Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

**Technical
Memo**
TM-91-06

Aspects of Cooperating Agents

Johannes Stein

September 1991

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum
für
Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988 by the shareholder companies ADV/Orga, AEG, IBM, Insiders, Fraunhofer Gesellschaft, GMD, Krupp-Atlas, Mannesmann-Kienzle, Siemens-Nixdorf, Philips and Siemens. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct systems with technical knowledge and common sense which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

☐ Intelligent Engineering Systems
☐ Intelligent User Interfaces
☐ Intelligent Communication Networks
☐ Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

Aspects of Cooperating Agents

Johannes Stein

DFKI-TM-91-06

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-8903 0).

Aspects of Cooperating Agents

Johannes STEIN

DFKI (German Research Center on Artificial Intelligence)
Research Group AKA
Postfach 2080
D-6750 KAISERSLAUTERN
GERMANY

Abstract

An overview on aspects about cooperating agents is presented. As multiagent systems are various, we start with a classification of multiagent systems which is particularly influenced by an article from Decker, Durfee, and Lesser [Decker& 89]. In the following the aspects communication, planning, and negotiation are examined. On the occasion of communication, the discussion is split into: no communication - simple protocol - artificial language. The discourse on planning is broken into sections: from classical to multiagent planning - a general multiagent planning theory - intention - intention-directed multiagent planning. Finally, a summary of Brigitte and Hassan Lâasri and Victor Lesser's negotiation theory will be presented.

1. Introduction

This report deals with the aspects of cooperation in *multiagent systems*. Multiagent systems form one direction in the research in *distributed artificial intelligence* (DAI); they have to be distinguished from *blackboard systems*, where, in contrast to multiagent systems, agents do not have autonomy.

Multiagent systems can be defined as societies of *agents* working together in order to achieve common goals. Agents are tried to define in many different ways. From a theoretical point of view, those definitions are best subsumed in the model of a finite non-deterministic automata. On the other hand, from a more practical standpoint, agents models are distinguished in two basic aspects: an *object-oriented* aspect and the agent´s *role*.

The object-oriented standpoint arises from the principle that the agent adopts information from outside which makes him acting. Note that this means more than pure message passing, because adopting information may imply receiving as well as perception, while acting may imply sending a message or affecting the environment by the agent's actuators (e.g. an arm of a robot).

The role-oriented aspect has to do with the agent´s behavior guided by his intentions. The notion of role grounds in social science, but even there, it lacks from a satisfying definition.

As stated above, we assume that agents work together to achieve common goals. Making this assumption, we already exclude some other issues to multiagent systems, because agents may also work against each other, thus, implying completely different problems which will not be considered in this discourse.

As the object- and role-oriented view of an agent are not very restrictive, the obtained models are various. Consequently, the resulting problems are various as well and so, a study on the cooperation in multiagent systems must integrate issues differing particularly in the agent models in the systems. Therefore, in a first step, we have to be aware what more precisely a multiagent system is and how to point out the differences between those systems. This will be the content of chapter 2.

Afterwards, we want to talk about the specific problems concerning multiagent systems, especially, problems of highly developed agents such as *communication* (chapter 3), *planning* (chapter 4), and *negotiation* (chapter 5).

2. Classifying Multiagent Systems

In this chapter, we try to give a structured overview on multiagent systems. The discussion is strongly influenced by [Decker& 89], but instead of representing the content of this paper, we tend to lead a more global discussion in order to point out some key problems of DAI. Further, we will demonstrate how these key problems could be examined in detail following some more precise criteria.

2.1. Hierarchical and Shallow Systems

The most obvious difference between multiagent systems is that some of them enclose several different abstracted layers (*hierarchical*) and other ones consist just of one layer (*shallow*). If we talk about hierarchical systems, three kinds of *architecture* must be distinguished:

- tree-like systems,

  where the idea is that the behavior of a society of agents emerges as the behavior of one agent on a more abstract layer; in the same manner, such higher level agents can be also grouped, thus realizing behavior on a more abstract level, and so on [Hultman& 89];

  a similar idea lays down the subsumption architecture: like operating systems, the higher levels are built up by the lower ones, and thereby, more qualified reasoning facilities can be added [Brooks 91];

- general purpose systems,

  where each purpose is realized in a fixed number of layer agents; several purposes can be integrated into the (open) system, and communication can take place, horizontally, between the agents on the same level, and vertically, between the agents pursuing the same purpose [Boissier& 91];

- market-like systems,

  where the former approaches are combined following the idea that singleton organizations specify producers of a certain product (they pursue a certain purpose) while the organizations themselves define tree-like systems; in order to achieve goals, the organizations compete among task-specific measurements as cost, duration, and size [Fox 89].

For the rest of this chapter, each layer of an hierarchical system should be considered as a shallow system, i.e. all agents on the same layer in, for instance, the general purpose system form together an organization which can be examined employing criteria as presented below.

2.2. Reactive and Rational Systems

Although it seems to be an evident difference between multiagent systems, the criteria to distinguish between reactive and rational systems does not exist in [Decker& 89]. Instead of this, the authors talk about agents which "react in a rational manner". Apparently, systems cannot be broken into reactive or rational ones in a binary choice, so this difference has to be explored under some more detailed viewpoints; some examples:

- The agents' behavior may be hardcoded into their architecture
- The agents' intentions may be changing
- The agents may have knowledge about the other agents and their environments; this knowledge may be dynamic

Refering to [Burmeister& 91], agents are described as a triple <I, R, B> with…

I: the intentions of an agent

R: the resources used by the agent[1]

B: knowledge of the execution of actions

Let us call the triple <I, R, B> the *internal state* of an agent, then our criteria might all aim at the same question

"How dynamic is the agent's internal state?"

We want to give some possible answers to this question; of course, these answers can differ with respect to the part of the internal state <I, R, B> in mind. Here, some examples:

- the internal state is hardcoded/compiled into the agent's architecture
- the internal state is initialized firmly in the knowledge base
- the content of the knowledge base is fully dynamic and can change any time using communication facilities

This view on autonomous agents is well suited in the design tool RATMAN [Bürckert& 90]. However, in contrast to the approach in [Burmeister& 91], the agents developed in RATMAN provide a much more sophisticated structuring for the internal state. They are developed on different layers between sensoric and learning capabilities. Obviously, if the centre of gravity of an agent architecture is on the sensoric layer, then the internal state following the former definition is more hardcoded and the agent's behavior is more reactive. On the other hand, if agents are developed providing powerful learning capabilities in order to achieve rational behavior, then more dynamic knowledge bases are needed.

