Chapter 5

Task Allocation Protocol in TRACE

The organizational knowledge that agents in TRACE possess only specifies the list of agents that currently belong to its organization. Agents need to identify team members for joint problem solving. Since agents do not maintain explicit models of other agents’ capabilities, a mechanism must be formulated for agents to find a 'capable' agent who is 'available' to perform this joint action. This is the objective of the task allocation protocol (TAP).

As agents operate in complex and dynamic environments, it is necessary to ensure that the activities of agents always remain coordinated. Joint intentions [101,102] guide problem-solving activity and play a key role in guaranteeing coordination among agents within an organization. In addition to this, we assume that agents possess anytime solutions [129] to goals. This is done so that an executing goal (that has anytime solution) can be terminated before its normal completion in order to accommodate higher priority requests.

Section 5.1 gives an overview of the task allocation protocol. Section 5.2 describes the task allocation protocol in detail. The method for resolving temporal conflicts among intentions is explained in Section 5.3. Section 5.4 provides the results of our experiment and finally Section 5.5 gives the conclusions.

5.1 Overview

Problem solving requests with priorities and deadlines arrive at each organization. The requests arriving at an organization are processed cooperatively by the agents of that organization and independently of other organizations.

To establish joint activity, an agent must firstly recognize the need for it. The agent who does this, is deemed the manager or organizer. Each social action has one organizer and at least one team member called the contractor (an acquaintance who
has agreed to participate). The manager's role involves obtaining a recipe, from the planner, contacting all the other agents of its organization to identify team members, determining when the actions will be performed and matching the team members with the actions to be performed.

If a goal can be achieved solely by the agent that receives it from the environment, it is a case of individual problem solving. Recognizing that a goal cannot be achieved all by itself puts the agent in an organizer's or manager's role. It then has to seek team members or contractors.

Once the need for joint action has been ascertained, the responsibility model requires the following conditions to be fulfilled before it can commence, other agents who are willing to participate and are able to contribute something must be identified, the fact that a common solution is required needs to be acknowledged, participants must agree to obey the responsibility code of conduct (described in Chapter 2), and finally the common solution by which the social goal will be attained must be developed.

The task allocation protocol does the following tasks:

- Identification of team members in order to achieve a social goal, and
- Development of a common solution that is mutually acceptable to the organizer and the team members

As the organizer has a recipe, which specifies the sub-goals (tasks) and their temporal orderings, development of a common solution involves finding the actual time at which the sub-goals can be executed. The fact that agents in a team will obey the responsibility code of conduct is implicit and does not require agents to acknowledge this for every joint activity.

We aim at developing a task allocation protocol for open multi agent systems where agents dynamically enter and leave the system. It is therefore difficult for agents to
always maintain a correct model of others' capabilities. In a scenario like this, ensuring that each of the above listed conditions is satisfied separately before the commencement of joint problem solving, involves a high degree of communication among agents. This will slow down the speed of operation of the system. To overcome this difficulty, the protocol that we propose settles more than one condition in a single message interchange.

The process of finding a team member and agreeing upon a suitable time is done for every sub-goal of the recipe in the temporal order specified by the planner. During this process, priorities are used to resolve any temporal conflicts that arise with preexisting commitments. The lower priority task is either rescheduled to accommodate a more critical task, or decommitted altogether if deadlines make rescheduling impossible. Deadlines therefore ensure termination of the protocol.

For the purpose of ensuring coordination among team members, all social activity is represented as a joint intention, which includes the list of team members, their roles and the common solution. Whenever an agent reschedules or decommits a goal, it notifies all associated team members. This keeps the entire team aware of the current state of problem solving activity and results in all team members either together progressing on the solution, or together dropping a goal if it is found unachievable.

As our protocol is directed toward time constrained domains, the planner determines if an anytime solution is available for the sub-goals. If so, it associates a minimum amount of time that needs to be spent for obtaining a meaningful solution. After this period of time elapses, execution of the sub-goal can be terminated to accommodate more critical requests, or continued to completion otherwise.

5.2 Task Allocation Protocol

The protocol is based on the following assumptions: Firstly, it is assumed that the communication is foolproof and that the message delay time is known to all agents.
Secondly, in order to carry out task allocation activity, agents share a global clock reference. Thirdly agents are able to accurately predict the time taken in terms of the global clock, to execute each domain level task. This facilitates the task allocation process and enables agents to make and honor commitments in a controllable manner.

