Task Decomposition in Dynamic Agent Societies*

N. Kuhn† H.J. Müller J.P. Müller
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, D-6600 Saarbrücken 11

Abstract

The Contract Net Protocol is one of the most prominent concepts for negotiation in Multi-Agent Systems. However, there are some unspecified features like the task decomposition procedures used by managers. Also the bottleneck induced by the central role of the managers will be identified as a weak point of the method. This is especially true if the amount of tasks in the system and the society of agents itself are dynamic. We will first investigate in a general task decomposition strategy. Then the central role of the manager will be softened by introducing the Decentralized Task Decomposition Model. Finally, in the Completely Decentralized Model, the manager will be totally eliminated. Our ideas are presented in a transportation domain, which is also described to some extent.

Keywords: DAI, Multi-Agent Systems, Negotiation, Task Decomposition

1 Introduction

The tasks computers have to deal with get more and more complex. Many of the problems under computation can be identified as being problems of an inherently distributed nature. Factory scheduling, aircraft control, route planning, fleet coordination, etc. might serve as examples. Moreover, these problems ask for more than mere computation. Knowledge, i.e. explicitly represented domain knowledge, should guide the search for a good solution [18]. The field of Distributed Artificial Intelligence tackles these problems by exploiting the naturally given distribution ([2] [15], [13]). Especially in the subfield of Multi-Agent Systems ([6], [7]) the research concentrates on agent models that enable the agents to work together effectively. The most important aspects in this context are the communication capabilities of the agents and their degree of autonomous coordination. In the combination of both, i.e. in using communication for coordination, lies the real power of heterogeneous agents in a multi-agent environment. Negotiation is the key word for this kind of cooperative behaviour, and it is THE topic throughout the work in DAI over the last decade (e.g. [22], [24], [25], [26]).

In this paper we concentrate on the problems of task decomposition, task allocation and task synthesis. We will start with a well known and often applied protocol for negotiation processes, the *Contract Net Protocol* (CNP) ([5]). It was originally designed to strengthen the efficiency of the task decomposition and task allocation process. To be more precise, the allocation of subtasks to "adequate" agents is performed by the means of single-agent decisions. Agents decide on their own whether they like to do a certain job or not. If they "want" to do it they have to give a bid to the offerer (manager). However, the manager has to do the decomposition of a task into a set of subtasks. He has to sample the bids and he has to decide to which bidder the contract will be given. So he is central to the whole process, whereas the other agents have a subordinate role. This is sufficient in domains where fixed roles are given to agents a priori (as in the distributed sensor application which has been used to introduce the CNP), but in dynamic societies where the environment and the tasks for the agents change dynamically, mechanisms are needed that go beyond the CNP idea.

After a brief description of our general application domain, which deals with the coordination of transportation companies, and a short revision of the CNP, we present a general model for the task decomposition process. Then we soften the central position of the manager by giving more decision power to the bidders. The idea is that the bidders should have the possibility to choose (not predefined) subtasks on their own. The role of the manager is to coordinate the wishes of the bidders. The new model will be called *The Decentralized Task Decomposition Model.* The next step is to eliminate the manager. In this model, called *The Completely Decentralized Model*, the agents will get the complete task directly. They choose their subtasks corresponding to their preferences and then negotiate on the global plan with their partners.

2 Cooperation and Negotiation in the MARS-Scenario

In the MARS-Scenario (Modeling a Multi-Agent Scenario for Shipping Companies) [3] a society of shipping companies is modeled whose task it is to transport goods between different places.

The transportation orders that the companies have to deliver are given to the system by one or a couple of users (which may themselves be shipping companies). A user can associate with his order a set of constraints that will have to be satisfied, e.g. he can address a particular shipping company, he can specify the means of transport (e.g. train or truck) or he may formulate some temporal constraints. The complex-

*This work has been supported by the German Federal Ministry of Reseach and Technology under grant ITW 9104.