Some other questions are related to the differentiation between reactive and rational systems, and we want to complete as follows:

- determinism

  As reactive agents are ideally completely hardcoded, they have to act deterministically where rational agents' behavior depends to a greater extend on previous events.

---

[1] other agents are considered as resources also

- locality of the sphere of action

  Does an agent know something about the entire world or does he depend on his perception exclusively?

- number of agents

  Is a given problem shared by few or many agents?

  Consider a complex problem. If the problem is distributed among a small number of agents, then everyone is still concerned with a rather complex subproblem (rational), but if there are a lot of agents, then there is a good chance that the individual problems are quite simple (reactive).

- composition versus decomposition

  What is the kind of goal we want to accomplish? Is it a distributed problem or do we want to distribute one single overall goal among the agents?

  In the latter case, the goal is achieved by decomposing it into subgoal that are distributed among the agents. After solving the local subproblems, the subgoals are recombined to form the overall goal [Georgeff 89a]. If the problem is of a distributed nature, then the steps of decomposition and recombination are not necessary; in this case, we only have to care for the good coordination of the local agent processes [Durfee& 89].

  As distributed problems are closer to reactive systems, the problem solving approach by decomposition is more the idea of rational agents.

And two further criteria that deal with the purpose of DAI:

- response time on receiving and sensing

  Can information be treated in real time?

  Of course, this depends on the kind of demand; the reaction time can be limited to less than a second, and in other cases, it can take several minutes. In general, fast response is the main purpose of reactive systems.

- coordinated joint actions

  A society of agents can fulfill tasks which are too hard for a single agent. For instance, two robots can lift something which is too heavy for each one of them. They have to recognize and to overcome this problem. Hence, this is the main purpose of rational systems.

2.3. Cooperation

In the following, we will examine the cooperation between agents. From a global point of view, *positive* and *negative* cooperation types are to be differentiated. Starting from the usual purpose of multiagent systems, we are more interested in the case of positive cooperation where agents work together towards a common goal. Mostly, the different agents cannot act independently, so some kind of synchronization has to be established. In the case of negative cooperation, we face agents in competition where an agent also tries to destroy the advantages of another one. As stated in the introduction, we want to exclude situations of negative cooperation in this report.

Concerning positive cooperation, one should further distinguish between cooperation with or without communication [Genesereth& 89], where communication may involve facilities from a simple protocol [Smith 89] up to an artificial language [Cohen& 90, Huhns& 90].

The cooperation mode of a system is particularly concerned with its classification. This is obvious, because the power (with respect to the real time constraints) of purely reactive systems relies on the fact that there is no need to communicate at all, where on the other side, the power of rational systems relies on the capability to coordinate agents' joint actions. Therefore, a general design tool for agents of a multiagent system like RATMAN [Bürckert& 90] must provide facilities to choose the most appropriate cooperation mode out of a variety of possible ones.

2.4. Homogeneous and Heterogeneous Agents

Mostly, it is easy to decide whether the agents of the system are homogeneous or heterogeneous. Nevertheless, if agents are heterogeneous in some respect, it may not be satisfactory to answer that the agents are not homogeneous, because, in fact, it is more the degree of heterogenity we are interested in. Agents can be rather similar and they only appear very heterogeneous because of the arbitrary roles they play. On the other hand, agents can be completely heterogeneous, if the system's conception is to combine complex agents with very specific capabilities. This difference is important, because it strongly affects cooperation: if the agents are very heterogeneous, they need to cooperate by a common language [Huhns& 90]; in the other case, simpler cooperation modes can be thought of [Smith 89].

2.5. Control

Finally, systems can be differentiated by their control We have to distinguish between…

- no control,
- centralized control,
- distributed control.

If the agents' roles are completely different, then they can act independently and there is no need of control. Furthermore, if communication facilities are not available, then all decisions are made locally without any control instance [Genesereth& 89]. If control is needed and if this control is distributed, then the agents negotiate (chapter 5) to achieve good coordination of their actions [Kreifelts& 90]. If the coordination is handled by a centralized approach, then we have to distinguish between the case when the manager who coordinates the agents' actions is always the same [Georgeff 89a] and when the manager role is assigned to agents dynamically [Cammarata& 89].

3. Issues to Cooperation

This chapter will give some issues to cooperation and the way it is concerned with communication. I will present, as follows, cooperation without communication, cooperation by simple communication facilities and by an artificial language. As already stated in chapter 2.3., we need to be aware about what kind of multiagent system we talk and when the one technique can or must be applied and when another one should be chosen. In any case, it has to be mentioned that the approaches presented below are only some examples to the idea pursued in the chapter.

3.1. Cooperation without Communication

Our discussion of cooperation without communication grounds on [Genesereth& 89] where the conception is implicit which has to be distinguished from the explicit issues for very reactive systems.

This approach aims at cases in which communication between agents is not possible because of, e.g., breakdown or absence of communication equipment. Furthermore, the analysis could contribute to a benevolent decrease of communication activities which can be very costly.

Despite the lack of communication, it is assumed that there is enough (sensory) information available to recognize the intentions and plans of other agents in the environment.

The basic idea in this paper is that the utility of an action is a function of the action itself and the actions of the other agents. This function can be described in the so called payoff matrix. Consider two agents a1 and a2 who can perform actions of a set S1 or S2. Then the payoff matrix will be a 2-dimensional table over S1 and S2 where, in position (m, n), 2 real numbers indicate the utility for agent a1 of action m when a2 performs n, and the utility for agent a2 to perform action n when a1 executes m[2].

The payoff matrix allows one to make rational decisions about the actions to execute. Therefore, the dominance of one action over another needs to be decided. This can be done by comparing the payoffs that follow from the choice of an action and the other agent's expected reaction (which is an action also). This kind of decision making is called *basic rationality*.

Some simple examples like complete independence of agents' actions, the authors are used to validate the correctness of their basic rationality definition.

---

[2] Those two values are not the same, in general.

Afterwards, the notion of *general rationality* is introduced. The definition is rather similar to the basic rationality definition. The aim of general rationality is to assess the goodness of various decision functions. An assessment is achieved by comparing the resulting scopes in the payoff matrix. The advantage of general rationality is that it allows one to eliminate joint actions in favour of other joint actions, thus giving an implicit best plan.