The following notation is used in the discussion that follows

• $a_i$ denotes action $i$
• $A_i$ denotes agent $i$
• $G_i$ denotes a goal
• $T_i$ denotes the time at which action $i$ is executed

After establishing that a goal is social, the organizer instantiates a representation of the social goal as a joint intention in its self-model (see Figure 5.1). The motivation slot indicates the reason for carrying out the joint intention. The recipe is a series of actions, which need to be performed together with some temporal ordering constraints, which will produce the desired outcome. The actions in Figure 5.1 $(a_1, a_2, a_3, a_4)$ are temporally ordered. The values $l_1, l_2, l_3$ and $l_4$ indicate the lower bounds on execution time for each of the actions, since actions are assumed to have an anytime solution. This protocol is however not limited to actions having anytime solutions. If an anytime solution is not available, then $l_i$ is the fixed period of time that needs to be spent for obtaining the solution. The recipe indicates what is to be done and in what order, not who is to do it nor the exact time at which it should be done.

Problem solving requests are assumed to arrive with an associated deadline. If an incoming request does not have an associated deadline, TRACE associates a default deadline with it. This is done to ensure termination of the task allocation process.

The start time and end time indicate the commonly agreed time for starting and ending the joint activity. The priority slot indicates the local agent's assessment of the importance of the intention and is used as the basis for computing its desirability.
Priorities are application dependent and the issue of determining priorities is therefore not addressed here.

The status slot of joint action description refers to the current activity of the task allocation protocol and has the value 'establishing group & developing solution' or 'executing joint goal'. Contribution slot records those agents that are capable of contributing and have agreed to the joint action. Initially the organizer, agent A1, has agreed to contribute by performing the actions a1 and a3. No other agent has yet agreed or even been asked to contribute anything. Contractors now need to be found for performing a2 and a4.

Name: G
Motivation:
Recipe: a1 t1, a2 t2, a3 t3, a4, t4
Deadline:
Start Time: End Time:
Priority: 23
Status: DEVELOPING SOLUTION / EXECUTING
Contributions:
A1      ORGANIZER     a1     t1     AGREED
? TEAM MEMBER    a2     t2
A1      ORGANIZER     a3     t3     AGREED
? TEAM MEMBER    a4     t4

Figure 5.1 Representation of Social Action (Joint Intention) in self model of A1

Having identified the need for joint goal, the process of establishing it and arming at a common solution can commence. This requires finalizing the detailed timings and duration of the actions. The team leader prepares an initial proposal for the individual action timings and fills in the joint goals duration and its start and end times in the joint intention.

A1      a1     t1
?       a2     t2
A1      a3     t3
?       a4     t4
The timing proposal takes into account the fact that some time is required to agree to the solution, work cannot commence immediately. The formula for calculating the time lag is given below. It takes the following factors into consideration: For each action which needs to be performed by an acquaintance at least two messages must be transferred (announcement to all agents of the organization and bid to the organizer) to establish its start time; an agent takes time to process a message and then an award message must be sent to all team members when the solution is agreed upon.

\[
\text{Start Time} = \text{current time} + \\
2 \times \text{number of nonlocal acts} \times \text{communication delay} - \ast \ast \\
3 \times \text{number of actions} \times \text{estimated time to process message} \\
+ \text{communication delay}
\]

The team organizer does not have a complete picture of the capabilities of other agents within the organization. In particular, the organizer does not know the existing commitments and desires of all its potential team members, so neither actions nor their exact timings can be dictated, they have to be negotiated. To avoid chaotic behavior and many iterations, the organizer takes each action in the recipe in a temporally sequential order. Note that only the task allocation is done sequentially - the tasks can be executed in parallel or overlapped if the plan has been so defined.

Consider a case where \( t_1 = 12 \), \( t_2 = 16 \), \( t_3 = 21 \), \( t_4 = 25 \). For each action, the organizer negotiates with the prospective team members (other agents of the organization) the appropriate time at which it should be performed. Thus action \( a_2 \) is negotiated first, and a time is agreed which fits in with the existing obligations of the prospective team member and the organizer's rating of the action's desirability (priority). Then \( A_1 \) finds a suitable time for \( a_3 \) and so on for each of the actions.