†e-mail: kuhn@dfki.uni-sb.de

ity of the orders given may exceed the capacities of every single company and thus, require for the cooperation of some of them. The orders are collected in an *order bulletin board*, which serves as a global control unit that passes the orders to the shipping companies agents according to the constraints the user has specified. In addition, a distribution strategy[1] can be chosen among a set of predefined heuristics. The *agent society* of the MARS scenario consists of shipping company agents and truck agents:

A *shipping company* agent is responsible for the disposition of the orders he has gained. Therefore, he has to distribute the goods belonging to the orders among his trucks according to the constraints given by the user and to his private optimality criteria.

The *truck* agents stand for the variety of means of transport that are part of real world transportation domains. Each truck agent is associated with a particular shipping company from which he gets orders of the form *"Load goods g and transport them to location l"* Given such an order, the truck agent does the planning of the route (cf. [17]), transports the goods, and informs the shipping company agent about the delivery, thus telling him about his free capacities. Furthermore, he reports the shipping company agent about remaining capacities and planned routes, and he is able to estimate the effort (and effects)[2] combined with an order.

This reveals one of the many aspects of cooperation that may be found in our scenario: the relationship between the trucks and their shipping company is a strictly hierarchical one that can be modeled by a contract net mechanism as described in section 4.1: only on request the trucks give a bid to the shipping company which evaluates these bids when looking for a truck to deliver an order.

Another essential part of this scenario is comprised by the cooperation between different shipping companies. To model this we incorporate several more complex mechanisms into our system, e.g. the mechanism of negotiation (see chapter 4.1) or a more decentralized mode of task decomposition (see chapter 4.2). We focus on two types of cooperation that are motivated by a detailed observation of real-world shipping companies:

- avoidance of rides without carriage.
- coupling of inter- and intraregional traffic.

It might be a bit surprising that cooperation between different shipping companies plays an important role in the highly competitive transportation domain. But this strong competition is one of the reasons for the shipping companies being cooperative at all: the competitive situation implies a decrease of sales per order. Thus, in order to maximize profits the companies try to decrease their cost per order. If, for an order which includes a ride from A to B, there is another transportation order from B to A, the cost for these two orders will be nearly the same as for each single one. The different companies now try to enforce this effect of cost saving by mutually exchanging transportation orders. If there is a long-time balance of the order exchange between two companies then, their number of orders keeps constant while the cost decreases. Similar considerations prove the efficiency of the second cooperation type mentioned above.

Figure 1 shows how these two types of cooperation could be combined to obtain a task decomposition for a situation with a set of orders $\{o_1, o_2, o_3\}$ for the shipping companies $\{s_1, s_2, s_3\}$. To achieve the solution that is in the right part of the figure we use a negotiation mechanism that could proceed as follows:

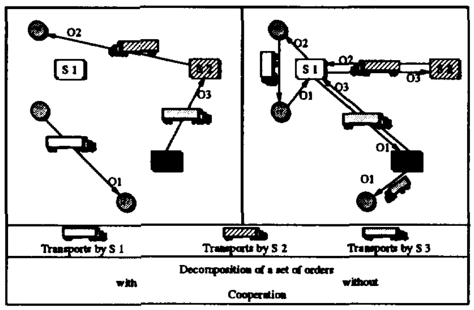

Figure 1: Cooperation Between Shipping Companies in the MARS Scenario

- s_1 asks s_3 to take over the local distribution of o_1 and he offers a free truck to s_3
- s_3 offers a free truck to s_2
- s_2 asks s_1 to take over the local distribution of o_2 and he offers a free truck to s_1
- s_1 agrees on doing the local distribution for o_2, if s_2 takes over order o_1
- s_2 disagrees on that because he does not want to go to the location of s_3
- s_1 updates his offer to s_3 concerning o_1 and asks s_3 to take over the long-distance part of o_1
- s_3 rejects because he has to deliver order o_3
- s_1 asks s_3 if it would be useful for him to have available the truck of s_2
- s_3 accepts the truck offered and replans the route for the order o_3

This protocol shows how the above solution can be constructed. The example will motivate the considerations that we are going to present in the following.

3 Task Handling in Multi-Agent Systems

In this chapter we briefly describe the characteristics of the multi-agent systems we want to consider.

