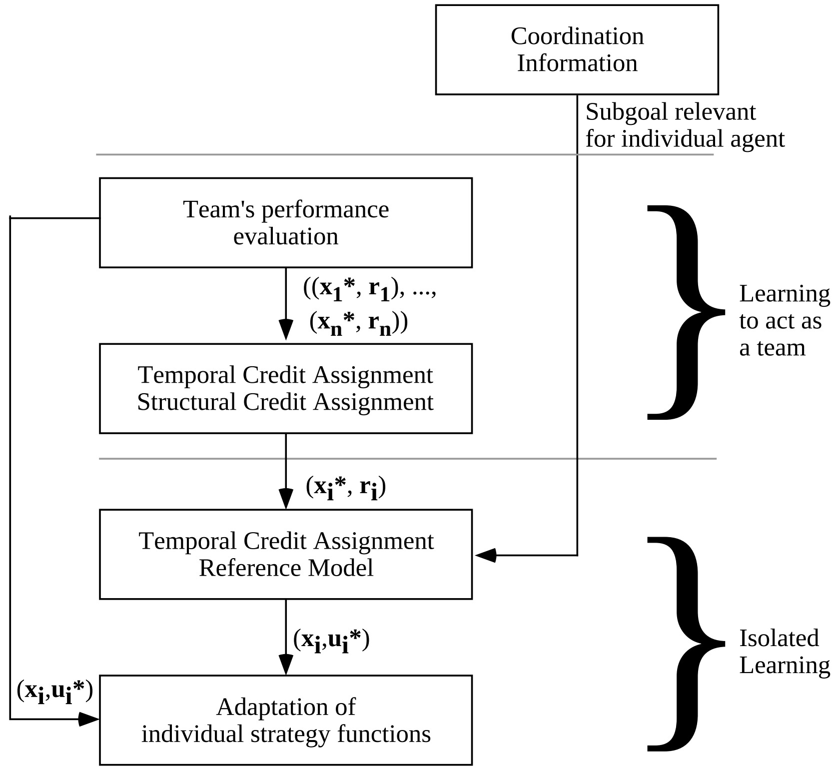

Figure 2: Reduction of a task of learning to act as a team to isolated learning tasks in both the supervised learning and the reinforcement learning scenario.

task of **temporal credit assignment,** i.e., the determination of those actions that are responsible for a delayed reward, learning to act as a team from feedback also involves a **structural credit assignment problem.** For each agent being a member of the team, its contribution (positive or negative) to the obtained reward must be determined. If agents are conscious about the performance of other members of the team, this process can be supported by requesting team members to provide individual evaluations.
Once credit assignment has taken place and subgoals g_{a_j} (or h_j or h, respectively) are known for any of the team members, team learning is reduced to several parallel steps of isolated learning (see Fig. 2 and, for example, [18]). It should, however, be noted that in some cases additional constraints are to be observed, since a single agent's environment (and, possibly, its target function) is continuously changing [16].

3.2.1 Communication issues in learning to act as a team

When learning to act as a team, the teacher instructs or evaluates a group of agents that cooperate towards a common goal g. Here, the same communication requirements as in the isolated learning case exist (see section 3.1.1). In addition, an agent a must be able to identify which strategy C_a^g should be subject to the taught changes. To facilitate this identification step, either

- the teacher is able to define an appropriate subgoal g_i for each agent a_i in the team or

- all instructed agents a_i are able to determine their respective subgoal g_i from the team-related goal g.

Consequently, agents must share a common terminology – symbols that describe team states, state-space trajectories, team actions, etc. Then, learning to act as a team comprises the following activities related to communication:

1. Negotiation for credit assignment (if supported).
2. Isolated learning for all team members.
3. If the teacher requires a model of the team, communication of information about success/failure of the learning process from each team member to the teacher.

3.3 Learning to communicate

The usefulness of communication depends on the ability of the communicating entities to understand each other. This is especially true for the communication between teacher and pupil, since the pupil performs self-modifications based on its understanding of the information obtained from the teacher.
In a Multi-Agent System, also an agent that coordinates a team of agents as well as agents cooperating in a team must be able to understand their respective counterparts – independent on the coordination/negotiation technique (see [19, 12] for examples) that is actually used. They must be able to understand what they are expected to do (or how they could contribute), and must be prepared to formulate their requests in an understandable manner.