3.2. Simple Communication

This chapter will talk about simple communication to enable cooperation between agents. we want to figure out the principal aspects of a communication protocol as presented in [Cammarata& 89], the messages used in the Contract Net Protocol [Smith 89], and some special messages for communication involving negotiation [Kreifelts& 90]. Finally, we want to come back to [Cammarata& 89] where cooperation with simple communication is demonstrated by an example within the air traffic control domain. Several alternatives will be presented.

3.2.1. The Requirements of Simple Communication

Communication between agents must be considered under two points of view:

- a technical view

    the focus of attention concerning the technical point of view is on the following issues:

    * should agents be addressed directly (agent-agent-communication) or in common by broadcasting;

    * should new information be propagated over the net or should (new) information be transmitted when required;

- an organizational view

    this has to do with questions like…

    * when does the organization arise?

    * how is task assignment achieved?

    * are agents addressed in a goal- or data-directed[3] manner?

    * are agents allowed to negotiate about the assignment of a task?

---

[3] goal-directed means that the agents receives a message which makes him perform a certain task, where data-directed means that the agent handles a value by the way he is designed.

3.2.2. A Simple Communication Protocol

We will introduce the *Contract Net Protocol* [Smith 89] as an example of simple communication. The idea behind the protocol is that the joint actions of a society of agents are coordinated by a *manager* who sends messages to his potential *contractors* announcing that there are some jobs to distribute among them. The *announcement* can be addressed to all agents (broadcast) or to only one of them (agent-agent communication). After that, all the agents who received a message check whether they are able to contribute to the solution. If this is the case, they answer with a *bid* where they specify the kind of contribution. Now, the manager, who collected all those responses in a queue, decides what agent should perform which task and informs him. In other words, there are three basic messages in the protocol: the task announcement, the bid, and the task award. These messages and their important features will be presented below. It should be mentioned that, depending upon the application, there are a lot of internal formats that need to be fixed by the designer.

- task announcement

  features are…

  * abstract task description

  * eligibility specification

  -> This is to specify the condition under which a potential contractor can bid.

  * bid specification

  -> This to declare the format of a bid.

  * expiration time

  -> This is to define a deadline for bids.

  * immediate response

  -> If this flag is set, the receiver has further options to respond.

- task bid

  features are…

  * node abstraction

  -> This is the bid: following the format declared in the task announcement, the agent makes some propositions of how he could contribute to the solution of the problem.

  If the immediate response flag was set in the announcement of the task, the agent can also answer with BUSY, INELIGIBLE, and LOW RANKING, where BUSY means that the node is too busy to answer in a short response time, INELIGIBLE means that the agent will be overconstrained together with his other jobs, and LOW RANKING means that the task has been accepted, but it is put on the queue with low ranking, so that a fast execution is not expected.

- task award

  features are…

  * task specification

  -> These are the actions the manager has chosen out of the actions proposed in the bid

Despite the messages I introduced above, there are still some more that I will introduce without a detailed discussion of their attributes.

- request message

  -> This is a request of information.

- information message

  -> This is to transmit information.

- report message

  -> This is to inform about the actual state of execution (intermediate or final).

- termination message

  -> This is to force an agent to stop his actual process.

- acknowledge message

  -> This is to make a contractor answer whether he intends to accept his job or not.

- node availability message

  -> This is to prevent announcements of tasks that cannot be fulfilled because all agents are too busy. Therefore, the manager sends a node availability message over the net by broadcast, and waits for an agent to respond. Only agents who fulfil a certain condition and who are not busy can respond to this message.

Those are all the messages!

3.2.3. Special Messages for Negotiation

In [Kreifelts& 90] there are some further special messages to enable negotiation. The most important ones are presented below:

- action

-> An agent propagates his plan in order to get it confirmed, modified, or rejected

- approval

-> Confirmation of a plan

- proposal

-> A plan announced before is returned modified

- counter

-> Some parameters of a previous plan need to be modified

- rejection

-> A previous plan is rejected

There are still many very specific messages which are not mentioned here. Furthermore, all those messages depend on the actual state of the contractor. These states refer to the negotiation process, they can be, e.g., initial, planned, unresolved, committed, terminated, among others. In fact, from this technical point of view, negotiation is rather complex, so a short overview on all the messages and states, and on how these messages affect these states is impossible. Moreover, it is not the purpose of this chapter to give an introduction to negotiation.

3.2.4. Examples

Finally, I will present some few examples about how cooperation among agents can be achieved using a simple communication protocol. The four examples are established in the air traffic control (ATC) domain where each aircraft approaching the airport under control can be represented as an agent. The conflicts are evidently the collision of two aircrafts. Moreover, conflicts can arise from further constraints, such as fuel reserves. We consider the following alternatives:

(1) centralized control, no information

The organization arises when there is an aircraft which becomes aware of another one. As this aircraft is the first to have the perception, he becomes the manager of the organization. The manager knowing about his own intentions will generate a plan for the other aircraft and will send it to him. Now, the other one adopts this plan and changes his own plan in a way that collisions can be avoided.

This solution takes no account of the load of the aircrafts. Thus, some agents can be driven into overconstrained situations due, for instance, to a loss of fuel. Furthermore, it can happen that it is always the same aircraft which changes his plans; it would be better to keep the loads of the different agents balanced.

(2) centralized control, explicit information about resources

This alternative solves the lack of information concerning the load of the aircrafts. This alternative is rather similar to the first one. Here, an aircraft becoming aware of another aircraft transmits his load to the other one. This one will react in the same way, so that, afterwards, both will have the information of the load of each other. In consequence, they can decide on who is the most constrained agent, and therefore, shall should be manager in this situation which means that it will be he who propagates his plan.

This solution allows safe landing of the aircrafts, but this way to achieve planning is not very satisfactory, because the aircraft with the highest load is not necessarily the best for planning. In general, it would be better, if the agent who is the most familiar with the intentions of all the aircrafts in the environment should perform planning in order to avoid avalanches of newly created conflicts.

(3) centralized control, explicit information about resources and implicit information about intentions

As criticized above, aside from the load factors, the agents also exchange knowledge factors, so that the most competent agent can be chosen for planning.

This solution has one big problem; that is, the intentions of the other agents must be deduced from their current behavior. In the case that one agent changes his behavior in an unexpectedly manner, all previous planning can become invalid and must be modified or recalculated.