[1] e.g., offer the order to all companies that are near its starting- or the target location i.e., the bulletin board acts like a yellow page server.

[2] i.e. cost, time, security of transport, ...

3.1 A Model for Task Decomposition

We define a multi-agent system (MAS) $\mathcal{M}$ to be a pair $\mathcal{M} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{A} = \{a_1, ..., a_n\}$ denotes the set of agents that comprise the system $\mathcal{M}$, and $\mathcal{T} = \{t_1, ..., t_l\}$ describes the set of *tasks* that the society of agents is able to perform. These tasks are accomplished by that the agents perform a set of *actions* they are capable of. To have an interface between the abstract language of the tasks and the language of the agent society we introduce the concept of the *goals*. The set of goals that can exist in $\mathcal{M}$ is denoted by the set $\mathcal{G} = \{g_1, ..., g_k\}$.

In general, the process of the *task decomposition* is as follows: Given a task $t \in \mathcal{T}$ as input, t has to be compiled into a set of goals $G_t = \{g_{t_i}, ..., g_{t_m}\} \subseteq \mathcal{G}$ that the agents have to accomplish. These goals have to be attached to particular agents. Usually, there may be several alternatives to compile a task into a set of goals according to different possible solutions to a problem (or task) or to the set of agents that are actually part of the system. Furthermore, even for the attachment of the goals to specific agents there may be different choices, e.g the task of carrying a table from a location A to a location B might be done either by one strong agent or by two weaker ones. In the latter case, in order to express the conjunction of the two agents for accomplishing the goal, the attachment process must break down further the goal under consideration, and it has to add constraints that must be satisfied when the goal is accomplished. This breaking down of a goal is an iterative process yielding for a set of goals $G_1 \subseteq \mathcal{G}$ a set $G_1' \subseteq \mathcal{G}$ which consists of atomic goals only, i.e. goals a set of agents is attached to[3]. In the following, we will refer to this process as *goal decomposition*.

3.2 Goal Decomposition and Planning

The multi-agent systems that we are interested in could be characterized by the term *dynamic multi-agent systems*. With this notion we want to stress the fact that we have no longer a system that is given a set of tasks or a set of goals at some starting time t and which will finish after having fulfilled all the initial tasks. But, the agents in our systems will have to deal with a more or less continuous stream of incoming tasks or goals. A first consequence of this assumption of dynamics is that the goal decomposition phase and the planning phase on the one hand, and the action execution phase on the other hand will have to be closely interleaved. This means that actions for the accomplishment of one goal will be executed while another goal is just being decomposed. This imposes a further consequence, namely that a new incoming goal can force the system to modify plans (or decompositions) that have been worked out before, because the former solution suddenly looks less reasonable now. This may even include a rollback of actions that have been already executed[4]. In other words, the input of new goals may imply the necessity of replanning sequences of actions for some of the agents.

On the agent level the attachment of a new goal to some agents can involve that these agents are no longer able to accomplish each goal they have been committed to before. Rather, some of the goals have to be retracted by the agents, and are thus open for decomposition again.

Therefore, a process for the decomposition of the goals in a MAS $\mathcal{M}$ should keep track of at least the following parameters:

1) The "state" of $\mathcal{M}$ from the viewpoint of the agents, yielding e.g. information about the current goal set in the system (i.e., which goals are open for decomposition; which decomposition has been chosen for the other goals), and knowledge about preferable decompositions.
2) agents that are in general suitable for the accomplishment of a particular goal
3) agents that are actually available for the accomplishment of a particular goal

There are several alternatives how to implement such a decomposition process in a real multi-agent system: One major criterion for the characterization of an implementation of this process in a MAS is whether it is implemented in a single centralized process or if this process is distributed all over the agent society. As regards the planning process, we could make a further distinction on whether this is done by a central planner or by each agent himself. But, as we are mainly interested in *autonomous* distributed systems, we will neglect this point and we will assume in the following that we are dealing with systems where each agent has its own planning capabilities.