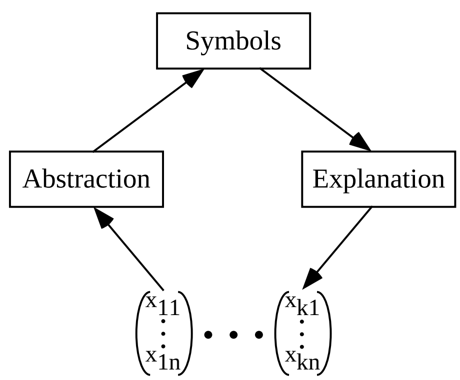

Figure 3: Duality of abstraction and explanation.

To understand the meaning of symbols used for communication, these symbols must be grounded on the representation primitives of each agent – its state and its action vector [14, 7]. Such symbols can be general, such that they describe phenomena that can only be observed within a team or on the system level and are only partially understandable for most individual agents. They may also be specific, such that they represent, for example, a capability of a single agent. In both cases, however, a symbol is only useful if it is understood by at least two agents, such that it can be used for coummunication purposes. Consequently, it is not sufficient to provide means for building symbols (via abstraction), it must also be possible to explain the meaning of symbols to agents (Fig. 3).

3.4 Abstraction: From signals to symbols

Learning to map signals onto symbols adds a new dimension to the relationship between teacher and instructed agent. A new symbol affects all agents within the Multi-Agent System, since it requires possibly all of them to **extend their vocabulary.** The targeted construction and invention of a new symbol is an extremely difficult task, and the individual problems discussed in the following are in general not yet solved.
Nevertheless, we can identify three situations that require to build a new symbol:

1. The activities of an agent or a team of agents yield a system state (or state-space trajectory) that is considered to be characteristic or useful, i.e., that represents a subgoal.

2. For specifying a task, a new subgoal (to be pursued by a single agent or a team) must be defined that is not yet represented by a symbol.

3. An agent or a team develops a new capability (or a new agent with a new capability joins the Multi-Agent System) that is useful to know and to use when specifying and negotiating about tasks.

Whenever such a situation occurs (this can possibly only be detected by a knowledgeable agent, such as a human user), the symbol to be defined will be grounded on the current state (or state-space trajectory) of the system, the respective team, or the respective agent. In this sense, the symbol represents a concept that is to be learned in a supervised manner, for example by methods such as those described in [6, 11].
In addition, the Multi-Agent System's task space must be extended by the new subgoal. Task-specifying external agents and the Multi-Agent System itself must incorporate the new symbol into their task-description language. To explain the meaning of the new symbol to those agents that did not participate in the grounding process, this symbol must be translated to those agents, as described in the next section.

3.5 Explaining the meaning of symbols

The communication activities related to the task of symbol learning differ significantly from those occurring within other learning tasks. This is due to the fact that – following the creation of a new symbol – we can't simply use that symbol for self-explanation. Instead, we must **explain** what it means, i.e., we must circumscribe its meaning in already well-established terms.
To perform this task, two possibilities exist. First, we may be lucky and find an agent within the Multi-Agent System that understands both the symbol to be explained and the language of the agent that needs explanation. For example, a human supervisor will almost certainly be such a knowledgeable agent that may explain the meaning of the newly generated symbol to the learning agent, e.g., via direct implementation.
Second, it might be possible to choose another communication channel different from the verbal/textual one to explain the symbol. In analogy to humans, this channel may be something like the gestural-visual one. The meaning of a symbol is explained by demonstration, i.e., by performing the actions or establishing the state (or state-space trajectory) that is represented by the symbol. Since there are agents on whose states, state-space trajectories, or actions the symbol has initially been defined, these agents can literally show

what the symbol means, while the agent needing explanation observers and analyzes this demonstration.
In principle, this mechanism works in two directions: Agents within a Multi-Agent System can perform actions that are observed by an agent that has just joined the system, in order to explain the meaning of symbols used for communication to that agent. Also, a new agent can demonstrate the meaning of its own symbols to the members of a team or a Multi-Agent System, in order to make himself understandable to those.
Obviously, observing other agents' actions and matching the observations against one's internal representations is not a trivial task. However, many techniques developed within the framework of Programming by Demonstration [2] may prove useful to support it. After all, the idea to explain the meaning of symbols ("words") in a non-verbal manner is applied very successfully in everyday human communication.