As agents do not maintain models of other agents that represent their capabilities, the organizer describes the task to the entire organization (the list of agents that comprise the organization is maintained in the organizational knowledge mentioned in Chapter 4) by broadcasting a *task announcement* message. This message, (see
Figure 5.2), indicates that the sender wishes to establish a joint goal and arrive at a common solution involving the recipient, states the team organizer's priority for the task, the action for which a contribution is required the time at which the action needs to be started, and a lower bound on execution time (for anytime solutions) that the prospective team member is expected to spend for that action.

Sender: A
Receiver: all agents within the organization
Type: TASK ANNOUNCEMENT MESSAGE
Contents:

Joint Goal G
Priority 23
Contribution a2
Contribution Time t2
Lower bound on time l2

Figure 5.2 Task Announcement Message

Upon receipt of proposal the team members evaluate it to see whether it is acceptable; refer to Section 5.3 for further details of this process. If there is no conflict, the agent sets up a joint intention similar to that of Figure 5.1, and an individual intention as shown in Figure 5.3. The motivation slot indicates the goal for which contribution is required. The status slot is 'pending'.

Action: a3
Motivation: Joint Goal G
Start Time: t2
Duration: Priority
Status Pending Lower bound on time l2

Figure 5.3 Individual Intention Representation for Agent A2

Agents then return a message indicating their acceptance to the team organizer (see Figure 5.4). The 'priority of decommited goal' slot indicates whether the prospective team member is able to accommodate the request by decommitting a pre-existing lower priority goal, and if so, the priority of that goal. The organizer can use this
information as the basis for selecting a team member during the bid evaluation process.

Sender: $A_2$
Receiver: $A_1$
Type: BID - ACCEPTANCE
Contents:

Joint Action $G$
Priority of decommited goal:
Contribution $a$
Contribution Time: $t_2$
Lower bound on time: $l_2$
Response: OK

Figure 5.4 Bid message indicating acceptance

If the suggested time is unacceptable, the prospective team member proposes a time at which the action can be fitted in with its existing commitments, makes a tentative commitment for this time and returns the suggestion to the organizer (see Figure 5.5). If the modified time is acceptable to the organizer, it will make appropriate adjustments to the subsequent solution timings and proceed with the next action. If the modified time proposal is unacceptable, the organizer will look for a new agent to perform the action from its list of proposed contributors.

Sender: $A_2$
Receiver: $A_1$
Type: BID - MODIFIED TIME
Contents

Joint Goal $G$
Priority of decommited goal
Contribution $a_2$
Proposed Contribution Time
Modified Contribution Time $t_2$
Lower bound on time
Response NOTOK

Figure 5.5 Bid message with modified time
From among the agents willing to participate, the organizer selects as team member the agent that can perform the task earliest. If there is more than one agent that can perform the task, the organizer selects the one which can perform it by decommitting the lowest priority task.

The process of agreeing at a time for each action continues until all actions have been successfully dealt with. At this point the common solution is agreed upon and the organizer informs all the team members of the final solution by means of an award message (see Figure 5.6).

The joint intention status slot is changed to 'executing-joint-action' and the contribution slot is updated to indicate that all team members have agreed to the goal, and a common solution and implicitly to the responsibility code of conduct, and are now in the process of executing the joint action. The status slot in the individual intention is also changed to 'executing'. On receiving the award message, the team members also make similar changes to their joint and individual intentions and become contractors for that goal. All the preliminaries for joint action have been satisfied and group action can begin.

Sender: A₁
Receiver: A₂
Type: JOINT SOLUTION AGREED (AWARD)
Contents:

```
Joint Goal G
Solution
A₁  a₁  t₁
A₂  a₂  t₂
A₃  a₃  t₃
A₄  a₄  t₄
```

Figure 5.6 Notification of Start of Joint Action (Award Message)

After completing an allocated task, the team members report the results of execution to the organizer (see Figure 5.7).

Sender: A₂
5.3 Resolving Temporal Incompatibilities

In order to exhibit correct behavior, agents need to ensure that their intentions always remain compatible. Two intentions are said to be incompatible if the times for which they are scheduled overlap, they are compatible if they are distinct. Consider an agent having two intentions for tasks a_1 and a: represented in its self-model as shown in Figure 5.8. These two intentions are compatible because the times at which they are carried out, 5 to 12 and 12 to 16 do not overlap.

![Figure 5.8 Consistent Intentions](image)

Before the commencement of their execution a new request arrives that corresponds to the intention shown in Figure 5.9.