This assumption has some direct consequence for the goal decomposition process. In general, the determination of a group of agents, which is suitable and which is available for the attachment of a particular goal does not only depend on the goals that have been attached to them before but furthermore, on the plan that has been worked out by the agents for these previous goals. This implies that any goal decomposition process will have need for information from 'within' the agents, e.g. whether or not an agent is satisfied at the moment with the goals he is committed to. After having collected this load information the decomposition process may make up its decision for decomposing an actual set of goals.

A very general idea to deal with this problem of distributed information has been provided by the *Contract Net Protocol*. This protocol and other approaches to the task decomposition problem will be discussed in the next section.

4 Models of Task Decomposition in Dynamic Agent Societies

[3]According to the attachment which is actually chosen we may sometimes regard a goal as an atomic one, in another situation this might be not the case. See the table-carrying problem as an example.

[4]This might be not the case for actions that consume some limited resources, like fuel, etc.!

4.1 The Contract Net Protocol

The Contract Net Protocol (CNP) was introduced by Smith and Davis in a series of publications ([5, 23]). The general idea is the following: A certain task is given to a society of agents. One agent, called the manager, receives the task and divides it into a set of subtasks. He announces them (in a sequence of announcements) to a set of eligible agents (chosen on the basis of his knowledge about the others). These agents process the task announcement, i.e., they rank the task relative to others currently under consideration. When being idle at some time, they prepare bids for stored tasks and send the bids to the respective managers. The manager ranks the incoming bids and after an expiring time he chooses the best one.

Though we think that the CNP is a very elegant way to view negotiation between agents, and though we especially agree with the authors that it is most useful in hierarchically organized societies, there is one big bottleneck with the approach. For many interesting applications, there are quite a few good reasons to consider the central role of the manager as being too powerful:

1. To choose a subset of eligible agents, the manager needs to have a large amount of knowledge about the other agents in the society. This does not correspond to the philosophy of decentralization of data. Even more, since each agent is a potential manager, the knowledge must be available to each agent.
2. The manager has to have several strategies to decompose a task. Choosing the wrong decomposition means that several rounds of announcing and bidding are necessary until a complete subtask allocation is installed.

We will eliminate the first problem in giving the task decomposition procedures to the contractors (bidders). This reduces the amount of global knowledge of the agents and allocates the different task decomposition procedures to responsible and eligible agents. In other words, each agent will decompose a given task on his own behalf and pick out a maximum executable subtask of his own. The role of the manager now becomes that of a solution synthesizing specialist who actually organizes the cooperative work of the group. The advantage of the described Decentralized Task Decomposition Model (DTDM) is the local expertise of agents.

However, the DTDM has still the problem of reliability due to the central position of the manager. The next step will eliminate this drawback in delegating the mission of the manager to the society (!). Since it can not be distributed in the same way as the task decomposition procedures, there must be another mechanism which goes beyond the interpretation of negotiation provided by the CNP. The agents explicitly have to negotiate on the subtasks they like to work on. By collecting piece by piece the subsolutions, a "joint plan" must be built. If there are conflicts, for instance if more than one agent applies to the same subtask, or if subtasks overlay, the agents have to negotiate with the aim of a balanced load.

4.2 The Decentralized Task Decomposition Model

Motivated by the shortcomings of the Contract Net Model of task decomposition for dynamic agent societies described in section 4.1, we would like to approach one stage closer to the paradigm of a decentralized system by a model which we call the Decentralized Task Decomposition Model (DTDM). In this model, the original structure of the Contract Net is softened by shifting the task decomposition to the society of contractors: the manager receives a task and passes it as a whole to a set of eligible contractors. The contractors work out a bid for a part of the task, and pass it back to the manager. Now, the manager can synthesize a plan for the task from the bids for subtasks received by some of the contractors, while rejecting the bids of other contractors. In section 5.2, we will describe how negotiation between the manager and the contractors can help to find more appropriate task decompositions which lead to better solutions to the overall task.

Compared to the Contract Net Model, the DTDM yields a more flexible behaviour of the system, since

- The manager needs less knowledge about the different contractors. Rather, each contractor may choose a subtask which seems appropriate to him. However, by knowing the subtasks offered by the contractors, the manager can have an important coordinating function.
- Communication costs are reduced, because instead of announcing each subtask, the manager only announces the task as a whole.
- By employing negotiation between the manager and the contractors, task decompositions can be achieved which are both locally and globally acceptable.