4 Summary and Conclusion

Throughout this paper, the individual learning tasks existing within Multi-Agent Systems have been analyzed with respect to their requirements regarding the communication between teacher and instructed agent(s). To support this analysis, a framework based on system theory and state-space models has been employed.
Learning requires communication and depends crucially on a mutual understanding between teacher and instructed agent. Within the developed framework,

1. the need for an ontology that is common to the teacher and the instructed agent, and
2. the need to extend this ontology in case agents or teams of agents develop new capabilities, in case additional agents join the Multi-Agent System, and for the specification of and negotiation about tasks unknown so far,

have been made explicit. The necessity to explain newly generated symbols to agents has been identified, and a proposal to realize such explanation via "demonstration and observation" has been presented.
As a conclusion, we have seen that communication and the generation of symbols for communication are key components of learning, and, especially, of learning in Multi-Agent Systems. To develop communication mechanisms that meet the requirements defined by learning agents is a challenging task. However, its solution is mandatory to develop a general understanding of and a general framework for learning in Multi-Agent Systems.

Acknowledgement

This work has been performed at the Institute for Real-Time Computer Systems & Robotics, Prof. Dr.-Ing. U. Rembold and Prof. Dr.-Ing. R. Dillmann, Department of Computer Science, University of Karlsruhe, Germany. The authors would like to thank the anonymous reviewers for their helpful comments.

References

[1] R. D. Beer. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72:173–215, 1995.

[2] A. I. Cypher. *Watch what I do – Programming by Demonstration.* MIT Press, Cambridge, Massachusetts, 1993.

[3] A. Farquhar, R. Fikes, W. Pratt, and J. Rice. Collaborative ontology construction for information integration. Technical report, Stanford University, 1995.

[4] Tim Finin, Rich Fritzson, Don McKay, and Robin McEntire. KQML - A language and protocol for knowledge and information exchange. Technical Report CS-94-02, Computer Science Department, University of Maryland and Valley Forge Engineering Center, Unisys Corporation, Computer Science Department, University of Maryland, UMBC Baltimore MD 21228, 1994.

[5] M. R. Genesereth, R. E. Fikes, et al. Knowledge interchange format, version 3.0 reference manual. Technical Report Logic-92-1, Computer Science Department, Stanford University, 1992.

[6] J. H. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. *Artificial Intelligence*, 40:11 – 61, 1989.

[7] Stevan Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[8] S. Jain and A. Sharma. Team learning of formal languages. In *Agents that learn from other agents: Proceedings of the ICML'95 Workshop*, Tahoe City, California, 1995.

[9] M. Kaiser and R. Dillmann. Building elementary robot skills from human demonstration. In *IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, USA, 1996.

[10] M. Kaiser, A. Retey, and R. Dillmann. Designing neural networks for adaptive control. In *IEEE International Conference on Decision and Control (34th CDC)*, 1995.

[11] Volker Klingspor, Katharina Morik, and Anke Rieger. Learning concepts from sensor data of a mobile robot. *Machine Learning*, 1996.

[12] Th. Längle, T. C. Lüth, and U. Rembold. A distributed control architecture for autonomous robot systems. In T. Kanade H. Bunke, H. Noltemeier, editor, *Modelling and Planning for Sensor Based Intelligent Robot Systems.* World Scientific, 1995.

[13] L. J. Lin. *Reinforcement learning for robots using neural networks.* PhD thesis, Carnegie Mellon University, School of Computer Science, 1993.

[14] C. Malcolm and T. Smithers. Symbol grounding via a hybrid architecture in an autonomous assembly system. *Robotics and Autonomous Systems Special Issue on Designing Autonomous Agents*, 6(1,2), 1990.

[15] P. Reignier, V. Hansen, and J.L. Crowley. Incremental supervised learning for mobile robot reactive control. In *Intelligent Autonomous Systems 4 (IAS-4)*, pages 287 – 294. IOS Press, 1995.

[16] J. Schmidhuber. A general method for multi-agent reinforcement learning in unrestricted environments. In S. Sen, editor, *AAAI Spring Symposium on Adaptation, Coevolution and Learning in Multiagent Systems.* AAAI Press, 1996.

[17] S. Sen, editor. *AAAI Spring Symposium on Adaptation, Coevolution and Learning in Multiagent Systems.* AAAI Press, 1996.

[18] K. T. Simsarian and M. J. Mataric. Learning to cooperate using two six-legged mobile robots. In M. Kaiser, editor, *Proceedings of the 3rd European Workshop on Learning Robots (EWLR-3)*, Heraklion, Crete, Greece, April 1995.

[19] R. G. Smith and R. Davis. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1):61–70, 1981.

[20] G. Weiss and S. Sen, editors. *Adaptation and Learning in Multi-Agent Systems.* Springer-Verlag Berlin, Heidelberg, New York, 1995.