(4) distributed control, explicit information about resources and implicit information about intentions

In this solution, we start by exchanging load and knowledge factors as described in the previous case. After that, the most suitable (following the criteria mentioned before) agent generates and transmits a plan. Now, the receiver checks this plan against his own intentions, and if there is disagreement, he sends back a modified plan, and so on.

This solution is highly dangeous, because it leads to an enormous increase in communication load, and that there is no guarantee that planning will ever succeed. In any case, as communication is very expensive, this cannot be an efficient solution to the problem.

3.3. Cooperation with an Artificial Language

The means of creating an artificial language is to achieve communication for cooperation among highly developed agents. This approach was first mentioned in articles presenting a theory of intelligent agent cooperation, especially, multiagent planning [Werner 89, Cohen& 90]. In those approaches, the main idea is that a good plan for a society of agents depends essentially on the interchange of agents' intentions, so that the notion of a simple communication has to be dropped due to the lack of flexibility of the underlying protocols.

As mentioned in chapter 2, it is not obvious how one distinguishes an artificial language from simple communication in the sense of protocols when only information for planning is exchanged.

One step further towards the creation of artificial languages is presented in [Wittig 90] and [Huhns& 90] which aim at combining existing expert systems to construct a more powerful overall system. In these approaches, the need of an artificial language is obvious, because the representations of knowledge (facts and actions) and intentions of the different systems can be rather different, thus, coordination can be achieved only by "conversation". And in fact, one of the basic ideas of these projects is the development of an artificial language.

In [Huhns& 90] the translation problems are solved introducing the language RAD. By the example of OPS5, the authors show how one translates commands of the standard network language RAD into commands of design languages for expert system and vice versa. They enclose every single system into a module called *communication aide* enabling the interchange of commands and information with other systems using RAD.

However, transfering knowledge from one node to another cannot be handled in both projects.

4. Planning

We want to discuss four papers about planning in multiagent systems. The first article [Georgeff 89a] presents an idea for the transition from classical planning towards multiagent planning. Afterwards, we want to introduce Georgeff's theory of multiagent planning [Georgeff 89b] and Cohen and Levesque's notion of intention [Cohen& 90]. Finally, we want to show how Rao [Rao& 91] fused former ideas to intention-directed multiagent planning.

4.1. From Classical to Multiagent Planning

In this approach [Georgeff 89a], an individual plan is defined as the sequence of actions. A rough view on how a suitable overall plan is created to accomplish a common goal is given. The method comprises four basic steps:

(1) The goal is decomposed into subgoals.

(2) The subgoals are distributed among the agents.

(3) The agents create local plans to achieve their individual goals.

(4) The local plans are synchronized.

The main problem of synchronization is to recognize *unsafe regions*, because within those regions the order of actions might misfit or deadlocks can occur. The following rules are given to decide if a region is unsafe or not. Let ai and bj be actions and P=a1..am and Q=b1..bn be individual plans. Then unsafe regions can be detected by looking for unsafe states:

(* order *)

(1) <begin(ai), begin(bj)>, if ai and bj do not commute;

(2) <begin(ai), end(bj-1)>, if ai is not allowed to happen before bj;

(* deadlock *)

(3) <begin(ai), begin(bj)>, if all successing states are unsafe;

(4) <begin(ai), end(bj)>, if <end(ai), end(bj)> is unsafe;

(5) <end(ai), end(bj)>, if all successing states are unsafe.

Checking a state to be safe or not is very expensive, because it relies on checking all successing states (rules 3-5). Therefore, a more efficient test based on two rules is presented:

(1) the region is safe, if <end(am), end(bn)> is safe.

(2) the region is unsafe, if <begin(a1), end(bn)> and <end(am), begin(b1)> are unsafe.

When unsafe regions are found synchronization can be established by inserting two primitive actions:

(1) Receive State from Process. (? s p)

(2) Send State to Process. (! s p)

Note that this is the step from classical planning to multiagent planning. In the classical approach we also start by decomposing a goal and associating subplans to subgoals, but we do not have a problem of synchronization, because actions are performed sequentially. And so, instead of synchronizing actions, we only have to check the preand the post-conditions of the actions and put them in the right order to avoid violation of conditions.

4.2. A General Theory of Planning in Multiagent Systems

The main idea of this theory [Georgeff 89b] is that each agent in the multiagent world is associated with a process, and so, a theoretical model of process must be developed. This is done in two steps: first, the process model of a single agent is created; second, two agents' process models are combined to one common process model (this is the idea of cooperation).

4.2.1. A Single Agent Process Model

A process is a 7 tuple consisting of…

(1) S, a set of world states;

(2) F, a set of atomic actions;

(3) C, a set of control points (where conditions are to be checked);

(4) ∂, a function to associate control points to actions in order to decide if those are suitable or not;

(5) P, a function which decides the set of reachable worlds given a control point c;

(6) and (7) initial and final control point of the process.

4.2.2. A Multiagent Process Model

Considering the single agent process model (4.2.1.), two (parallel) process models can be combined into one common process model as follows:

(1) S is the set of world states as before;

(2) $F = F1 \cup F2$;

(3) $C = C1 \times C2$ (the control points are pairs of control points);

(4) For the association of a control point to an action each agent follows his own control points (agent1 C1/ agent2 C2), where the other component of the joint control point stays unchanged;

(5) $P(<c1, c2>) = P1(c1) \cap P2(c2)$[4] ;

(6) <ci1, ci2> (the common initial control point);

---

[4] The intersection explains why there may be a deadlock. In the case that, for instance, each agent has allocated a resource, they can block each other by the conditions specified in the control points and no more actions can be executed.

(7) <cf1, cf2> (the common final control point).

Note: This general theory of multiagent planning represents an important basic issue to the development of distributed planning systems. The idea that joint behavior of a society of agents emerges as the behavior of one high-level agent is consequently pursued in the issue presented in chapter 4.4., thus, realizing a planning system conform to the theory as just introduced.

4.3. Intention

The good behavior of highly developed agents depends on the rational balance of their cognitive basic features as belief, goal, plan, intention, commitment, and action. There are a lot of papers where the authors mention something like intention-driven behavior of agents (e.g. [Cammarata& 89, Werner 89, Singh 90, Rao& 91]), but only less work aims at really formalizing the notion of intention.