However, for some domains, even the existence of a manager is not desired or just impossible to assume. For these domains, the DTDM might be regarded not satisfactory. Therefore, in subsection 4.3, we introduce a model which provides a degree of decentralization which is even higher than in the case of the DTDM.

4.3 The Completely Decentralized Model

In the Completely Decentralized Model (CDM), the society of agents has to decompose and to allocate the tasks and to synthesize a plan for carrying out the task without the help of a manager. This decentralized task decomposition and task synthesis can be viewed as a decentralized planning process. Agents may either propose whole plans or partial plans to other agents, or they may construct a joint plan e.g. by using a system of a circular letter which is sent from agent to agent, and which can be modified by each agent, until a complete plan is built which is accepted by all participants. The absence of a central instance causes many new problems to occur: agents may have different and even inconsistent intentions, different degrees of cooperativeness, very diverse amounts and types of knowledge and beliefs, and different skills and abilities. Finding coordinated plans requires such an agent society to communicate, to exchange goals, plans, arguments, and intentions, to cope with conflicts etc.

Negotiation[1, 9, 16] establishes a very powerful tool for handling this kind of problem. By negotiation, conflicts between agents can be bridged, an agent can convince another agent of the benefits of his proposal, or the frame conditions for a joint plan and the joint plan itself, i. e. the task decomposition and allocation, can be agreed on. In section 5 we will give an overview on different ways to use negotiation for task decomposition in Multi-Agent Systems. In the case of the CDM, we can say that to use some form of negotiation between agents is not a choice which is up to the agents (or to the designer of the agent society).

5 Task Decomposition by Negotiation

In chapter 4 we introduced several models of task decomposition which differed by their degree of decentralization. There we supposed that agents would send proposals for task decomposition to other agents, and that these might either accept or reject the proposal.

However, if we want to obtain a more realistic view on negation, aspects of negotiation should be integrated. By [4], negotiation in a multiagent context is defined as *the communication process of a group of agents in order to reach a mutually accepted agreement on some matter*. According to this definition, the task decomposition itself may be negotiated on. In this section we would like to outline how task decomposition can be negotiated in the models defined in section 4.

5.1 Negotiation in the Contract Net Model

The task model of the Contract Net, which has been described in section 4.1, is characterized by centralized task decomposition and centralized task synthesis. The manager splits a task into several subtasks and announces each subtask to one agent or a group of agents. In the original Contract Net Protocol, each agent may either make a bid for a subtask, or he may show no interest for doing that subtask. Thus, in order to find a suitable task decomposition, the manager needs profound knowledge of other agents' problem solving capabilities and even of their internal representations. Otherwise, there is a considerable risk that for a given subtask no contractor will be found.

A more flexible mechanism for task decomposition in the Contract Net model can be achieved by allowing negotiation on the frame conditions of a subtask between the manager and the potential contractors. By this, satisfactory task decompositions can be reached even when the manager has no complete knowledge (or even wrong beliefs) of the potential contractors. We would like to show this by an example from the MARS domain which is illustrated in figure 2.

Example 1 *Assume S_1, S_2 are shipping companies. S_1 owns one truck with a loading capacity of* 20 *units, S_2 owns one truck with a loading capacity of* 10 *units. A customer C has a task T =*"Transport 6 pallets each of five units from place A to place B!". *Assume that C has no knowledge about the loading capacities of S_1 and S_2, respectively. In this case, he may use a heuristics, namely to decompose the order in two equal parts T_1 =* "Transport 3 pallets from A to B!" *which he decides to send to S_1 and T_2 =* "Transport 3 pallets from A to B!" *which he sends to S_2.*

Using the normal CNP, S_2 would recognize that he is not able to carry out the task (at least not directly). Therefore, S_1 would be granted T_1 whereas T_2 could not be carried out, at all. Moreover, assume that S_1 does not know what a pallet *is. Since he does not know what it is, obviously he cannot carry out T_1!. In conclusion, although from a global point of view it is obvious that the task could be executed if an appropriate task decomposition and representation were chosen, the system is not able to carry out the task.*

