Chapter 7
Conclusions and Future Work

This is the concluding chapter of this thesis and summarizes the main contributions and the methodology used in this research. Finally some areas for further work are presented.

7.1 Summary and Methodological Issues

The rapid proliferation of multi-agent system applications, like management of networks for electricity transport and telecommunication, in factories to control manufacturing processes, in the medical domain for monitoring the condition of patients in an ICU, etc, places increased demands on the multi-agent system builder [97]. The multi-agent systems of today are expected to operate in increasingly complicated environments that are dynamic and unpredictable. The type and frequency of requests in these applications varies non-deterministically. Consequently it is becoming increasingly important to address the issue of adaptability of multi-agent systems to changing environmental conditions.

This research started with the aim of developing an adaptive organizational policy for multi-agent systems that are targeted towards soft real-time domains. Our intention was to obtain a robust multi-agent system that could withstand computational load variations. As agents operate in dynamic and unpredictable environments, where their beliefs and goals are neither correct nor complete, it is certainly necessary for each agent to be endowed with team rationality [96]. This helps in keeping the amount of wasted resources to a minimum. In order to cater to time constrained domains, there also has to be a means of preempting lower priority tasks in preference to higher priority ones.

However, it is not enough if individual agents possess team rationality and the ability to preempt tasks; there should also be a means of handling computational load variations. This is possible only if the entire multi-agent system has the ability to
dynamically change its organization to suit the existing problem solving requirements. This reorganization should be done so that the demand for agents with a certain skill always matches the supply of agents having that skill. In case of overloads, preference has to be given to higher priority tasks.

Existing methods for implementing organizational policies [8,24,62,68,71,111], address only some of the above mentioned issues in isolation and hence cannot meet all the requirements of dynamic adaptation. This thesis proposes a single comprehensive organizational policy (TRACE) that can operate under time constraints and varying computational loads.

The entire MAS is viewed as consisting of several problem-solving organizations. Each organization in turn consists of multiple agents and a resource manager. Problem solving requests with an associated deadline arrive at the agents of these organizations. A request that arrives at an organization is solved cooperatively by agents within that organization and independently of the other organizations. The rate of arrival of problem solving requests at each of these organizations varies with time. As a result of this variation, the requirement for resources at each organization also keeps changing. At any particular instant, some organizations may have additional resources, while others may need more resources resulting in some requests that cannot be completed in time. In order to minimize these lost requests, the resource managers dynamically reallocate resources to organizations so as to balance the demand for resources with its supply. This is done by means of a market-oriented protocol. Whenever reallocation is done, the most critical tasks are allocated resources first.

Following a layered approach, the problem of designing such an organizational policy, is divided into the following two sub-problems.

1. Allocation of tasks to agents within an organization through the task allocation protocol (TAP), and
2. Allocation of resources to each of these organizations, through the resource allocation protocol (RAP).

The agents in each organization cooperatively process problem-solving requests by making use of the TAP. At regular intervals of time (called reorganization cycle), these agents report statistics of the general pattern of requests, the number of decommitments, the percentage idle time, and an indication of the criticality of the decommited requests through funds. The resource managers apply the RAP to this information and reallocate resources. This causes an increase/decrease in the number of agents in the organizations and the distribution of knowledge to agents, and thereby reorganizes the multi-agent system.

To evaluate the effectiveness of the proposed approach simulated experiments were done. The behavior of the system was studied under varying problem solving demands. This was compared with the behavior of a MAS with fixed number of agents. On the basis of these experiments it is demonstrated that the proposed protocol possesses the following properties:

- Allows agents to
  
  i) Adapt to unpredictable changes in problem solving environment (by keeping its beliefs and goals always consistent with the latest information that it receives from the environment/ other agents).

  ii) Focus on higher priority tasks.

- Allows the multi-agent system to
  
  i) Adapt to changes in load by diverting resources where they are needed most

  ii) Add new agents for problem solving in an incremental manner.
Figure 7.1 Comparative Study

<table>
<thead>
<tr>
<th>Organizational Policy</th>
<th>Maintaining models of other agents</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchical</td>
<td>Yes</td>
<td>Direct addressing</td>
</tr>
<tr>
<td>CNP</td>
<td>Yes</td>
<td>Directed addressing</td>
</tr>
<tr>
<td>SRM</td>
<td>No</td>
<td>Broadcast</td>
</tr>
<tr>
<td>Economic Models</td>
<td>Yes</td>
<td>Directed addressing</td>
</tr>
<tr>
<td>TRACE</td>
<td>No</td>
<td>Broadcast within an org for task allocation and additional communication for resource allocation once every reorganization cycle</td>
</tr>
</tbody>
</table>

Figure 7.2 Comparison of Overheads

Figure 7.1 shows a comparison of TRACE with the existing mechanisms. Figure 7.2 compares the overhead associated with all these methods. As can be seen from these figures, TRACE possesses all the required features and therefore can be said to be...
the most adaptive of all. This adaptability is achieved at the cost of a small increase in overhead that is incurred, once every reorganization cycle, for reorganizing the multi-agent system.

7.2 Future Work

TRACE meets all the requirements listed in Figure 7.1, and is therefore more adaptive than the other approaches. However, as is the case with any computational solution, this mechanism also has certain areas for improvement. The first is fault tolerance. In TRACE it is currently the responsibility of the application to recover from failures. In future we intend to achieve goal survivability, that is, whatever might happen to the individual agent, the multi-agent system makes the commitment to meet the goal. Consequently, TRACE should ensure task reassignment to accomplish the task.

Another issue not addressed in this thesis is multi-level contracting. We allowed single level contracting, where a single team organizer and some team members engage themselves in joint problem solving. This framework could be extended, by having team members delegate part of the job, allocated to them by their organizer, to other agents and in turn become organizers for those parts. We could then apply this framework to supply chains by having both linear and tree like organizer-contractor relationships.

In addition to this we propose to work towards dynamically varying the reorganization cycle time. Since the environment is unpredictable, having a fixed cycle time may not be acceptable always. The introduction of some means of dynamically varying the reorganization cycle time will make the framework more adaptive.

In TRACE, allocation of resources is done using the price oriented approach. This results in allocations that are fair but not perfectly feasible. In future we will
incorporate the resource directed approach also into this framework. This will provide the application with an option of selecting one of these two methods depending on its desire to achieve fairness or feasibility.