A nice introduction to this notion was given in the article from Cohen and Levesque [Cohen& 90]. As intention has become a very important aspect for distributed planning, we want to give a short summary of this article. In fact, [Cohen& 90] aims at much more than just the formalization of intention, but we want to limit our attention to only this point. Concerning the intention notion an intuitive and a philosophical issue, and finally, a formal definition will be presented.

4.3.1. An Intuitive Issue.

Intention can be determined in the following five rules:

(1) Adopt suitable intention

(2) Keep intentions, but not forever

(3) Drop satisfied intentions

(4) Change your intentions when you change your beliefs

(5) Adopt subsidiary intentions for planning

As shown above, intention is rather near to the notions plan and goal. This motivates an operational formalization of intention, but due to the following problems, the idea of considering intentions as the content of plans has to be dropped.

> (1) In classical approaches, plans and actions are separate, but what we want is that, by commitment, planning leads immediately to action.
>
> (2) Normally, an intention is more abstract than the plan to fulfil it.
>
> (3) Reasoning on an operational structure is rather complicated.

4.3.2. A Philosophical Issue.

In some philosophical approaches [Bratman 87, Searle 90], the notion of intention is reduced to belief and desire (in an AI terminology goal). Furthermore, present- and future-directed intentions (in other papers introduced as short- and long-term intentions) can be distinguished, where present-directed intentions lead to commitment immediately. Despite that, Searle [Searle 90] points out that intention is self-referential which means that someone who really intends to do something will act to fulfil this intention.

Finally, going back to an idea of Bratman [Bratman 87] who defines intention as a choice (which depends on his believes and goals) and commitment, we get the simple formula:

Intention = Choice + Commitment

In order to settle intention in a formal system, Bratman proposes rules as follows:

(1) Agents have to point out ways to accomplish their goals.

(2) Adopting an intention must not conflict with the constraints of other intentions of the agent.

(3) Agents must "track" the success of the execution of their actions, and in case of failure, replanning should emerge.

…and 4 criteria for choosing the best intention i:

(4) The agent believes i is possible.

(5) The agent does not believe that his plan will fail.

(6) The agent beliefs, he will achieve i under some condition.

(7) The agent need not intend all side-effects of his intention.

4.3.3. A Formal Approach.

The transformation of Bratman's idea into a formal system and the discourses of the semantics involve too many axioms that cannot be presented in short, so we only want to point out some basic aspects. The first one is that we must have a notion of persistent goal which is meant to specify a (long-term) intention. In order to describe a persistent goal, the use of modal logic must be provided to our system. Furthermore, four operators are defined:

- HAPPENS specifying the next action to perform[5],
- DONE specifying the last action to be executed[6],

---

[5] implying commitment for the short-term intention

[6] This is necessary to check the outcomes of a previous action, especially actions of other agents.

- GOAL specifying a long-term intention,
- BELief states whether a goal or fact is believed to be achieved or not.

Associated with the operators HAPPENS and DONE are formulas about actions that can be as well primitive as composite, where composite expressions are constructed by sequencing, non-determinsitic choice, test (e.g., to check the effects of a previous action) and iteration.

The organizational aspects, especially the notions social intention and social commitment are omitted in this paper.

Note: In contrast to this approach, Werner proposes a formalization of intention as a set of strategies to fulfil a given goal [Werner 89].

4.4. A General Approach to Multiagent Planning

Recently, Rao, Georgeff, and Sonenberg [Rao& 91] take some ideas presented in the chapters 4.1. - 4.3. and put them together to a new approach for multiagent planning. The first idea is to combine simpler plans to complex plans as demonstrated in [Georgeff 89b]. In analogy, the authors combine groups of agents to more sophisticated agents with high developed capabilities, so called *social agents*. And the second idea is to have agents guided by their intentions.

Rao, Georgeff, and Sonenberg distinguish between goals and intentions as persistent goals. The idea is to drop intentions the moment they can no longer be satisfied or the agents succeeds in accomplishing them. The approach includes modal and temporal logics that are missing in the first issue by Georgeff [Georgeff 89a].

Plans are defined as

(1) primitive plans,
(2) complex plans

where the latter are closed under

(a) sequencing (p1;p2),
(b) parallelism (p1||p2),
(c) non-deterministic choice (p1|p2),
(d) plans to achieve a value α (!α)[7],

---

[7] This was the syntax of receiving in Georgeff´s article (4.1.).

(e) plans to ask a value $\propto$ to be true ($?\propto$)[8].

In a next step, the plans constructed as described above must be associated with agents. In analogy to the plans, agents are distinguished in single and *social agents* (a group of some or all agents). A plan associated with some kind of agent (single or social) is called a *social plan expression* (agents commit to a plan). Such a social plan expression can be decomposed from an abstract level into more basic levels by distributing the less abstract subplans among the agents of the society where the decomposition can continue in the same manner: the agents of this society can be social (composite) agents and the subplans can be complex plans, permitting distribution of subplans among agents, and so on...

Afterwards, in order to allow intention-directed behavior a temporal tree logic and some modal operators are introduced. Thus, two kinds of operators are defined:

(1) path operators and

(2) state operators.

The state operators are valid for (state) formulas at a point in time and the path operators belong to (path) formulas on branches in the time tree. The path operators are interesting for warranting persistent goals.

Modal operators are:

(1) paths:

- optional -> there exists a path…
- inevitable -> on all paths…

(2) states:

- BEL -> beliefs that a formula is valid
- GOAL -> specifying a long-term intention
- INTEND -> specifying a short-term intention

Those three state operators are defined for single agents, and for some or all members of a society separately. So, for instance, the mutual belief of a group can be distinguished from the belief of a single agent.

5. Negotiation

This discussion about negotiation is particularly influenced by [Lâasri& 90], because it is first of all this article that presents a general introduction to the subject. Closer to an application, some approaches have been presented in the context of simple communication; in particular [Kreifelts& 90] gives details concerning the

---

[8] This was the syntax of sending in Georgeff´s article. Note that this is not deciding $\propto$; maybe, the execution of this plan will never succeed.

implementational requirements for negotiation between agents with a centralized negotiation manager (chapter 3.2.3.).

Multiagent systems are concerned with two kinds of conflicts (problems):

- the domain problem
- the control problem

The domain problem arises from inconsistent solutions to some local tasks, and the control problem is the problem of achieving good coordination of agents' actions (e.g. no deadlocks). Both problems can be solved by negotiation.

In multiagent systems, negotiation can be defined as a process of reaching agreement among agents; it can be expressed in terms of proposals and critiques.