If we allow negotiation between the customer and the potential contractors, the following will happen: S_2 might tell C: "I cannot transport 15 *units, but I can transport* 10 *units." At the same time, S_1 might tell C: "I do not understand the word "pallet", can you be more precise?" Now, on the one hand, C can use the knowledge obtained by the negotiation with S_2 in order to choose another task decomposition consisting of T_1' = "Transport 2 pallets from A to B" which he sends to S_2. Now, he can use the information obtained by S_1's response and change the representation of the subtask T_2' to T_2' = "Transport 20 units from A to B", a representation which can be understood by C.*

Thus, an appropriate task decomposition can be found.

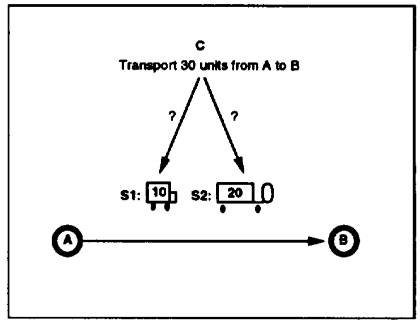

Figure 2: Example 1

5.2 Negotiation in the Decentralized Task Decomposition Model

In the decentralized task decomposition model of section 4.2, the manager is no longer responsible for task decomposition. Instead, he sends the task as a whole to the potential contractors, each of which may cut a slice of the task for himself, and announce to the manager his interest in that paricular part of the task. The manager now synthesizes a plan for the complete task from the proposals of the agents.

In some ways, decentralized task decomposition models suffer from their locality. The contractors have only a local view, and they will choose subtasks without taking into consideration the behaviour of other agents. Therefore, it often happens that either there are conflicts between several contractors (e.g., some contractor wants to do the task as a whole, which is certainly impossible), or that parts of the task are not chosen by any contractor. Negotiation can be considered as a solution to these problems: there can be a negotiation between the manager and potential contractors in order to modify announcements of subtasks made by a contractor. Here, the manager can take advantage of his more global view obtained by knowing the offers of several contractors. On the other hand, contractors can negotiate with each other. This is a step into the direction of a completely decentralized system, where no manager is required (cf. subsection 5.3) at all. However we can imagine a hybrid solution where a manager announces the tasks and receives and synthesizes offers for task decomposition, but where negotiation between contractors (e.g., in order to form a group solving a single subtask) is possible.

Again, we would like to show by an example how task decomposition proposals made by potential contractors can be modified by negotiation in order to reach a better solution of the overall task. The example is illustrated by figure 3.

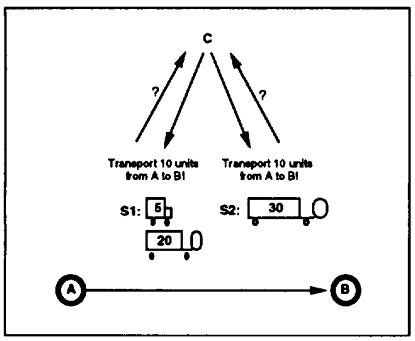

Figure 3: Example 2

Example 2 *Again there are a customer C, two companies S_1 and S_2. S_1 has two trucks with loading-capacities of* 5 *and* 20 *units, respectively, and S_2 owns one truck with a loading capacity of* 30 *units. Now let the task T be* "Transport 10 units from A to B". *Both S_1 and S_2 receive T and check which parts of T they are capable and willing to carry out. Now assume that both S_1 and S_2 use a heuristics which says not to apply for a task if the truck which is to perform it cannot be loaded by more than* 50% *of its loading capacity. In this case, S_1 would apply for transporting* 5 *units with his small truck, and S_2 would not apply for T, at all.*

If we allow negotiation, S_1 could propose to C to transport 10 *units, i. e. to carry out T completely, if C will pay more for it, and they could agree on a higher price for performing T.*

In conclusion, the use of negotiation in decentralized task decomposition models allows higher flexibility and a better performance of the system as a whole.