![Figure 5.9 Request Intention](image)
Figure 5.9 New Intention

The inconsistency resolver now has to determine whether the new proposal is compatible with the agents existing intentions. As a result of this analysis the inconsistency resolver will indicate that the new intention is compatible because even though the times overlap, $a_2$ requires an anytime solution (lower bound $<$ duration) and can therefore be accommodated with the new intention. This requires termination of $a_2$ at time 15 in order to start $a_3$.

In case an anytime solution is not available for $a_2$, then it becomes incompatible with $a_3$. The inconsistency resolver resolves this by making use of the priority values for each of the intentions. If the new request is less desirable than the existing commitments, then the agent proposes a modified time that can be fitted in with the existing commitments. In the above example however, $a_3$ is more desirable Therefore the agent forms the intention to achieve $a_3$ from time 15 to 20 and reschedules $a_2$ after $a_3$. The new intention for $a_2$ now becomes

- **Name:** $a_2$
- **Motivation:** $G_2$
- **Start Time:** 20
- **Max End Time:** 24
- **Duration:** 4
- **Priority:** 8
- **Status:** Pending
- **Lower bound on Time:** 4

The other actions of $G_2$ that get affected due to this change also need to be rescheduled. If the new schedule for $G_2$ does not conform to its deadline, then the agent decommits $G_2$ and updates the number of recommitments This is all that needs to be done if $G_2$ is a primitive goal However if $a_2$ corresponds to a joint goal just rescheduling $a_2$ is not enough The agent must inform all team members about its decommitment to the originally agreed solution (see Figure 5.10)
Chapter 5. Task Allocation Protocol in TRACE

Sender: A₂
Receiver: All team members
Type: DECOMMITMENT TO SOLUTION
Contents
Joint Goal G₂
Priority
Contribution a²
Old Contribution Time: t₁ New Contribution Time: t₂
Lower bound on time:

Figure 5.10 Decommitment to solution Message

Upon receipt of this message, the other team members also drop commitment to the common solution. When the team organizer receives this message it reschedules G₂ if possible, otherwise decommits the goal G₂, informs all team members about the decommitment to the joint goal (see Figure 5.11), and updates the number of decommitments. In this way all organizers record information about their decommitted goals and convey this information to the resource manager, which utilizes it for performing resource allocation (explained in Chapter 6)

Sender: A₁
Receiver: All team members
Type: DECOMMITMENT TO JOINT GOAL
Contents
Joint Goal G₂
Priority
Contribution a₂
Contribution Time
Lower bound on time

Figure 5.11 Decommitment to joint goal message

The task allocation protocol is summarized in Figure 5.12

5.4 Experiment

The inclusion of anytime solutions results in a considerable improvement in the performance of agents. This can be demonstrated by an experiment (see Figure 5.13). The protocol was implemented in Java and run for two organizations of five agents
Chapter 5. Task Allocation Protocol in TRACE

each. Agents in one organization used anytime algorithms and agents in the other used standard algorithms. Several problem-solving requests were made randomly to each of these organizations, half of which were assumed to have anytime solution. The organizations can handle requests (without decommitments), if they arrive at

![Team Organizer Diagram]

**Figure 5.12 Task Allocation Protocol**

<table>
<thead>
<tr>
<th>Task arrival rate (every n sec)</th>
<th>Anytime algorithm % goals decommited</th>
<th>Standard algorithm % goals decommited</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>46</td>
<td>93</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>55</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>28</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 5.13 Performance of agents using anytime and standard solutions**
Chapter 5. Task Allocation Protocol in TRACE

intervals of 16 or more. If the frequency of requests increases, the number of decommitments also increases correspondingly. The performance was measured in terms of the goals that were decommited. Since "anytime" algorithms can provide 'some' solutions even in lesser time, the agent can take up other goals if required. As a result, the number of decommitments is far less compared to the organization with standard solutions.

5.5 Conclusions

This chapter described the task allocation protocol. Tasks have an associated deadline and priority. They are assigned to agents so that higher priority tasks are executed in time. Since the computational load on any organization of the multi-agent system is unpredictable, a situation could arise where an organization is overloaded, but the multi-agent system as a whole has the required resources to take on that load. The resource manager of each organization collects statistics of goals decommited from its agents and performs resource reallocation. This is described in the next chapter.