Proposals can be[9]

- specific solutions
- partial results
- cluster solutions
- partial cluster solutions;

they can be created

- independently
- in reaction to another proposal
- in reaction to the critique of a previous proposal.

Critiques are represented as composite objects including…

- a positive part (agreement)
- a negative part (disagreement)
- an explanation (the reasons for agreeing or disagreeing)
- a counterproposal

Another problem in negotiation is to decide for when it should take place. In principal, negotiation may occur in each phase of joint problem solving; this means: during goal formulation (e.g. decomposition), goal determination (the phase to decide the next action), goal allocation (distribution of subgoals among agents), and goal achievement (e.g., recombination). The proper conflict detection that makes negotiation necessary may appear…

---

[9] Unforunately, the authors do not explain what partial or cluster solutions look like for actions in the control problem.

- in the local environment
- because of activities of the other agents
- during the negotiation process.

After the detection of a conflict, proposals have to be signaled to other agents that need to be specified implicitly or explicitly in advance.

6. Other Aspects

Aside the cooperation aspects mentioned above, there are some more that have not been integrated, yet.

First of all, we have to recognize that we started our discussion about multiagent systems with the notice that the most obvious difference is that some systems are hierarchical and others are shallow, and we broke down the hierarchical systems into their layers and considered each layer as an entire shallow multiagent system. All aspects of cooperation in multiagent systems we discussed in this report, assumed that cooperation takes place horizontally. But, in fact, concerning hierarchical systems, this is only one of two aspects, because the cooperation of the different layers has to be examined, too.

A discourse about vertical cooperation could be split into three different points of view: intention, resources, and behavior [Burmeister& 91].

Concerning the intentional aspect, this is still a completely open question.

Concerning the resources of the agents, we have to be aware that agents on different layers are heterogeneous, and therefore, the interchange of information may require some kind of artificial language and special translation mechanisms. Those translation mechanisms could particularly be examined with respect to the three measurements presented in [Fox 89]: cost, time [Stein 91], and space.

Concerning the behavior, this has to do with the interrelation between the agents' architecture and their cooperation. Especially for reactive systems without communication, this is a crucial question: it deals with how agents' individual behavior need to be designed to emerge as the desired behavior of a higher level agent [Wavish 91].

Finally, there is still a very important aspect that is not treated at all in the articles we have found: this is the assessment of competence. This is important, because it is not clear for a social goal of a social agent, who is the best single agent to commit which (sub-) goal. In general, it will not be sufficient to decide this problem by means of a payoff matrix. Here is still a lot of work to do.

7. Literature

[Boissier& 91] Olivier Boissier, Yves Demazeau: A Distributed Artificial Intelligence View on General Purpose Vision Systems, in: MAAMAW 91

[Bratman 87] M. Bratman: Intention, Plans, and Practical Reason, in: Harvard University Press, 1987

[Brooks 91] Rodney A. Brooks: Intelligence without Representation, in: Artificial Intelligence 47, 1991

[Bürckert& 90] Hans-Jürgen Bürckert, Jürgen Müller: RATMAN: Rational Agents Testbed for Multi-Agent Networks, in: MAAMAW 90

[Burmeister& 91] Birgit Burmeister, Kurt Sundermeyer: Cooperative Problem-Solving Guided by Intentions and Perception, in: MAAMAW 91

[Cammarata& 89] Stephanie Cammarata, David McArthur, Randall Steeb: Strategies of Cooperation in Distributed Problem Solving, in: [Gasser& 89]

[Cohen& 90] Philip R. Cohen, Hector J. Levesque: Intention is Choice with Commitment, in: Artificial Intelligence 42, 1990

[Decker& 89] Keith S. Decker, Edmund H. Durfee, Victor R. Lesser: Evaluation Research in Cooperative Distributed Problem Solving, in: DAI II, 1989

[Demazeau& 90] Y. Demazeau, J.-P. Muller: Decentralized Artificial Intelligence, Vol. I, Elsevier Sc. Pub/North Holland, 1990

[Demazeau& 91] Y. Demazeau, J.-P. Muller: Decentralized Artificial Intelligence, Vol. II, Elsevier Sc. Pub/North Holland, 1991

[Durfee& 89] Edmund H. Durfee, Victor R. Lesser: Using Partial Global Plans to Coordinate Distributed Problem Solvers, in: [Gasser& 89]

[Fox 89] Mark S. Fox: An Organizational View of Distributed Systems, in: [Gasser& 89]

[Gasser& 89] Les Gasser, Michael N. Huhns: Readings in Distributed Artificial Intelligence, Vol.II, Pitman & Morgan Kaufmann, 1989

[Genesereth& 89] Michael R. Genesereth, Matthew L. Ginsberg, Jeffrey S. Rosenschein: Cooperation without Communication, in: [Gasser& 89]

[Georgeff 89a] Michael Georgeff: Communication and Interaction in Multi-Agent Planning, in: [Gasser& 89]

[Georgeff 89b] Michael Georgeff: A Theory of Action for Multi-Agent Planning, in: [Gasser& 89]

[Huhns& 90] Michael N. Huhns, David Murray Bridgeland, Nitraj Vider Arni: A DAI Communication Aide, in: Proceedings of the 10th Workshop on DAI, Texas, 1990

[Hultman& 89] Johan Hultman, Anders Nyberg, Mikael Svensson: A Software Architecture for Autonomous Systems, in: First PROMETHEUS Scientific Workshop, Wolfsburg, 1989

[Kreifelts& 90] Thomas Kreifelts, Frank von Martial: A Negotiation Framework for Autonomous Agents, in: MAAMAW 90

[Lâasri& 90] Brigitte and Hassan Lâasri, Victor R. Lesser: Negotiation and its Role in Cooperative Distributed Problem Solving, in: Proceedings of the 10th Workshop on DAI, Texas, 1990

[Maes 89] Pattie Maes: The Dynamics of Action Selection, in: IJCAI 89

[Minski 86] Marvin Minsky: The Society of Mind, in: Simon and Shuster, New York, 1986

[Moyson& 88] Frans Moyson, Bernard Manderick: The Collective Behavior of Ants: An Example of Self-Organization in Massive Parallelism, in: AI-Memo 88-7, Brussels, 1988

[Rao& 91] Anand S. Rao, Michael P. Georgeff, Elizabeth A. Sonenberg: Social Plans: A Preliminary Report, in: MAAMAW 91