5.3 Negotiation in the Completely Decentralized Model

As described in section 4.3, by decentralizing the synthesis of tasks we obtain a completely decentralized task model. Here, the manager has become superfluous. Rather, the agent society decomposes the task in a set of subtasks and combines the solution to the subtasks to a plan for the task as a whole.

In completely decentralized models negotiation is not only reasonable, but it is very necessary, since it allows agents to cope with tasks without having complete knowledge about the abilities of others. Therefore, agents maintain models of other agents which contain their beliefs about the capabilities, intentions, and plans of these agents. The partner models are updated by messages received from other agents, and by perceiving the behaviour of these agents. Lacking information needed for making decisions can be acquired from other agents by asking them questions.

As we said before, negotiation between agents is performed via the sending of messages. Agents may create plans for the task or for subtasks and send them to other agents who can accept, reject, refine, or modify these plans (cf. [8, 16] as examples), thus finally agreeing on a joint plan.

In the MARS-scenario one objective of the use of negotiation is the avoidance of rides without carriage. The negotiation protocol for this cooperation type is initiated by a truck who recognizes an unbooked leg in the route he has planned for the delivery of his orders. He announces it to his company agent, who decides what he wants to do with this free capacity. One possibility is to offer it to eligible other companies (e.g., partner companies) who may then apply for it[5]. After an expiration time (e.g., when the truck has to start to deliver the next orders in time) either the company agent chooses the best order among the applications and allocates it to the truck or he allows the truck to leave without an additional order.
There may be other cases that make need for a *revoke* message for a previously announced unbooked leg e.g., a new order received from the bulletin-board or that the truck could rearrange his route and does not have the unbooked leg any more. All these cases are synchronized by the company agent. The protocol primitives for this cooperation mechanism are sampled in figure 4. Together with the protocols for the coupling of inter- and intra-regional traffic and the special CNP within one company[6], the negotiation mechanism built upon them is the key for the completely decentralized task decomposition process in this domain.

6 Conclusion and Outlook

In this paper we have described a new framework for task decomposition in dynamic societies of au-

[5] Another possibility would be to keep it and to wait for a suitable order.

[6] for reasons of space we have to omit them here

Truck → Company:	
ANNOUNCE:	announce an unbooked leg.
REVOKE:	revoke a previously announced unbooked leg.
FINISH:	Indicates the end of the validity of a previous announcement, triggers the confirmation the leg to a suitable applicant.
Company → Truck:	
INSERT:	if a suitable applicant has been elected, the truck shall update its plan.
FREE:	response to a REVOKE, by which the company frees the truck from its obligation as regards a previously announced unbooked leg.
Company → Company:	
OFFER:	A company offers free capacity to another company.
REVOKE:	A company revokes a previously offered unbooked leg.
APPLY:	A company applies to an offer for an unbooked leg.
GRANT:	A company grants the unbooked leg to a suitable applicant.

Figure 4: Protocol Primitives for the Unbooked Leg Cooperation

tonomous agents. We presented a general model of task decomposition in Multi-Agent Systems. Then, starting from the Contract Net model for task decomposition and allocation we have developed two stages of higher decentralization, the *Decentralized Task Decomposition Model* and the *Completely Decentralized Model.* These models provide higher flexibility in cooperative problem solving processes, and more adequate forms of communication and coordination between agents having equal rights. We gave an idea how negotiation can be applied for task decomposition problems in a concrete scenario, the MARS scenario of shipping companies.

Certainly, many important questions in the area of task decomposition and negotiation are not covered by this paper. E.g., one question concerns the negotiation process itself: what directives guide the decision process of an agent during a negotiation. In this respect, questions of agents beliefs and partner modeling [21, 10, 11] are of interest. Another question is to what extent agents are willing to make concessions in a negotiation which depends on factors such the degree of their general cooperativeness, the commitments they made [19, 12, 14], the behaviour of the partner (see for example some game-theoretic approaches to negotiation, such as [26]), their knowledge about their capabilities and resources which includes autoepistemic knowledge[20] etc. These and relevant issues will be subject to further research in our group.