[Searle 90] John R. Searle: Collective Intentionality, in: Cohen, Morgan, Pollack: Intentions in Communication, MIT Press, 1990

[Seel 91] Nigel Seel: The 'Logical Omniscience' of Reactive Systems, in: AISB 91

[Singh 90] Munindar P. Singh: Group Ability and Structure, in: MAAMAW 90

[Smith 89] Reid G. Smith: The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver, in: [Gasser& 89]

[Stein 91] Johannes Stein: Entwicklung eines Zeitmodells mit der Fähigkeit zur Variation der Granularität und dessen Implementierung im Repräsentationswerkzeug TROPES, in: Diplomarbeit, Universität Kaiserslautern, 1991

[Sundermeyer 90] Kurt Sundermeyer: Modellierung von Szenarien kooperierender Akteure, in: GWAI 90

[Wavish 91] Peter Wavish: Exploiting Emergent Behavior in Multi-Agent Systems, in: MAAMAW 91

[Werner 89] Eric Werner: Cooperating Agents: A Unified Theory of Communication and Social Structure, in: DAI II, 1989

[Wittig 90] Thies Wittig: ARCHON - Cooperation of Heterogeneous On-line Systems, ESPRIT-Project 2256, 1990

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
67608 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder per anonymem ftp von ftp.dfki.uni-kl.de (131.246.241.100) unter pub/Publications bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or via anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) under pub/Publications.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-93-14
Joachim Niehren, Andreas Podelski, Ralf Treinen: Equational and Membership Constraints for Infinite Trees
33 pages

RR-93-15
Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster: PLUS - Plan-based User Support Final Project Report
33 pages

RR-93-16
Gert Smolka, Martin Henz, Jörg Würtz: Object-Oriented Concurrent Constraint Programming in Oz
17 pages

RR-93-17
Rolf Backofen:
Regular Path Expressions in Feature Logic
37 pages

RR-93-18
Klaus Schild: Terminological Cycles and the Propositional μ-Calculus
32 pages

RR-93-20
Franz Baader, Bernhard Hollunder:
Embedding Defaults into Terminological Knowledge Representation Formalisms
34 pages

RR-93-22
Manfred Meyer, Jörg Müller:
Weak Looking-Ahead and its Application in Computer-Aided Process Planning
17 pages

RR-93-23
Andreas Dengel, Ottmar Lutzy:
Comparative Study of Connectionist Simulators
20 pages

RR-93-24
Rainer Hoch, Andreas Dengel:
Document Highlighting —
Message Classification in Printed Business Letters
17 pages

RR-93-25
Klaus Fischer, Norbert Kuhn: A DAI Approach to Modeling the Transportation Domain
93 pages

RR-93-26
Jörg P. Müller, Markus Pischel: The Agent Architecture InteRRaP: Concept and Application
99 pages

RR-93-27
Hans-Ulrich Krieger:
Derivation Without Lexical Rules
33 pages

RR-93-28
Hans-Ulrich Krieger, John Nerbonne, Hannes Pirker: Feature-Based Allomorphy
8 pages

RR-93-29
Armin Laux: Representing Belief in Multi-Agent Worlds viaTerminological Logics
35 pages

RR-93-30
Stephen P. Spackman, Elizabeth A. Hinkelman:
Corporate Agents
14 pages

RR-93-31
Elizabeth A. Hinkelman, Stephen P. Spackman:
Abductive Speech Act Recognition, Corporate Agents and the COSMA System
34 pages

RR-93-32
David R. Traum, Elizabeth A. Hinkelman:
Conversation Acts in Task-Oriented Spoken Dialogue
28 pages

RR-93-33
Bernhard Nebel, Jana Koehler:
Plan Reuse versus Plan Generation: A Theoretical and Empirical Analysis
33 pages

RR-93-34
Wolfgang Wahlster:
Verbmobil Translation of Face-To-Face Dialogs
10 pages

RR-93-35
Harold Boley, François Bry, Ulrich Geske (Eds.): Neuere Entwicklungen der deklarativen KI-Programmierung — *Proceedings*
150 Seiten
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

RR-93-36
Michael M. Richter, Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Gabriele Schmidt: Von IDA bis IMCOD: Expertensysteme im CIM-Umfeld
13 Seiten

RR-93-38
Stephan Baumann: Document Recognition of Printed Scores and Transformation into MIDI
24 pages

RR-93-40
Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, Andrea Schaerf:
Queries, Rules and Definitions as Epistemic Statements in Concept Languages
23 pages

RR-93-41
Winfried H. Graf: LAYLAB: A Constraint-Based Layout Manager for Multimedia Presentations
9 pages

RR-93-42
Hubert Comon, Ralf Treinen:
The First-Order Theory of Lexicographic Path Orderings is Undecidable
9 pages

RR-93-43
M. Bauer, G. Paul: Logic-based Plan Recognition for Intelligent Help Systems
15 pages

RR-93-44
Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, Martin Staudt: Subsumption between Queries to Object-Oriented Databases
36 pages

RR-93-45
Rainer Hoch: On Virtual Partitioning of Large Dictionaries for Contextual Post-Processing to Improve Character Recognition
21 pages

RR-93-46
Philipp Hanschke: A Declarative Integration of Terminological, Constraint-based, Data-driven, and Goal-directed Reasoning
81 pages

RR-93-48
Franz Baader, Martin Buchheit, Bernhard Hollunder: Cardinality Restrictions on Concepts
20 pages

RR-94-01
Elisabeth André, Thomas Rist:
Multimedia Presentations:
The Support of Passive and Active Viewing
15 pages

RR-94-02
Elisabeth André, Thomas Rist:
Von Textgeneratoren zu Intellimedia-Präsentationssystemen
22 Seiten

RR-94-03
Gert Smolka:
A Calculus for Higher-Order Concurrent Constraint Programming with Deep Guards
34 pages

RR-94-05
Franz Schmalhofer,
J.Stuart Aitken, Lyle E. Bourne jr.:
Beyond the Knowledge Level: Descriptions of Rational Behavior for Sharing and Reuse
81 pages

RR-94-06
Dietmar Dengler:
An Adaptive Deductive Planning System
17 pages

RR-94-07
Harold Boley: Finite Domains and Exclusions as First-Class Citizens
25 pages

RR-94-08
Otto Kühn, Björn Höfling: Conserving Corporate Knowledge for Crankshaft Design
17 pages

RR-94-10
Knut Hinkelmann, Helge Hintze:
Computing Cost Estimates for Proof Strategies
22 pages

RR-94-11
Knut Hinkelmann: A Consequence Finding Approach for Feature Recognition in CAPP
18 pages

RR-94-12
Hubert Comon, Ralf Treinen:
Ordering Constraints on Trees
34 pages

RR-94-13
Jana Koehler: Planning from Second Principles —A Logic-based Approach
49 pages

RR-94-14
Harold Boley, Ulrich Buhrmann, Christof Kremer:
Towards a Sharable Knowledge Base on Recyclable Plastics
14 pages

RR-94-15
Winfried H. Graf, Stefan Neurohr: Using Graphical Style and Visibility Constraints for a Meaningful Layout in Visual Programming Interfaces
20 pages

RR-94-16
Gert Smolka: A Foundation for Higher-order Concurrent Constraint Programming
26 pages

RR-94-17
Georg Struth:
Philosophical Logics—A Survey and a Bibliography
58 pages

RR-94-18
Rolf Backofen, Ralf Treinen:
How to Win a Game with Features
18 pages

RR-94-20
Christian Schulte, Gert Smolka, Jörg Würtz:
Encapsulated Search and Constraint Programming in Oz
21 pages

RR-94-31
Otto Kühn, Volker Becker, Georg Lohse, Philipp Neumann:
Integrated Knowledge Utilization and Evolution for the Conservation of Corporate Know-How
17 pages

RR-94-33
Franz Baader, Armin Laux:
Terminological Logics with Modal Operators
29 pages

DFKI Technical Memos

TM-92-04
Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer:
On the Representation of Temporal Knowledge
61 pages

TM-92-05
Franz Schmalhofer, Christoph Globig, Jörg Thoben:
The refitting of plans by a human expert
10 pages

TM-92-06
Otto Kühn, Franz Schmalhofer: Hierarchical skeletal plan refinement: Task- and inference structures
14 pages

TM-92-08
Anne Kilger: Realization of Tree Adjoining Grammars with Unification
27 pages

TM-93-01
Otto Kühn, Andreas Birk: Reconstructive Integrated Explanation of Lathe Production Plans
20 pages

TM-93-02
Pierre Sablayrolles, Achim Schupeta:
Conlfict Resolving Negotiation for COoperative Schedule Management
21 pages

TM-93-03
Harold Boley, Ulrich Buhrmann, Christof Kremer:
Konzeption einer deklarativen Wissensbasis über recyclingrelevante Materialien
11 pages

TM-93-04
Hans-Günther Hein:
Propagation Techniques in WAM-based Architectures — The FIDO-III Approach
105 pages

TM-93-05
Michael Sintek: Indexing PROLOG Procedures into DAGs by Heuristic Classification
64 pages

TM-94-01
Rainer Bleisinger, Klaus-Peter Gores:
Text Skimming as a Part in Paper Document Understanding
14 pages

TM-94-02
Rainer Bleisinger, Berthold Kröll:
Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations
11 pages

DFKI Documents

D-93-14
Manfred Meyer (Ed.): Constraint Processing – Proceedings of the International Workshop at CSAM'93, July 20-21, 1993
264 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-15
Robert Laux:
Untersuchung maschineller Lernverfahren und heuristischer Methoden im Hinblick auf deren Kombination zur Unterstützung eines Chart-Parsers
86 Seiten

D-93-16
Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Gabriele Schmidt: Design & KI
74 Seiten

D-93-20
Bernhard Herbig: Eine homogene Implementierungsebene für einen hybriden Wissensrepräsentationsformalismus
97 Seiten

D-93-21
Dennis Drollinger:
Intelligentes Backtracking in Inferenzsystemen am Beispiel Terminologischer Logiken
53 Seiten

D-93-22
Andreas Abecker:
Implementierung graphischer Benutzungsoberflächen mit Tcl/Tk und Common Lisp
44 Seiten

D-93-24
Brigitte Krenn, Martin Volk:
DiTo-Datenbank: Datendokumentation zu Funktionsverbgefügen und Relativsätzen
66 Seiten

D-93-25
Hans-Jürgen Bürckert, Werner Nutt (Eds.):
Modeling Epistemic Propositions
118 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-26
Frank Peters: Unterstützung des Experten bei der Formalisierung von Textwissen
INFOCOM:
Eine interaktive Formalisierungskomponente
58 Seiten

D-93-27
Rolf Backofen, Hans-Ulrich Krieger, Stephen P. Spackman, Hans Uszkoreit (Eds.):
Report of theEAGLES Workshop on Implemented Formalisms at DFKI, Saarbrücken
110 pages

D-94-01
Josua Boon (Ed.):
DFKI-Publications: The First Four Years
1990 - 1993
93 pages

D-94-02
Markus Steffens: Wissenserhebung und Analyse zum Entwicklungsprozeß eines Druckbehälters aus Faserverbundstoff
90 pages

D-94-03
Franz Schmalhofer: Maschinelles Lernen:
Eine kognitionswissenschaftliche Betrachtung
54 pages

D-94-04
Franz Schmalhofer, Ludger van Elst:
Entwicklung von Expertensystemen:
Prototypen, Tiefenmodellierung und kooperative Wissensevolution
22 pages

D-94-06
Ulrich Buhrmann: Erstellung einer deklarativen Wissensbasis über recyclingrelevante Materialien
117 pages

D-94-07
Claudia Wenzel, Rainer Hoch:
Eine Übersicht über Information Retrieval (IR) und NLP-Verfahren zur Klassifikation von Texten
25 Seiten

D-94-08
Harald Feibel: IGLOO 1.0 - Eine grafikunterstützte Beweisentwicklungsumgebung
58 Seiten

D-94-09
DFKI Wissenschaftlich-Technischer Jahresbericht 1993
145 Seiten

D-94-10
F. Baader, M. Lenzerini, W. Nutt, P. F. Patel-Schneider (Eds.): Working Notes of the 1994 International Workshop on Description Logics
118 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-94-11
F. Baader, M. Buchheit, M. A. Jeusfeld, W. Nutt (Eds.):
Working Notes of the KI'94 Workshop:
KRDB'94 - Reasoning about Structured Objects: Knowledge Representation Meets Databases
65 Seiten

D-94-12
Arthur Sehn, Serge Autexier (Hrsg.):
Proceedings des Studentenprogramms der 18. Deutschen Jahrestagung für Künstliche Intelligenz KI-94
69 Seiten