References

[1] O.J Bartos. *Process and Outcome of Negotiation.* Columbia Univ. Press, New York and London, 1974.

[2] A. Bond and L. Gasser. *Readings in DAI.* Morgan Kaufmann, Los Angeles, CA, 1988.

[3] M. Buchheit, N. Kuhn, J. P. Müller, and M. Pischel. Mars: Modelling a multiagent scenario for shipping companies. In *Proc. ESS92, Dresden.* SCS, 1992.

[4] S. Bussmann. Simulation Environment for Multi-Agent Worlds. Techn. Rep. D-92-01, DFKI, 1992.

[5] R. Davis and R.G. Smith. Negotiation as a metaphor for distributed problem solving. In *Artificial Intelligence, 20(1),* 1983.

[6] Y. Demazeau and J.-P. Müller, editors. *Decentralized A.I.* North-Holland, 1990.

[7] Y. Demazeau and J.-P. Müller. *Decentralized A.I. 2.* North Holland, 1991.

[8] E. H. Durfee and V. R. Lesser. Negotiating task decomposition and allocation using partial global planning. In [13].

[9] E. H. Durfee and T. A. Montgomery. A hierarchical protocol for coordination of multiagent behaviour. In *Proc. of the 8th National Conf. on AI.* Boston, 1990.

[10] R. Fagin and J. Y. Halpern. Uncertainty, belief, and probability. *Computational Intelligence,* 7, 1991.

[11] R. Fagin and J. Y. Halpern. Two views of belief. *Artificial Intelligence,* 54(3), April 1992.

[12] L. Gasser. Social Conceptions of Knowledge and Action: DAI Foundations and Open System Semantics. *Artificial Intelligence,* 47, 1991.

[13] L. Gasser and M.N. Huhns. *Distributed Artificial Intelligence, Volume II.* Research Notes in Artificial Intelligence. Morgan Kaufmann, San Mateo, 1989.

[14] C. Hewitt. Open information systems semantics for distributed artificial intelligence. *Artificial Intelligence,* 47, 1991.

[15] M.N. Huhns. *Distributed Artificial Intelligence.* Pitman/Morgan Kaufmann, San Mateo, CA, 1987.

[16] T. Kreifelts and F. v. Martial. A negotiation framework for autonomous agents. In [7]

[17] N. Kuhn and H.-J. Müller. Multi-agent planning. In S. Biundo and F. Schmalhofer, editors, *DFKI-Workshop on Planning,* 1992.

[18] A. Laux and J. P. Müller. On the representation of knowledge and belief in multiagent systems. In *Proc. of 3rd. Workshop on Belief Representation and Agent Architectures.* Univ. of Durham, UK, 1992.

[19] H. J. Levesque, P. R. Cohen, and J. H. T. Nunez. On acting together. In *Proc. of the 8th National Conf. on AI.* Boston, MA, 1990.

[20] J.-J. C. Meyer, W. van der Hoek, and G. A. W. Vreeswijk. Epistemic logic for computer science: A tutorial (part two). In *Bulletin of the EATCS,* volume 45, Europ. Assoc. for Theor. Comp. Sc., 1991.

[21] Y. Moses. Resource-bounded knowledge and belief. In M. Y. Vardi, editor, *Proc. of the Second TARK.* Morgan-Kaufmann, Inc., 1988.

[22] D.G. Pruitt. *Negotiation Behavior.* Academic Press, 1981.

[23] R. G. Smith and R. Davis. Frameworks for Cooperation in Distributed Problem Solving. In [2].

[24] K. P. Sycara. Multiagent compromise via negotiation. In [13]

[25] K.J. Werkman. Knowledge-based model of negotiation using shareable perspectives. In M. Huhns, editor, *10th International Workshop on DAI, Bandera.* MCC Techn. Rep. No. ACT-AI-355-90, 1990.

[26] G. Zlotkin and J. S. Rosenschein. Negotiation and Task Sharing Among Autonomous Agents in Cooperative Domains In *Proc. of the 11th IJCAI,* pg. 912–917, Detroit, Michigan, 1989.